Ridgetop Group Inc
ENGINEERING INNOVATION

ARULE™

User Guide

To maintain the quality of our publications, we need your comments on the accuracy, clarity, organization, and value of this book.

Address correspondence to:

Technical Support

Ridgetop Group Inc.
3580 West Ina Road
Tucson, AZ 85741 USA
Phone: +1 (520)–742–3300
Email: support@ridgetopgroup.com

Revision Record

Release	Date	Remarks
1.0	July 28, 2021	First release.
1.1	December 02, 2021	Updated installation process.
1.2	January 06, 2022	Second release.
1.3	February 15, 2022	Added XFD5 – XFD9.
1.4	June 02, 2022	Updated Tables & Figures.
1.5	February 13, 2023	New GUI and updated examples.
1.6	April 24, 2023	Added Sections 1.2,1.3, 1.4, 1.5, 2.5, 2.6, all of Section 3, and Section 5.3.
1.7	January 17, 2024	Updated release.

Contact Information

Corporate Headquarters:

Ridgetop Group Inc.
3580 West Ina Road
Tucson, AZ 85741 USA
Email: support@ridgetopgroup.com (Tech. Support)
sales@ridgetopgroup.com (Sales & Marketing)
Phone: +1 (520)–742–3300
Fax: +1 (520)–544–3180
Website: www.ridgetopgroup.com

Table of Contents

1. Introduction ... 1
  1.1 ARULE™ Overview ... 2
  1.2 3-Stage Prognostic Health Management (PHM) System Architecture ... 3
  1.3 Sensing Stage ... 4
    1.3.1 Noisy Data ... 5
    1.3.2 Continual and Periodic Sampling ... 5
    1.3.3 Periodic Burst-Mode Sampling with Averaging ... 6
  1.4 Feature Extraction Stage: Application-Specific Data Processing ... 6
    1.4.1 Data-Processing Methods ... 7
    1.4.2 Extracting FD ... 7
    1.4.3 Modeling FD ... 9
    1.4.4 FD Signatures ... 9
  1.5 Prognosing Stage ... 10
    1.5.1 Signatures Subsystem ... 10
    1.5.2 Prognostics Subsystem ... 12
2. ARULE: Description and Operation ... 13
  2.1 Graphical User Interface Diagram ... 14
  2.2 Definition Files ... 14
  2.3 Directory Structure: Description and Contents ... 16
  2.4 BNF for Definition Files ... 16
  2.5 ARULE: Signature Processing ... 17
    2.5.1 Transforming an FD Signature into a Fault-to-Failure (FFP) Signature ... 18
    2.5.2 Transform FFP Signature Data into DPS Data ... 19
    2.5.3 Significance of the FFP-to-DPS Transform ... 20
    2.5.4 The FFS-to-DPS Model ... 20
    2.5.5 Transform FFP to DPS Example: Modeling and Specification ... 20
    2.5.6 Transforming DPS Data into Functional-Failure-Signature Data ... 21
  2.6 ARULE: Prognostic Processing and Filtering Methods and Results ... 23
    2.6.1 Second Model for EKF Processing ... 24
    2.6.2 Estimated Prognostic Information ... 24
    2.6.3 Examples of Selecting and Specifying the 2nd EKF Model ... 25

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

2.6.4 Example Plots: Prognostic Estimates 26

3. Prognostic Information: Evaluation of Accuracy 27

3.1 Ideal Prognostic Information 27

3.2 Idealistic, But Not Exact, Prognostic Information 28

3.3 Accuracy 30

3.3.1 Near-real Time Estimates of Accuracy 30

3.3.2 Alpha (α), Margin of Accuracy 31

3.3.3 Convergence of Accuracy 31

3.4 Contributing Factors Related to Inaccuracy 33

3.4.1 Signature Processing 33

3.4.2 External Factors 34

3.5 Prognostic Information: Review of RUL, PH, and SOH 34

3.5.1 RUL 35

3.5.2 PH 35

3.5.3 SOH 36

3.6 Prognostic Events 36

3.6.1 Beginning of Degradation 36

3.6.2 End of Life – Functional Failure 37

3.7 Prognostic Distance (PD) 37

3.7.1 True Prognostic Distance 37

3.7.2 Maximum Prognostic Distance 38

3.7.3 Beginning of Degradation, Offset 38

3.7.4 End of Life, Offset 38

3.7.5 Methods to Mitigate Offset Error 38

3.8 Summary of Prognostic Terminology and Performance Metrics 39

4. ARULE™ Installation 41

4.1 System Requirements 41

4.2 Installing the Software for the First Time 41

4.3 Activating License and Verifying Software Installation 45

4.4 ARULE™ Views and Display Scale and Layout 49

5. ARULE™ User Interface 50

5.1 ARULE™ GUI Overview 50

5.2 System Definition Input 51

5.2.1 System Definition (SDEF) Files ... 51
5.2.2 Node Definition (NDEF) Files ... 54
5.2.3 Node Definition User Form ... 56
5.2.4 Raw NDEF File in Notepad ... 58
5.2.5 Input Feature Data Preview ... 58
5.2.6 ARULE™ Run Arguments ... 59
5.3 Data Output ... 62
5.3.1 Data Output Dropdown Options ... 63
5.3.2 Data Outputs in .csv File ... 65
5.3.3 System Messages ... 67
5.4 ARULE™ Global Features ... 68
5.4.1 Global Menu Options ... 68
6. Real World Examples ... 70
6.1 DEMO1 ... 71
6.1.1 DEMO1: Definitions ... 72
6.1.2 DEMO1: Input Feature Data ... 74
6.1.3 DEMO1: Data Outputs and Prognostic Information ... 74
6.1.4 DEMO1: Discussion ... 75
6.2 DEMO2 ... 75
6.2.1 DEMO2: Definitions ... 75
6.2.2 DEMO2: Input Feature Data ... 78
6.2.3 DEMO2: Data Outputs and Prognostic Information ... 78
6.2.4 DEMO2: Discussion ... 80
6.3 XFD1: Transmission Power ... 81
6.3.1 XFD1: Definitions ... 81
6.3.2 XFD1: Input Feature Data ... 83
6.3.3 XFD1: Data Outputs and Prognostic Information ... 84
6.3.4 XFD1: Discussion ... 84
6.4 XFD2: Helicopter Gear Box – Pinion Gear ... 85
6.4.1 XFD2: Definitions ... 85
6.4.2 XFD2: Input Feature Data ... 87
6.4.3 XFD2: Data Outputs and Prognostic Information ... 88
6.4.4 XFD2: Discussion ... 89

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

6.5 XFD3: Ripple Voltage, PC Power Supply, Full–PI filter ... 91
6.5.1 XFD3: Definitions ... 91
6.5.2 XFD3: Input Feature Data ... 93
6.5.3 XFD3: Data Outputs and Prognostic Information ... 94
6.5.4 XFD3: Discussion ... 95
6.6 XFD4: Ripple Voltage, Helicopter Power Supply, Half–PI filter ... 96
6.6.1 XFD4: Definitions ... 96
6.6.2 XFD4: Input Feature Data ... 97
6.6.3 XFD4: Data Outputs and Prognostic Information ... 98
6.6.4 XFD4: Discussion ... 98
6.7 XFD5: Capacitor Leakage, Power Supply Filter ... 99
6.7.1 XFD5: Definitions ... 99
6.7.2 XFD5: Input Feature Data ... 101
6.7.3 XFD5: Data Outputs and Prognostic Information ... 101
6.7.4 XFD5: Discussion ... 102
6.8 XFD6: Battery Internal Resistance ... 103
6.8.1 XFD6: Definitions ... 103
6.8.2 XFD6: Input Feature Data ... 104
6.8.3 XFD6: Data Outputs and Prognostic Information ... 105
6.8.4 XFD6: Discussion ... 105
6.9 XFD7: Battery Charging Capacity ... 106
6.9.1 XFD7: Definitions ... 107
6.9.2 XFD7: Input Feature Data ... 110
6.9.3 XFD7: Data Outputs and Prognostic Information ... 111
6.9.4 XFD7: Discussion ... 112
6.10 XFD8: High Data Rate Cables ... 113
6.10.1 XFD8: Definitions ... 116
6.10.2 XFD8: Input Feature Data ... 118
6.10.3 XFD8: Data Outputs and Prognostic Information ... 118
6.10.4 XFD8: Discussion ... 119
6.11 XFD9: Electro–mechanical Actuators (EMAs) ... 120
6.11.1 XFD9: Definitions ... 129
6.11.2 XFD9: Input Feature Data ... 133

6.11.3 XFD9: Data Outputs and Prognostic Information 135
6.11.4 XFD9: Discussion 136
7. How to Analyze Your Own Data with ARULE™? 138
8. Appendix 146
8.1 Frequently Asked Questions (FAQ) 146
8.2 Table of Figures 147

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

1. Introduction

ARULE™ is an innovative suite of software programs that interfaces with the Ridgetop Group Inc. (Ridgetop) advanced prognostic prediction kernel, Adaptive Remaining Useful Life Estimator (ARULE™). This leading–edge prognostic prediction kernel is based on over a decade of research and development, and has been proven effective for various Prognostic Health Management (PHM), Integrated Vehicle Health Management (IVHM), and Condition–Based Maintenance (CBM) applications. ARULE™ is a powerful reasoner that can determine the Remaining Useful Life (RUL), State of Health (SoH), and Prognostic Horizon (PH) of complex systems and subsystems. Working from acquired sensor data, ARULE™ employs an advanced prediction method related to extended Kalman Filtering as well as other techniques and methods to produce new RUL, SoH, and PH estimates for each new sensor data point.

Ridgetop, founded in 2000, is an established engineering and technology company that provides PHM, CBM, and reliability engineering solutions to government and commercial organizations around the world. ARULE™ is a key component in Ridgetop's technology set which is aimed at helping customers ensure precise identification and isolation of system anomalies, advance notice of impending failure, and the necessary combination of firmware, hardware, and software solutions for mission critical systems. More information on Ridgetop's complete list of products and services can be reviewed on our website at www.ridgetopgroup.com.

The sections in this document are the following:

1. Introduction
2. ARULE: Description and Operation
3. Prognostic Information: Evaluation of Accuracy
4. ARULE™ Installation
5. ARULE™ User Interface
6. Real World Examples
7. How to Analyze your Own Data with ARULE™
8. Appendix

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

1.1 ARULE™ Overview

ARULE™ is an advanced predictive analytics software application that was designed to process and analyze arbitrary types of Feature Data (FD) that was extracted from condition–based data (CBD) as a means to provide key prognostic estimates for Remaining Useful Life (RUL), State–of–Health (SoH), and Prognostic Horizon (PH). ARULE™ was designed based on Ridgetop's PHM framework as shown in Figure 1.

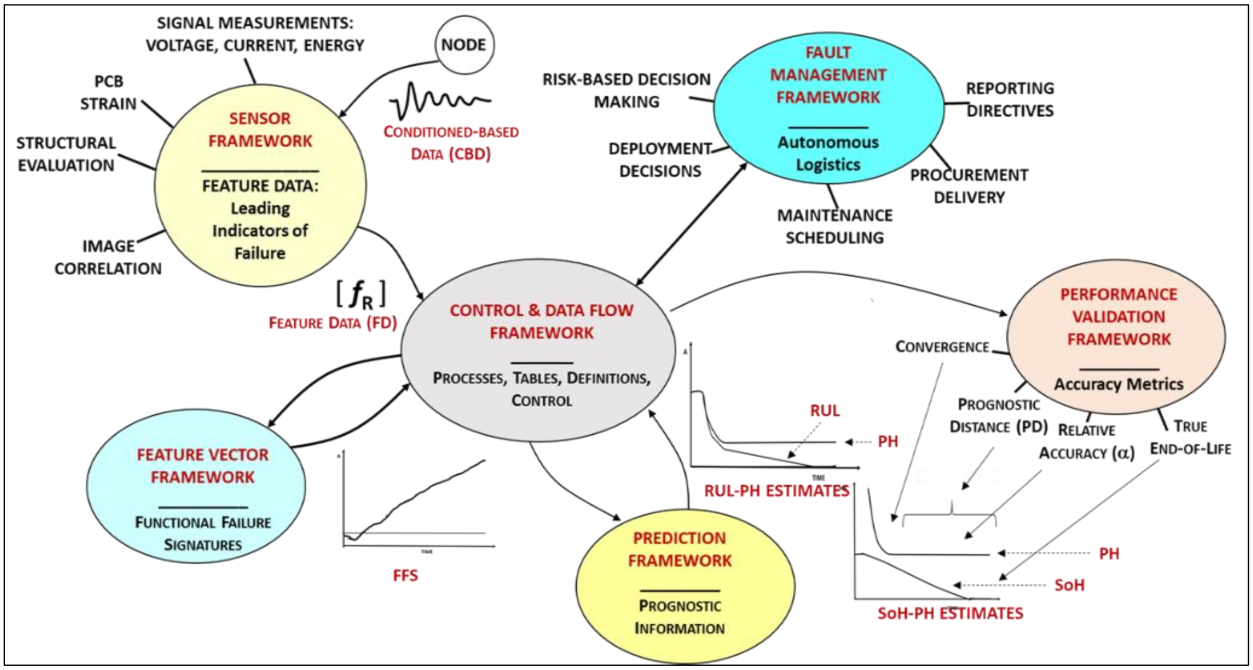


Figure 1 – Framework diagram for a PHM system

A PHM framework is a conceptual structure that can be realized in any number of ways. For example, the Feature Vector and Prediction Frameworks and their relationship to the Control and Data Flow Framework can be structured as a programming framework as shown in Figure 1, or as a partial collection of frameworks that can be integrated into a 3-stage PHM system architecture as shown in Figure 2.

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

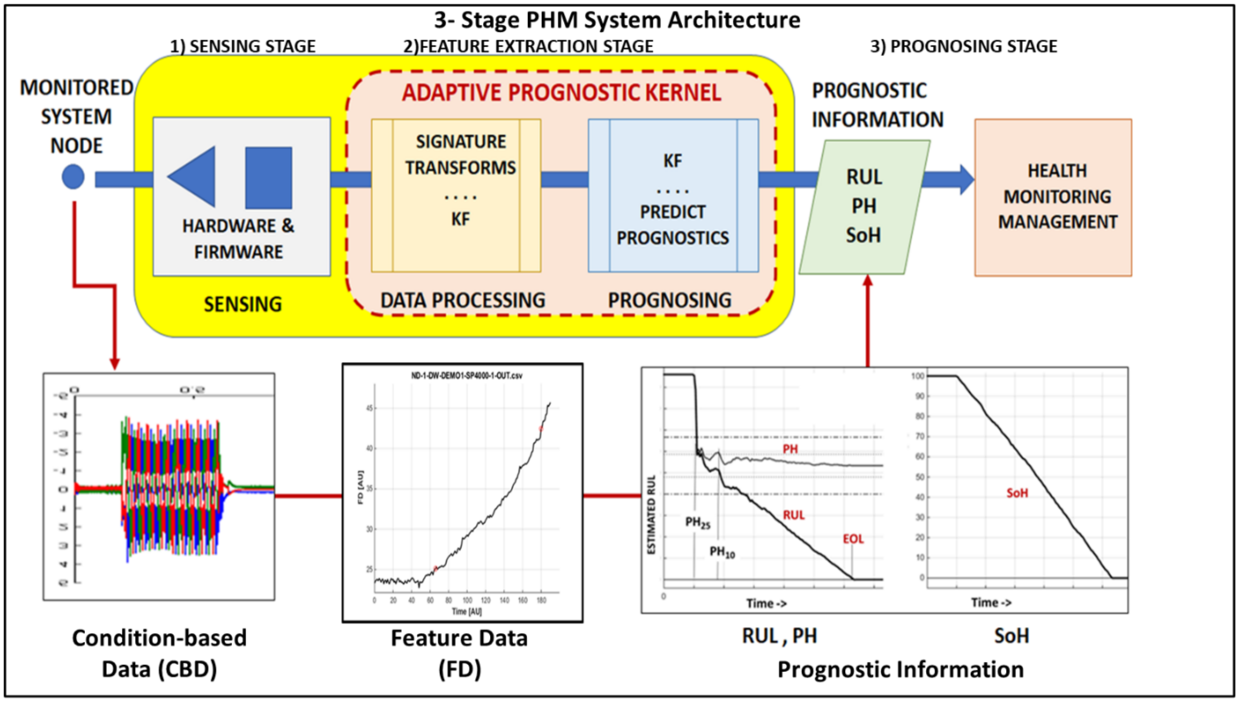


Figure 2 - Diagram of a 3-stage PHM system: (1) SENSING STAGE, (2) FEATURE EXTRACTION STAGE, and (3) PROGNOSING STAGE to Produce Prognostic Information with ARULE

Ridgetop's 3-Stage PHM system architecture can be used to conceptualize how different types of condition–based data (CBD) from which sets of Feature Data (FD) have been extracted, stored in files, and subsequently downloaded for post–processing can be analyzed to determine the health, service, and maintenance needs for a given system. Over the years Ridgetop has applied its PHM, CBM, and IVHM technologies to many critical applications. This User Guide will cover several real-world example data sets and will describe how to use ARULE™ to process and analyze unique end–user application data.

1.2 3-Stage Prognostic Health Management (PHM) System Architecture

A 3-stage PHM system architecture as shown in Figure 2 comprises the following:

- A Sensing Stage that monitors a system node or a collection of nodes that sense and collect condition-based data (CBD) indicative of damage or degradation.
- A Feature Extraction Stage with application-specific data processing routines that condition and transform CBD into Feature Data (FD) in a form or signature that is correlated to increasing levels of damage leading to functional failure.

- A Prognosing Stage that processes input Functional Failure Signature (FFS) data to prognose a future time of functional failure and outputs prognostic information such as remaining useful life (RUL), state-of-health (SOH), and prognostic horizon (PH).

The described 3-stage architecture is fully compliant with the IEEE 1856-2017 Standard Framework for Prognostics and Health Management of Electronic Systems, 7 Sep. 2017 (Figure 3):

- Sensing stage in Figure 2 corresponds to S and DA in Figure 3.
- Feature Extraction stage in Figure 2 corresponds to DA and DM in Figure 3.
- Prognosing stage in Figure 2 corresponds to SD, HA, and PA in Figure 3.
- The right-most two blocks after the last stage in Figure 2 corresponds to AG and HM in Figure 3.

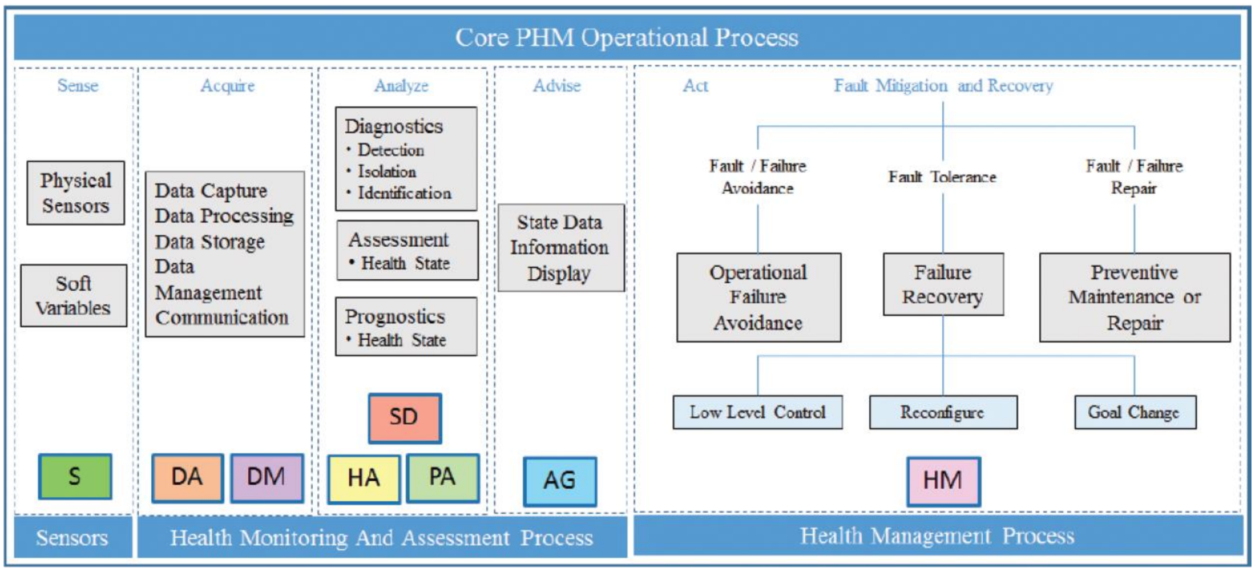


Figure 3 - General operational view of how the sensors, health monitoring, and assessment functions inherent within the electronic system enable the PHM capability operational processes [IEEE 1856-2017]

1.3 Sensing Stage

The sensing stage attaches to one or more system nodes being monitored for health and comprises one or more sensors that monitor the nodes by collecting condition-based data (CBD). These sensors can be electrical, mechanical, frequency-based, and so on. As an example, a monitored system node is the physical location on top of a quadcopter body that experiences minimal shock and vibration forces when the quadcopter is powered but is neither hovering nor flying. CBD examples include, vibration, shock, temperature (heat), light, and electrical (voltage and current). The output of the sensing stage is usually a filtered and/or windowed CBD output file. It is also important to note, that the there are many factors that impact the sensing stage such as noisy data, data sampling rates, sampling periods, etc. For some advanced PHM systems, the sensing stage may also have edge computing capabilities to also extract Feature Data (FD).

Significant subtopics related to the Sensing Stage are listed as follows:

- Noisy Data
- Continual and Periodic Sampling
- Periodic Burst-Mode Sampling with Averaging

1.3.1 Noisy Data

CBD is typically very noisy (see Figure 4), comprising, for example, features not related to damage or degradation, and various types of noise, such as switching, thermal, and signal distortion. Because CBD is noisy and because characterization of useful features often requires multiple inputs of measured data, further processing and filtering is usually required: wholly internal within a 'smart' sensor, wholly external to a 'traditional' sensor, or partly internal and partly external to the collection of sensors. Said further, additional data processing is usually required to condition the raw data, to perform any necessary data and/or domain transforms, and to extract useful feature information such as condition indicators and/or precursors to failure.

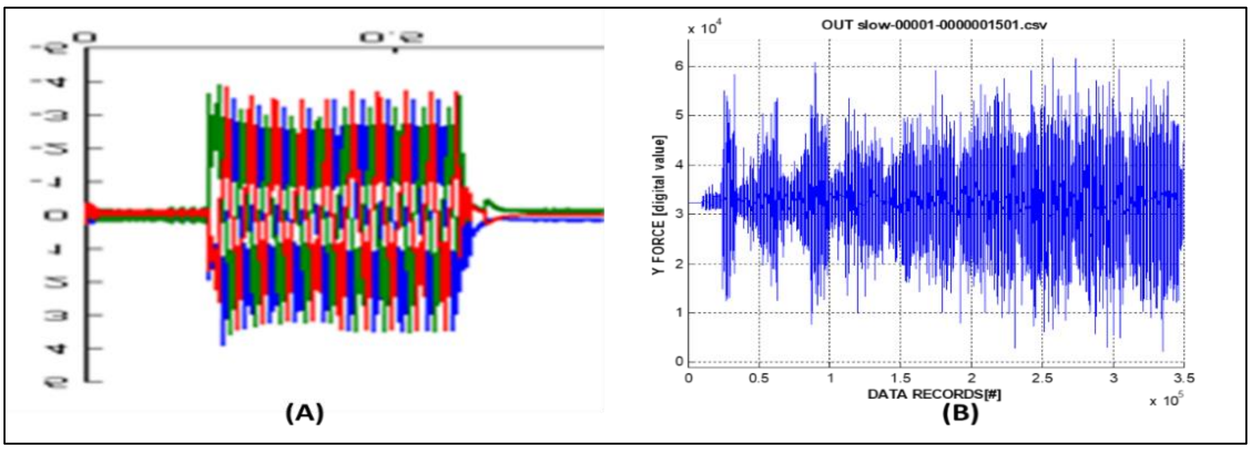


Figure 4 - Examples of noisy CBD: (A) phase currents and (B) y-direction accelerometer forces

1.3.2 Continual and Periodic Sampling

Referring to Figure 5, sampling is a form of low-pass filtering that can be continual at a frequency high enough to pass all signal frequencies or periodic at a frequency low enough to exclude high frequencies that are not of interest.in the analog signal. Sampling can also be used to reduce the amount of data that is collected, processed, and stored, which reduces the required communication bandwidth of a system.

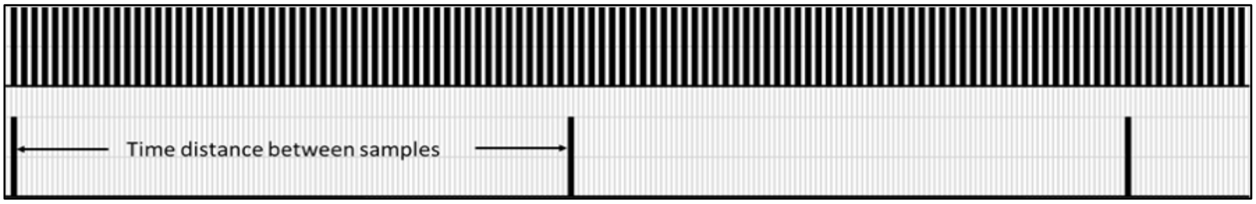


Figure 5 - Continual (top) and periodic sampling (bottom)

1.3.3 Periodic Burst-Mode Sampling with Averaging

Analog output from prognostic targets can be periodically sampled at a low rate (for example once per hour) using a high-rate burst-mode sampling (for example 200 kHz): an example.

Figure 6 illustrates relatively slow sampling at times TS and period TP combined with burst-mode sampling at a higher rate in a burst period TB. The result of each burst-mode sample is averaged to produce an FD point. This is an often-used method to capture certain high-frequency features at slower sampling rates to reduce the amount of data that is collected, processed, and saved: however high-frequency features that occur between the periodic sampling windows will not be captured.

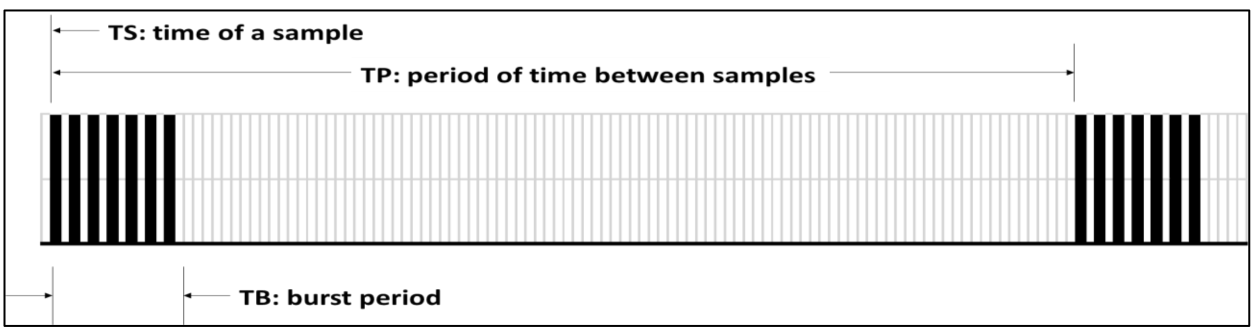


Figure 6 - Data sampling at 1/TP rate with burst mode at (s=7)/TB rate

1.4 Feature Extraction Stage: Application-Specific Data Processing

The Feature Extraction Stage accepts one or more Condition-Based Data (CBD) inputs from one or more sensors attached to one or more nodes of a system. This stage performs application-specific data processing to extract different types of Feature Data (FD) that are correlated to system degradation being caused by one or more failure modes. Application-specific data-processing methods are many, are varied, and are complex: including for example, data fusion such as fusing voltage and current data to produce resistivity data; domain transforms to convert time-domain data into frequency-domain data; and data-type transforms such as fusing temperature-dependent resistivity values with temperature data and a temperature-resistivity model to produce temperature-independent resistivity values.

Significant subtopics related to Feature Extraction are listed as follows:

- Data-Processing Methods
- Extracting FD
- Modeling FD
- FD Signatures

1.4.1 Data-Processing Methods

Extraction of FD from CBD to form signatures is accomplished by any number of application-specific, data-processing algorithms and techniques including classical model-based and data-driven such as reliability, distribution, physics-of-failure, statistical, and machine learning. Examples include the following: k-nearest-neighbor (KNN) comparisons to identify and select features of interest, such as for example, temperature-related drifts in value; distance calculations such as Euclidean or Mahalanobis to determine magnitude of changes in measurements related to damage/degradation; and physics-of-failure models to determine, for examples, changes in parameter values correlated to increasing damage/degradation. The objective of this stage is to extract FD that forms a signature comprising changes in value that are correlated to increasing levels of damage/degradation.

1.4.2 Extracting FD

An important function of the 2nd-stage is to extract useful FD from CBD: such extraction is often very complex. For example, consider the case of prognostic-enabling the output filter of a switched-mode power supply (SMPS) for a failure mode where filtering capacitance degrades as the result of successive events where parallel-connected capacitors successively fail short and burn open. The selected node to be monitored is the output of the SMPS: see Figure 7 and Figure 8.

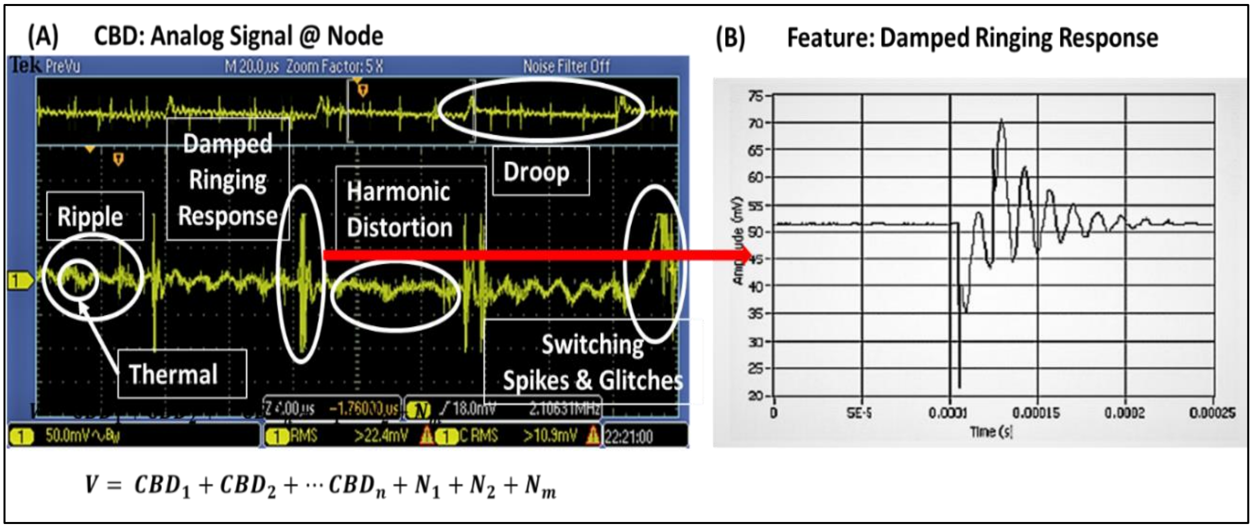


Figure 7 - SMPS output: six extractable features (A) and an extracted damped-ringing response (B)

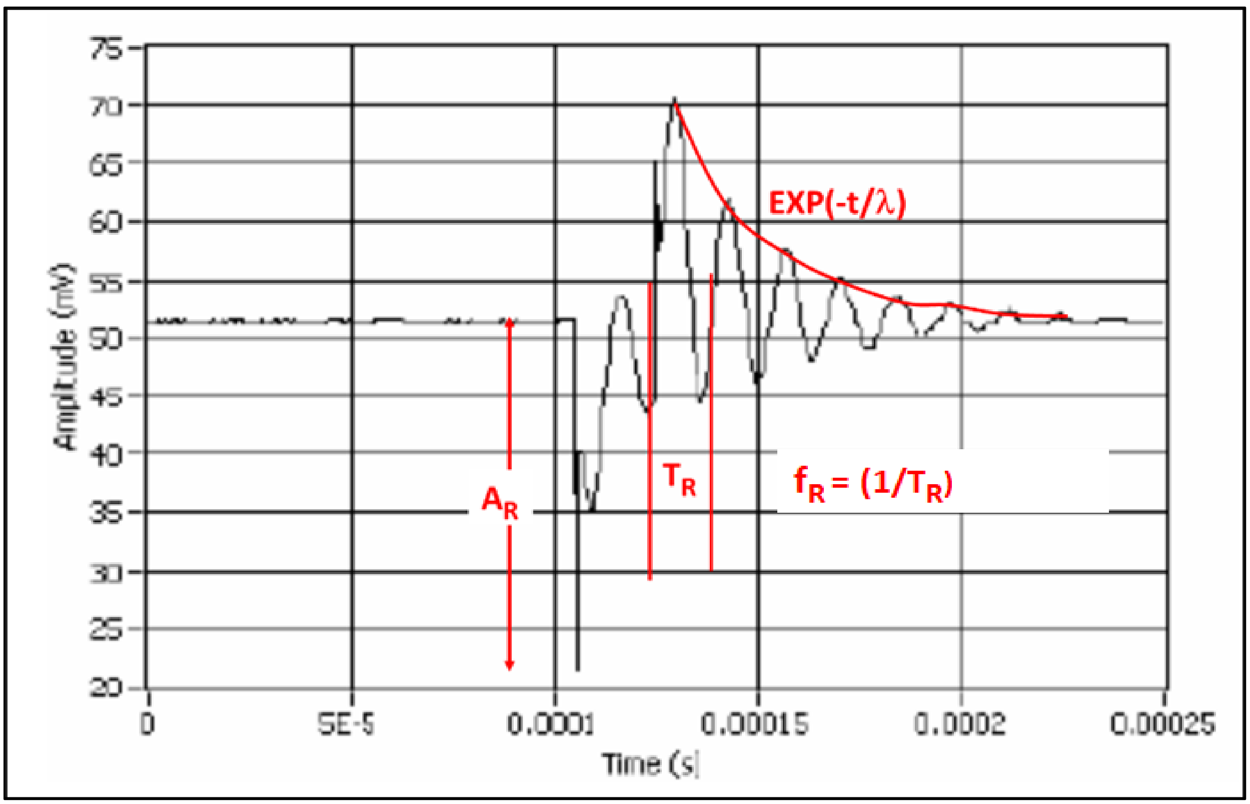


Figure 8 - Idealized damped-ringing response showing other extractable features

Figure 7 (A) is an analog capture of a sensor output connected to an SMPS output node identifying six extractable features that could be used as precursors to failure related to loss of filtering capacitance in the output filtering circuit. After examining Figure 7 (A), performing physics-of-failure (PoF) analyses and modeling, the damped-ringing response is chosen as an extracted feature, Figure 7 (B), which, in turn, contains other extractable features (Figure 8) that also could be used as precursors to failure related to loss of filtering capacitance. For example: the amplitude (AR) is a function of the gain in the feedback loop and a function of the impedance of the filter, which includes the filtering capacitance; the resonant frequency (fR) and period (TR) are functions of the inductance and capacitance of the output filter, and the exponential lifetime value (·) is a function of the resistance and capacitance of the output filter.

1.4.3 Modeling FD

Physics of Failure (PoF) analyses includes modeling, which is used to select the resonant frequency as the FD. That selection is based on a failure effects analysis that indicates that changes in resonant frequency are primarily due to changes in capacitance rather than, for example, changes in loading and temperature. Such modeling includes the following:

- Multiple features (CBD) and noise (N)

  $$\boldsymbol{V = CBD_1 + CBD_2 + \cdots CBD_n + N_1 + N_2 + N_m} \quad \textbf{(1)}$$

- Damped-ringing response (a CBD feature)

  $$\boldsymbol{V_0 = V_{DC} + A_R\{exp(-t/\lambda)\}\{cos(\omega t + \phi)\}} \quad \textbf{(2)}$$

- Natural resonant frequency (a feature of a damped-ringing response):

  $$\boldsymbol{\omega_0 = \sqrt{A_R + 1}\,(1/\sqrt{LC})} \quad \textbf{(3)}$$

- Circuit quality:

  $$\boldsymbol{Q = \sqrt{A_R + 1}\,(1/R)(\sqrt{C/L})} \quad \textbf{(4)}$$

- Measurable frequency:

  $$\boldsymbol{\omega = \omega_0\sqrt{1 - 1/(4\,Q^2)}} \quad \textbf{(5)}$$

- Simplify frequency model using substitution, assuming a high value of Q (≥ 10), and using insignificant variations in the values of A, R, and L compared to changes in value of C due to degradation:

  $$\boldsymbol{\omega \approx \omega_0\sqrt{(C0/(C0 - dC))}} \qquad \textit{C0 is the base value of C} \quad \textbf{(6)}$$

- Feature data:

  $$\boldsymbol{FD = \omega/2\pi} \quad \textbf{(7)}$$

1.4.4 FD Signatures

Referring to Figure 9, in the absence of degradation, FD data forms an essentially flat signature (bottom left part of the plot, and in the presence of degradation, FD data forms a curvilinear signature (upward curving part of the plot. Figure 10 identifies four types of noise present in the signature.

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

1.5 Prognosing Stage

The ARULE Adaptive Prediction Kernel (APK) used in the 3rd prognosing stage shown in Figure 2 comprises two subsystems: Signatures and Prognostics. The Signatures subsystem conditions and linearizes the input FD signature, and the Prognostics subsystem uses prediction, adaptation, and other computational routines to produce accurate prognostic information such as RUL, SOH, and PH.

Significant subtopics related to the Prognosing Stage are listed as follows:

- Signatures Subsystem
- Prognostics Subsystem

1.5.1 Signatures Subsystem

The primary function of the Signatures Subsystem of ARULE is to transform FD signatures into other signatures at the request of a feature extraction stage. The signature transform is done sequentially, one data point at a time, as opposed to batch-mode processing of an entire data set: this architecture supports near real-time prognostic results. To accommodate robust extraction and/or robust data processing, ARULE lets a feature extraction stage specify the following: (1) the step at which signature processing is to begin and (2) parameter values to be used in signature processing. The steps taken by the Signature Subsystem of ARULE are the following:

1. Transform FD signature data into Fault-to-Failure Progression (FFP) signature data.
2. Transform FFP signature data into Degradation Progression Signature (DPS) data.
3. Transform DPS data into Functional Failure Signature (FFS) data.

Secondary functions of the Signatures Subsystem are to perform, by request, noise mitigation such as data smoothing and noise margins.

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

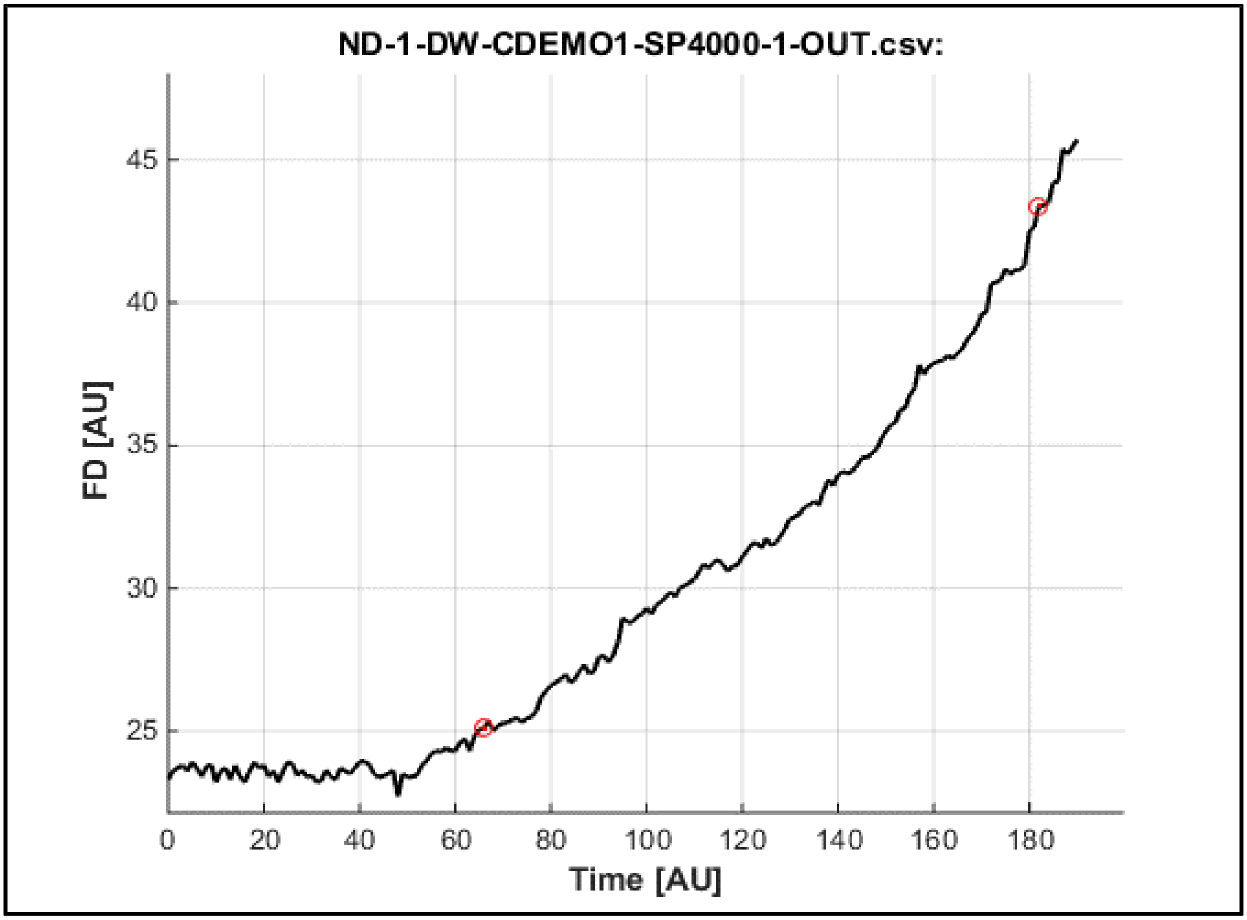


Figure 9 - Example of an FD signature (resonant frequency)

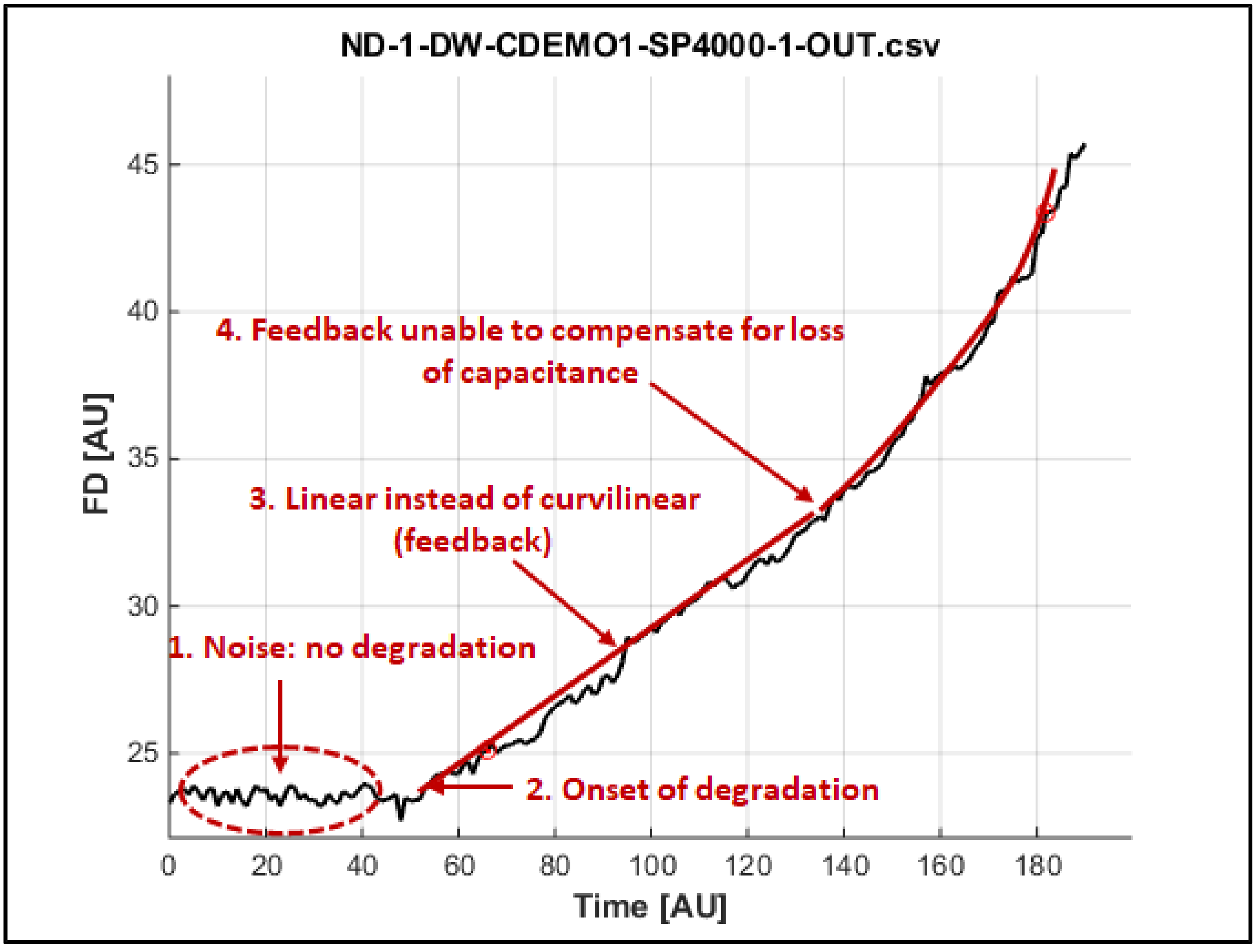


Figure 10 - Annotated FD signature: noise, degradation, linear degradation (feedback), and curvilinear degradation (feedback no longer effective)

1.5.2 Prognostics Subsystem

The function of the Prognostics subsystem of ARULE is to accept and process FFS data to produce prognostic information: RUL, SOH, and PH. The Prognostics Subsystem is a fast, accurate prognosing APK that uses several algorithms and methods to rapidly and accurately converge to prediction of a true time of functional failure: identifying the point in time where a prognostic target is no longer capable of operating within specifications. It uses sophisticated, yet fast and efficient, algorithms and methods to process input data and produce estimates of prognostic information: one data point at a time.

2. ARULE: Description and Operation

Referring to Figure 11, ARULE receives Feature Data (FD) input and outputs prognostic information one data point at a time.

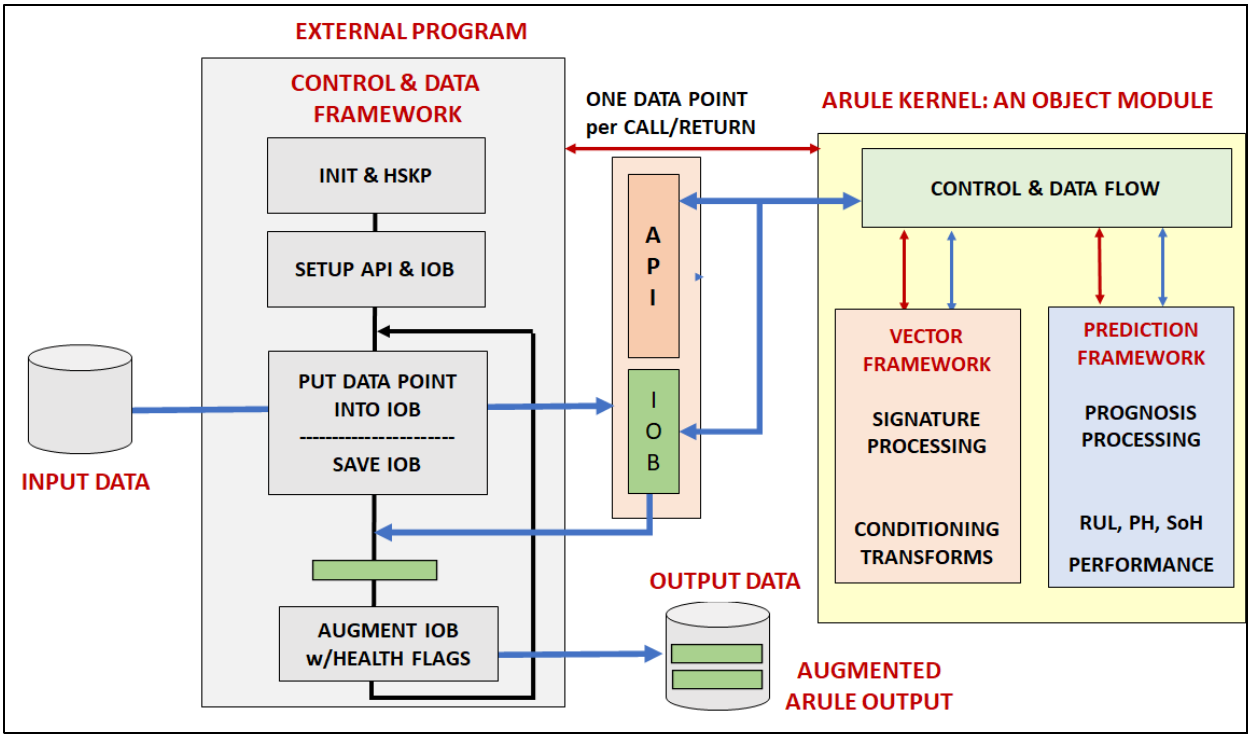


Figure 11 - Operational diagram: external program using ARULE through its API

ARULE can be applied to a wide range and size of systems and subsystems: including electrical, mechanical, and electro-mechanical systems when there is condition-based sensor data and feature data correlated to aging or degradation. ARULE correctly deals with aging, degradation, and recovery effects to produce prognostic predictions that are very accurate: the number of input data points needed for ARULE to reach an accurate solution is dependent on how noisy the data is and the amount of difference between a system-assigned value for expected time-to-functional failure (TTFF), or end-of-life (EOL) after onset of degradation and the actual TTFF. For example, benchmark testing using low-noise data shows ARULE needs only 2 to 3 data points to converge within a 10% accuracy margin, meaning that the estimate is more than 90% accurate for a low-noise data situation and small difference in TTFF values. For high-noise and large difference in TTFF values, ARULE typically needs only between 15 to 30 data points to converge to within 10% accuracy margin.

2.1 Graphical User Interface Diagram

The ARULE Graphical User Interface (GUI) was designed and implemented as example of the "External Program" referenced in Figure 11. The underlying Adaptive Predictive Kernel (APK) within ARULE is a command line interface (CLI) program that can be embedded or called by an "External Program" such as the ARULE™ GUI. The CLI utilizes a flexible API, and Ridgetop offers engineering services to create customized installations for real–time system health monitoring. This User Guide covers the ARULE™ GUI implementation that links each of the PHM frameworks shown in Figure 12.

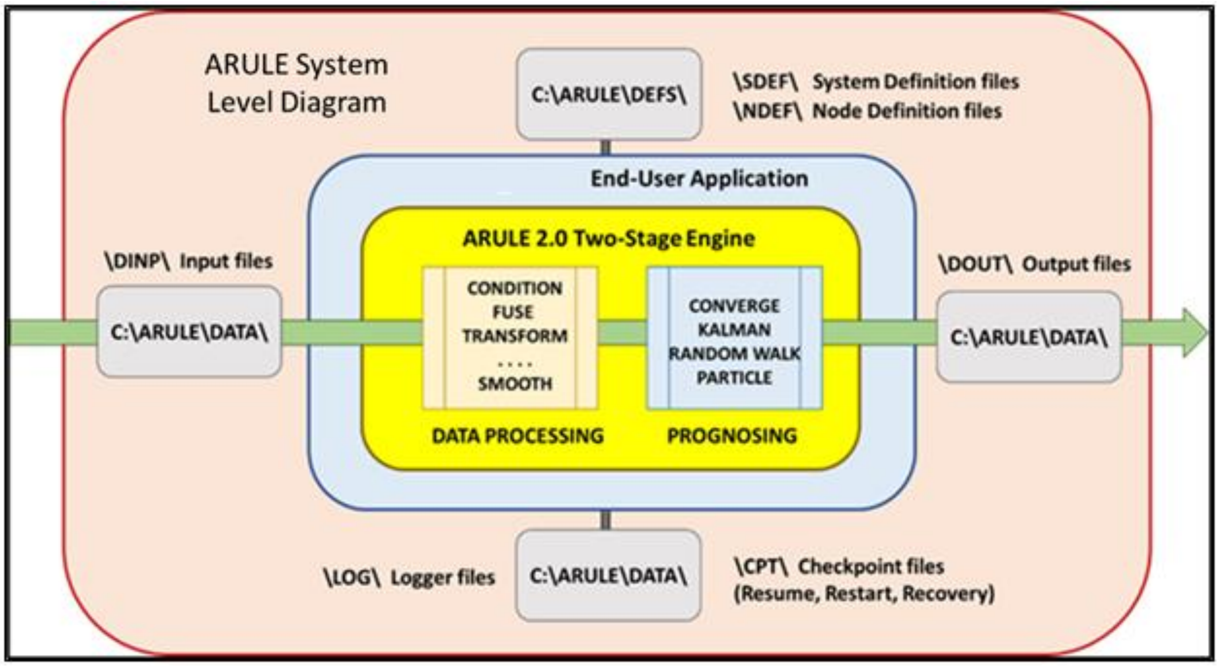


Figure 12 – Framework for a GUI–based ARULE™

2.2 Definition Files

The processing and control of ARULE™, including the ARULE™ Adaptive Prediction Kernel (APK), is driven by tables created from definition files: those tables are specified as editable files instead of programmed tables; they are called definition files and are described as follows:

- **System Definition (SDEF) Files** – Define the number of nodes in a system, and points to a corresponding node definition file.
- **Node Definition (NDEF) Files** – Defines the path for the input sensor data file, along with the user defined parameters and values required to process the input data with the ARULE™ APK. **Table 1** shows the complete list of NDEF parameters.

When an SDEF file and the corresponding NDEF files are created, the program can process the data input to generate key prognostic estimates for SoH, RUL, and PH. The generated data output files, log files, and system files support checkpoint/restart and recovery from unexpected end–of program events. The ARULE™ GUI also supports creating, editing, and visualization of the definition files, as well as plotting capabilities to preview the data output results.

Table 1. List of Node Definition (NDEF) Parameters

ID NAME		DESCRIPTION	VALUE: TYPE, RANGE, and DEFAULT
1.	**FDZ**	Specifies a nominal FD value to use when transforming FD into FFP signature data: FFP = [FD – FDC*(1 + FDNM/100)]/FDZ	• Any positive value • 0 means use FDC value
2.	**FDC**	Specifies a nominal FD starting value to use in the absence of degradation/damage and a moving average is used to calculate FDC based on FDCPTS.	• 0.0 to any positive value • 1.0 (default)
3.	**FDCPTS**	Specifies the number of pts to calculate the moving average of FDC.	• 0 to 25.0 • 5.0 (default)
4.	**FDPTSf**	Specifies number of data points as integer to average for FD.	• 1 to 5 • 5 (default)
5.	**FDNM**	Specifies a noise margin; percent of the nominal value of FDC to use as a margin to mitigate noise	• 0 to 25.0 • 5.0 (default)
6.	**FDNV**	Specifies the n or lambda value for transforming FFP signature data to DPS signature data	• Any non-zero positive value • 1.0 (default)
7.	**FFPFAIL**	Specifies a percentage of FFP to use as a threshold value for functional failure: FF = FFP/(FFPFAIL/100)	• Any non-zero positive value • 70.0 (default)
8.	**PITTFF**	Specifies an initial value to use for time-to-functional failure (TTFF): becomes the RUL value in the absence of degradation	• Any non-zero positive value • 200.0 (default)
9.	**PIFFSMOD**	Specifies the expected shape of an FFS curve for EKF modeling: 1: concave, 2: linear, 3: convex, 4: convex-concave, 5: concave-convex;	• 1 to 5 • linear (default)

2.3 Directory Structure: Description and Contents

ARULE™ GUI files are installed in the default windows directory as outlined below:

C:\Program Files\ARULE		**Executable subdirectory**
1.	DEPENDENCIES\	ARULE™ GUI Dependencies
2.	DOCS\	User Guide Document & Product Brief
3.	ARULE.exe	ARULE Executable Program

Other ARULE™ program files such as the input data files, output data files, and the system and node definition files are kept in a user specified directory selected when the software is first installed. There are two (2) main directories and seven (7) subdirectories that are installed in the user specified location. An example of how these subdirectories are organized in the default installation (**C:\Users*username*\My Documents\ARULE**) is shown below.

...\ARULE\DATA		**Data set of subdirectories**
1.	CPT\	Checkpoint Files
2.	DINP\	Data Input Files
3.	DOUT\	Data Output Files
4.	LOG\	Logger Files
5.	SAVE\	Default Save/Download Directory
...\ARULE\DEFS		**Definition set of subdirectories**
6.	NDEF\	Node Definition Files
7.	SDEF\	System Definition Files

2.4 BNF for Definition Files

The ARULE™ GUI offers intuitive user forms to edit definition files, but it is important to note for freeform editing purposes, ARULE™ uses a simple BNF (Backus–Naur Form or Backus–Normal Form) for creating definition files:

- Input records are a maximum of 80 characters in length: including the new line character that ends the record.
- A percent sign (%) character starts a comment: it and all other characters that follow are ignored.
- Continuation of one record to the next is not supported.
- Data format is operand operator operand:

    - The first operand is a parameter keyword (such as NDNUMID).
    - There is only one operator – the equal sign (=) character.
    - The second operand is a keyword value to be assigned to the parameter keyword.
- Apostrophe (‘) characters are used to delimit the beginning and ending of string type of keyword values and are required when a string includes white space type of characters. To avoid errors in usage, the ARULE™ GUI automatically inserts apostrophes for you – do not enter them yourself.
- A semicolon (;) character to delimit a keyword value is automatically inserted by the ARULE™ GUI – so do not enter a semicolon yourself.

2.5 ARULE: Signature Processing

Referring back to Figure 10, ARULE comprises data algorithms and methods that have been incorporated into a Feature Vector Framework that processes signature data and a Prediction Framework that produces prognostic information. For each input data, ARULE produces a set of prognostic information before another data point is processed: ARULE uses data models, signature transforms, Extended Kalman Filtering, geometry-based computations, and various linearization and machine learning to produce fast, efficient, and accurate estimates of RUL, PH, and SoH. To illustrate and describe ARULE, the definition values for FDZ and FDNV shown in Figure 14 were changed respectively from 24.0 to 0.0 and from 2 to 1.275.

```
%*******************************************************************
% NODE1FULL: Full NDEF definition equivalent to CNODE1
%*******************************************************************
%**Feature Data: FD = FDZ*(dP/P)^FDNV + DC + NOISE
FDC       = 24.0;   % Feature Data (FD), DC Value
FDZ       = 24.0;   % Nominal AC value, when 0 use FDC value
FDNM      = 5.0;    % Noise margin - percent of the value of FDC
FDCPTS    = 10;     % data points to average for FDC: up to 25
FDNV      = 2;      % Degradation power n
%**Prognostic Modeling
FFPFAIL   = 70.0;   % Failure margin - percent above nominal
PITTFF    = 200.0;  % Default RUL = TTFF value
PIFFSMOD  = 2;      % model 1=Convex, 2=Linear, 3=Concave,
%                    4=convex-concave, 5=concave-convex, 6=convex-concave
%**       File Dependent Parameters
INFILE    = 'SP4000_1';  % Input file name, _OUT  appended for output
INTYPE    = '.txt' ;    % also .csv Input file type
OUTTYPE   = '.csv' ;    % also .txt Output file type
%*******************************************************************
ENDDEF    = -9;     % end of node definition
```

Figure 13 - An example instantiation of defined set of API keywords and values to use the ARULE kernel

2.5.1 Transforming an FD Signature into a Fault-to-Failure (FFP) Signature

ARULE conditions and transforms input FD signature data into Fault-to-Failure Progression (FFP) signature data:

$$\boldsymbol{FD_i = DC_i + N_i + FD_0(f(dP, P))} \tag{8}$$

And ideal set of FFP is defined as,

$$\boldsymbol{FFP_i = FD_i / FD_0} \tag{9}$$

Referring back to Figure 13, specify the following.

$$FDC \geq \max(\mathrm{DC_i})\,, \qquad (FDNM/100) \geq \max(N_{\mathrm{i}})/100, \;\; FDZ = \mathrm{FD_0}$$

$$FFP_i = (-FDC - NM)/FDZ + f(dP_i, P) \text{ [\textbf{from (8) and (9)}]}$$

$$FFP_i = f(dP, P) - c \textit{ and when } c \ll max(f(dP_i, P))$$

$$\boldsymbol{FFP_i \leq f(dP_i, P)} \tag{10}$$

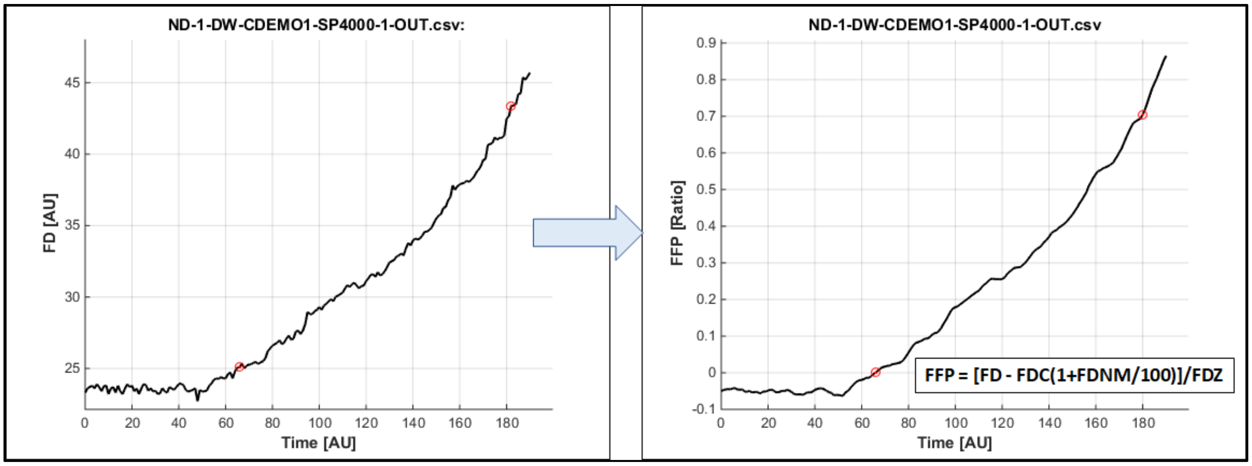


Figure 14 - FD signature data transformed into an FFP signature data

Note the following about the FD to FFP transform shown in Figure 14 :

1. The data has been shifted from a non-zero, application-specific reference of about 24 to a generic zero reference:
   - A nominal FD value was determined by averaging input data in the absence of degradation using values specified by FDC, FDCPTS, and FDCVAR.
   - A noise-margin value determined by a percent value of FDC as specified by FDNM
2. The data has been smoothed to allow specification of a smaller FDNM value to reduce the magnitude of offset distortion introduced by a noise-margin shift of the data.
3. Division of the data by the value of FDZ transforms the Y-axis values from units-of-measure to generic ratios, which reduces modeling complexity.

2.5.2 Transform FFP Signature Data into DPS Data

Modeling complexity and use is reduced in ARULE by use of a single model for DPS data based on the inherent inaccuracies associated with processing CBD that is noisy, distorted, digitized, and discretely sampled.

$\boldsymbol{DPS} = (\boldsymbol{dP_i}/\boldsymbol{P})$ **where P represents a parameter that degrades** **(11)**

$\boldsymbol{f}(\boldsymbol{dP_i}, \boldsymbol{P}) = (\boldsymbol{dP_i}/\boldsymbol{P})^{\boldsymbol{n}}$ ***power of n function*** **(12)**

Then by **(11) and (12)**,

$$\boldsymbol{DPS} = (\boldsymbol{FD_i}/\boldsymbol{FD_o})^{\boldsymbol{1/n}} = (\boldsymbol{FFP_i})^{\boldsymbol{1/n}} \quad \textbf{(13)}$$

2.5.3 Significance of the FFP-to-DPS Transform

Because (dP_i/P) is a derivation (the rate of change of P to P), transforming FFP signature data into DPS data linearizes a curvilinear signature (Figure 15).

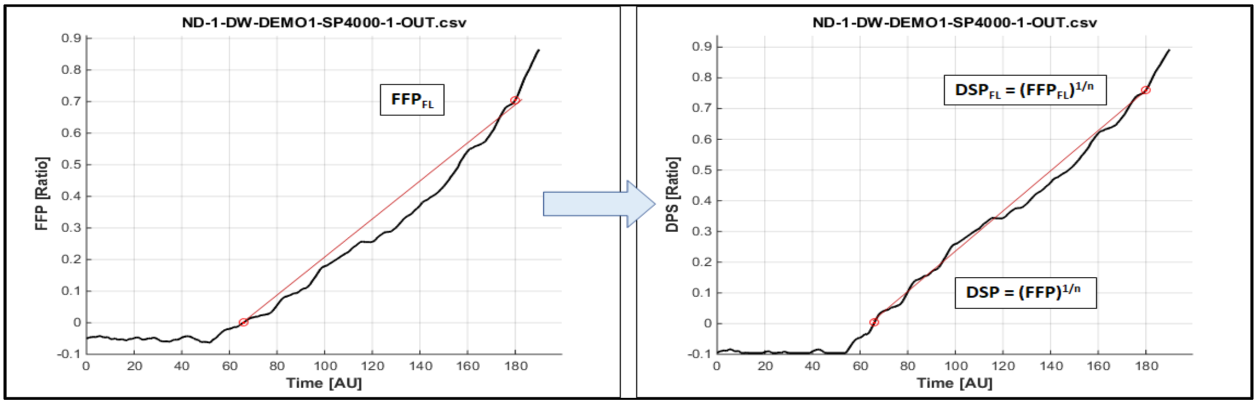


Figure 15 - FFP signature data transformed into DPS data

Referring back to Figure 15, note the following about the FFP to DPS transform:

4. Negative FPP values are transformed to zero (0): no detectable degradation.
5. The curvilinear FFP data has been linearized,

2.5.4 The FFS-to-DPS Model

Figure 16 contains ideal FFP plots for selected values of n (left) and the corresponding DPS plots: an ideal straight line.

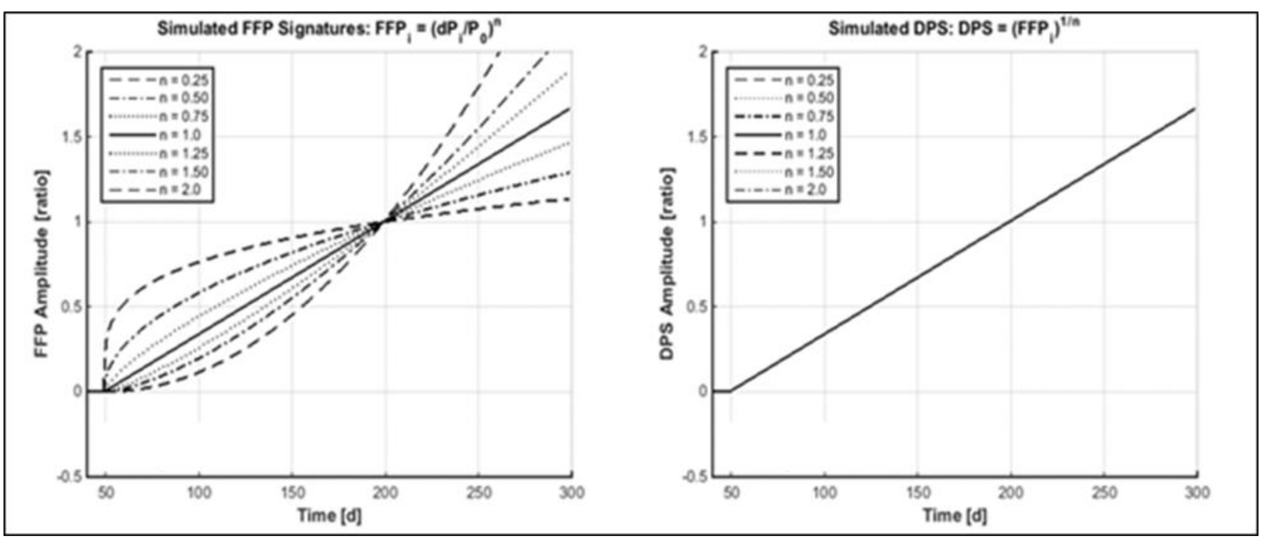


Figure 16 - Power Function: FFP on the left, DPS on the right

2.5.5 Transform FFP to DPS Example: Modeling and Specification

The DPS to FFS transform comprises the following steps: (1) obtain experimental and/or simulation data for the failure mode to be prognosticated; (2) select a beginning and ending point on the curve – above the noise level and below a level of functional failure; (3) determine the characteristic-signature curve of the failure mode – convex, linear, or concave; (4) and select the power value (see Figure 16 and Figure 17); (5) use the FDNV keyword to specify the value; (6) process the input FD; and (7) verify, validate, and change the specified FDNV to obtain a desired/reasonable level of DPS linearity (see Figure 18).

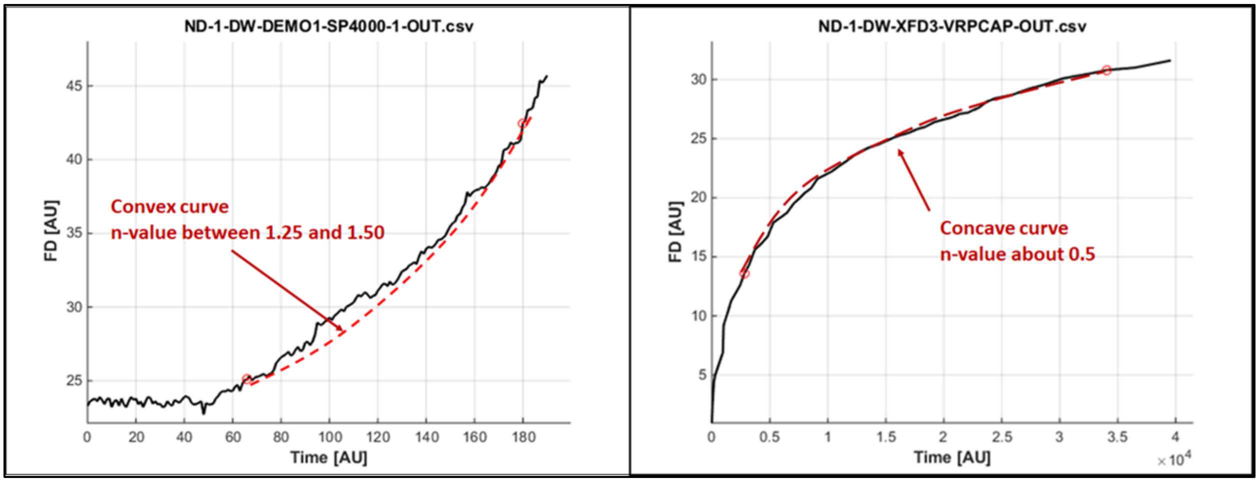


Figure 17 - Examples of FD input, their characteristic curves, and initial evaluation of probable n values

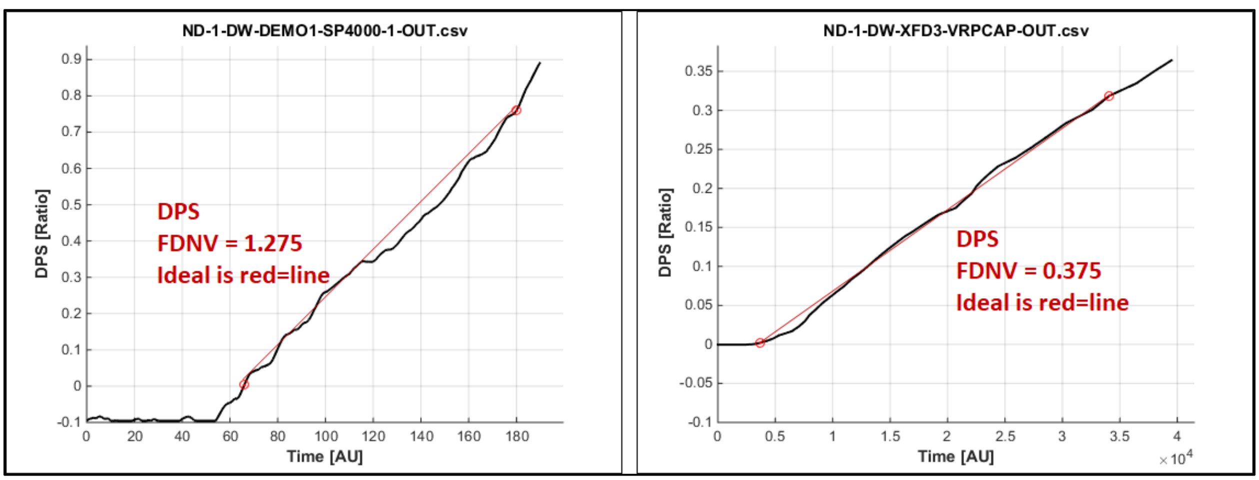


Figure 18 - Examples of DPS transforms for specified FDNV values for the FD input in Figure 17

2.5.6 Transforming DPS Data into Functional-Failure-Signature Data

A failure, as it relates to prognostics, is often ill-defined. For example, does failure occur when a drive shaft shears, or when the friction between the shaft and a roller bearing reaches a level where torque on the shaft exceeds design specifications, or when the shaft becomes warped and the unit begins to vibrate?

ARULE predictions are based on functional failure rather than simply failure or physical failure. Functional failure occurs when a prognostic target (a device, component, subassembly, and so on) is no longer capable of operating within specifications. Functional failure, therefore, occurs before physical failure or operational failure: much like a tire that is thread bare but still fully inflated.

The FFPFAIL keyword defines the FFP threshold at which functional failure occurs. ARULE autonomously transforms that value into its equivalent DPS value which transforms that value into an FFS value of 1, which is then transformed into a value of 100 percent (Figure 19):

$$\boldsymbol{FFS_i = 100\ (DPS_i / FL_{DPS})} \qquad \textbf{(14)}$$

Significant results of the transform are the following:

1. The DPS y-ratio values are transformed into FFS y-percent values.
2. The FFS y-values mean the following:
   a. FFS value = 0 -> degradation is not detectable.
   b. FFS value >= 100 -> functional failure has occurred
   c. FFS value between 0 and 100 -> degradation is detected

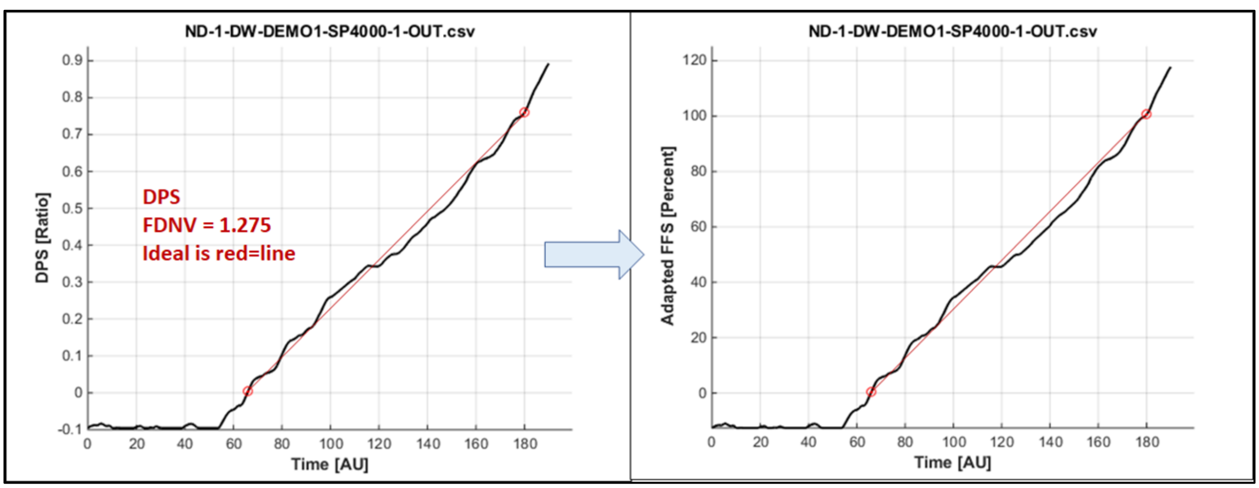


Figure 19 - DPS data transformed into FFS data – ideal transforms are red-colored lines

2.6 ARULE: Prognostic Processing and Filtering Methods and Results

ARULE employs Extended Kalman Filtering (EKF) in that input FFS data is compared to two sets of models. A data point is compared to a first, SoH model, then the model is adapted to the data point, and then the data point is adapted to the model with the adaptation biased in favor of a pessimistic level of SoH: the adaptation supports both gradual changes in health caused by increasing degradation and rapid changes in health caused by abrupt changes in health. The adapted data point is then compared to a second, RUL model, the model is adapted to the data point, then the adapted RUL model is used to project a future time of failure (Figure 20).

1. As though data points are particles that exhibit properties of momentum and inertia.
2. Data points travel in random-walk fashion through three data spaces as illustrated in Figure 21.

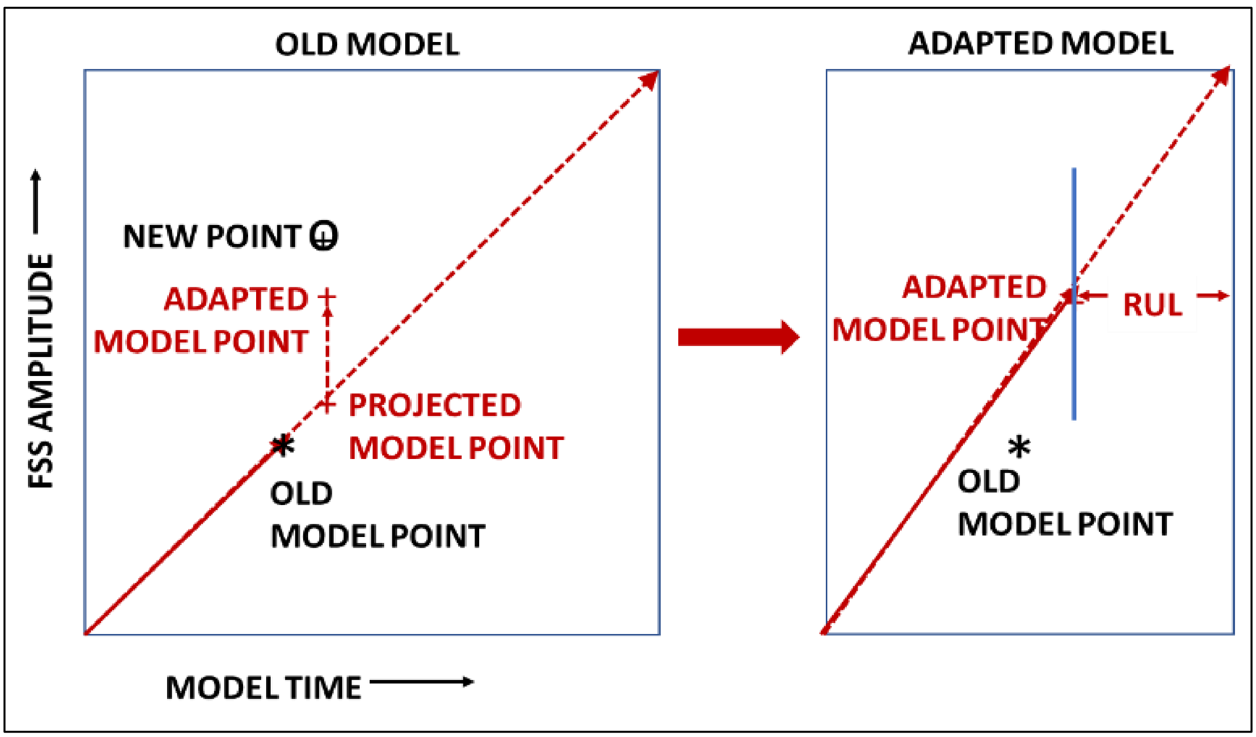


Figure 20 - EKF: adapt data point to 1st model, adapt 2nd model to adapted data, then use adapted 2nd model to predict RUL, PH, and SoH.

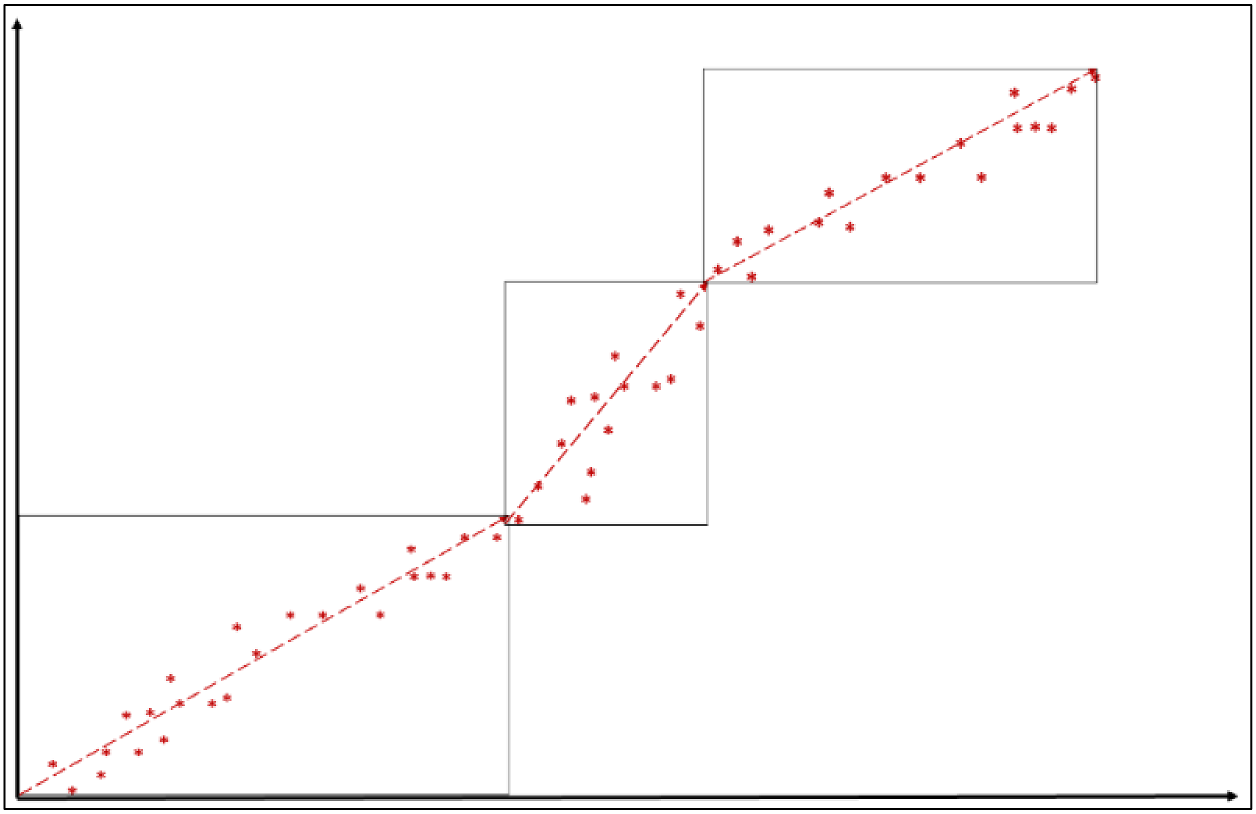


Figure 21 - Piecewise-linear 2nd EKF model: random-walk of data through multiple data spaces

2.6.1 Second Model for EKF Processing

ARULE supports a PIFFSMOD keyword (**Table 1**, page 15) to specify one of six different models to be used as the final model for EKF processing. After conditioning and transforming, FFS data will exhibit a characteristic shape resembling one of those models: specify that model.

2.6.2 Estimated Prognostic Information

ARULE conditions and then partially adapts a first data model to FFS data, then adapts FFS data to the partially adapted first data model; then adapts a final model to the adapted FFS data and uses the adapted second model to estimate a future time of failure to produce prognostic information: RUL, PH, and SoH (Figure 24). Notice the very linear and accurate estimates of SoH.

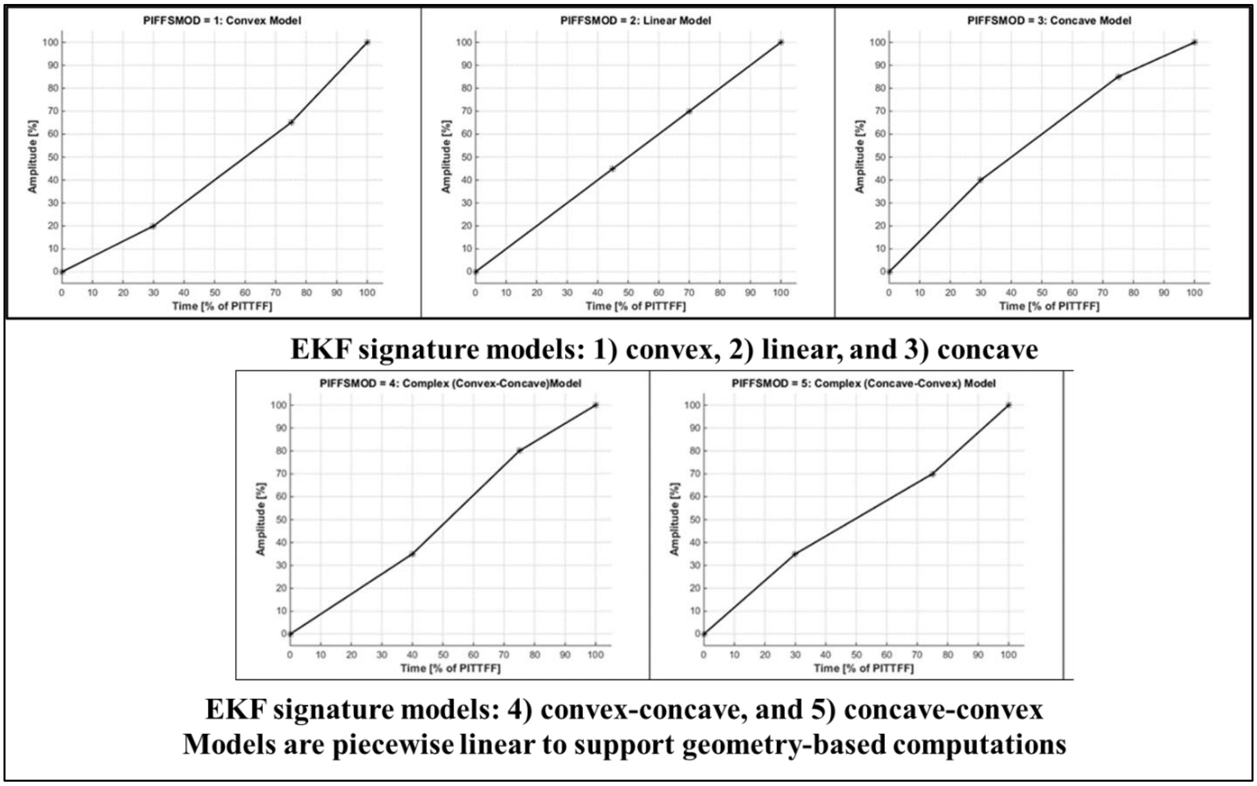


Figure 22 - Plots of the final signature models specified by the PIFFSMOD input parameter for ARULE to perform EKF processing of FFS data

2.6.3 Examples of Selecting and Specifying the 2nd EKF Model

The PIFFSMOD = 2 specifies final EKF processing is to use a linear model (Figure 22, top row, middle) and the final results are shown in Figure 23 (right-hand plots).

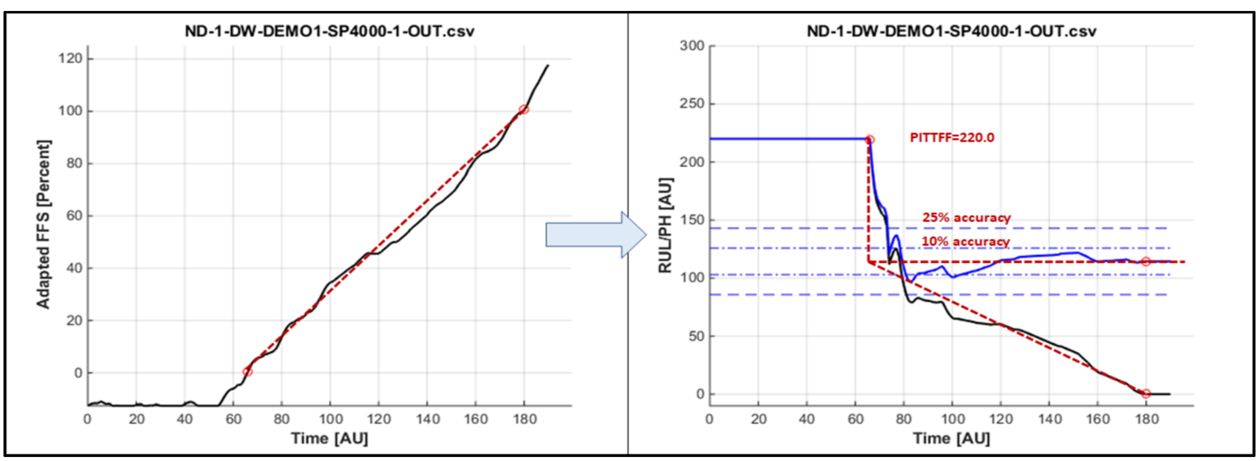


Figure 23 - Adjusted FFS plot (left) of the transformed input to final prediction stage (Extended Kalman Filtering) and plots of the resulting RUL/PH estimates.

2.6.4 Example Plots: Prognostic Estimates

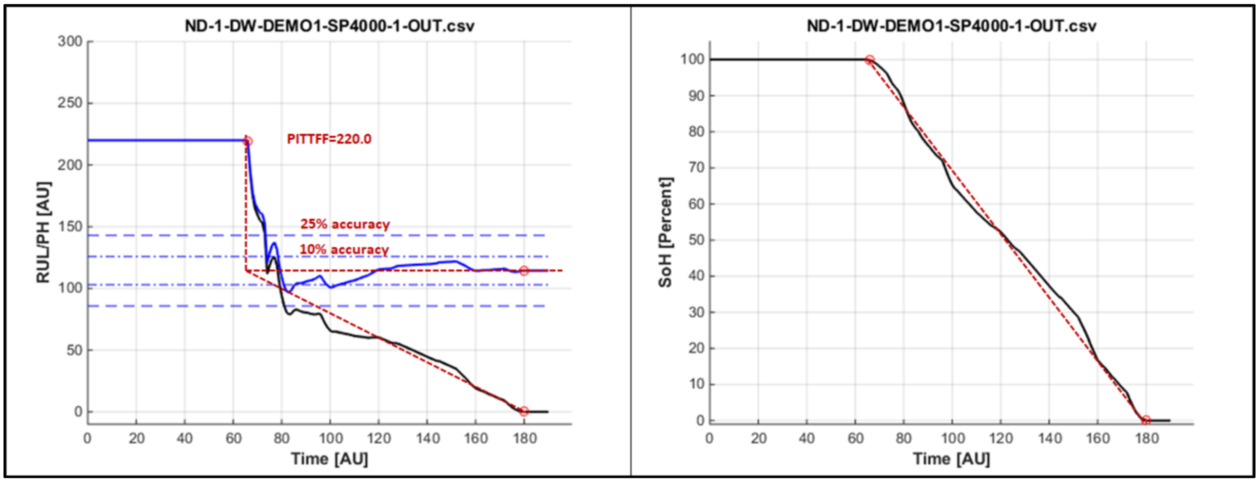


Figure 24 - Plots of RUL/PH estimates (left) and SoH estimates (right): 25% &10% margins of accuracy (blue dot-dashed lines) and ideal estimates are included (red-dashed lines).

3. Prognostic Information: Evaluation of Accuracy

The purpose of prognostics is to detect degradation and create prognostic information such as estimates of state-of-health (SOH) and remaining useful life (RUL) of systems for the following benefits: (1) reduce operation costs, (2) provide advance warning of failures; (3) minimize unscheduled maintenance; (4) predict the time to perform preventive replacement; (5) increase maintenance cycles and operational readiness; (6) reduce sustainment costs by decreasing inspection, inventory, and down-time costs; and (7) increase reliability by improving the design and logistic support of existing systems.

3.1 Ideal Prognostic Information

Ideally, every instantiation of a prognostic target will fail in the same amount of time between the onset of degradation and the time when functional failure occurs; and there will be zero noise in the system; and the prognostic monitoring system will detect degradation exactly when it first occurs and will detect functional failure exactly when it occurs; and a prognostic engineer species the exact value to use as an estimated RUL value prior to the detection of degradation, enabling the PHM system to produce ideal RUL and PH values such as those plotted in Figure 25.

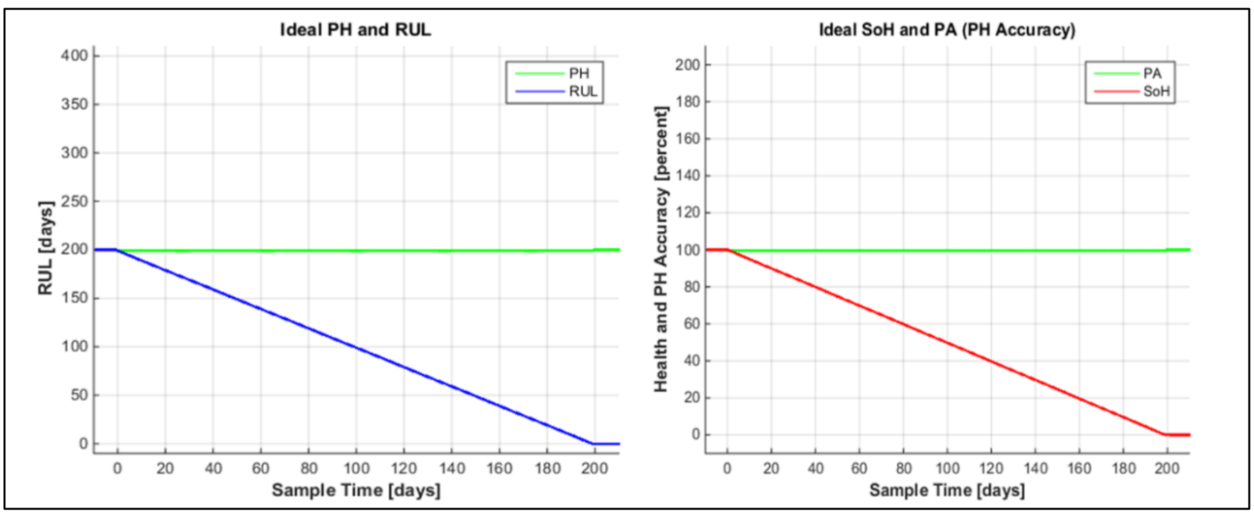


Figure 25 - Ideal plots of RUL, PH, and SOH values with 100% prognostic accuracy

3.2 Idealistic, But Not Exact, Prognostic Information

Realistically, ideal plots cannot occur because, if nothing else, prognostic targets have a failure distribution. Referring to Figure 26, some targets have a fast rate of failure, some have a slow rate of failure, and the rest fail between the fastest and slowest. A temporizing approach is to specify some mean value, such as MTTF (mean-time-to-failure), as a Predicted Time-to-Functional-Failure (PITTFF) value.

Given a PITTFF value that is 50% higher or lower in value compared to a true time of failure, then in the absence of noise in the data (anything not directly related to a failure mode of interest), ARULE produces prognostic information such as the idealized plots shown in Figure 27. Those plots illustrate the following: (1) fast convergence from initial-estimate values to true time of failure and (2) ability to converge from both high- and low-value initial estimates to very accurate estimates. But there are other inaccuracies that occur because of noisy data. Plots of RUL and PH are more likely to resemble one of the two non-ideal plots shown in Figure 28 rather than Figure 27.

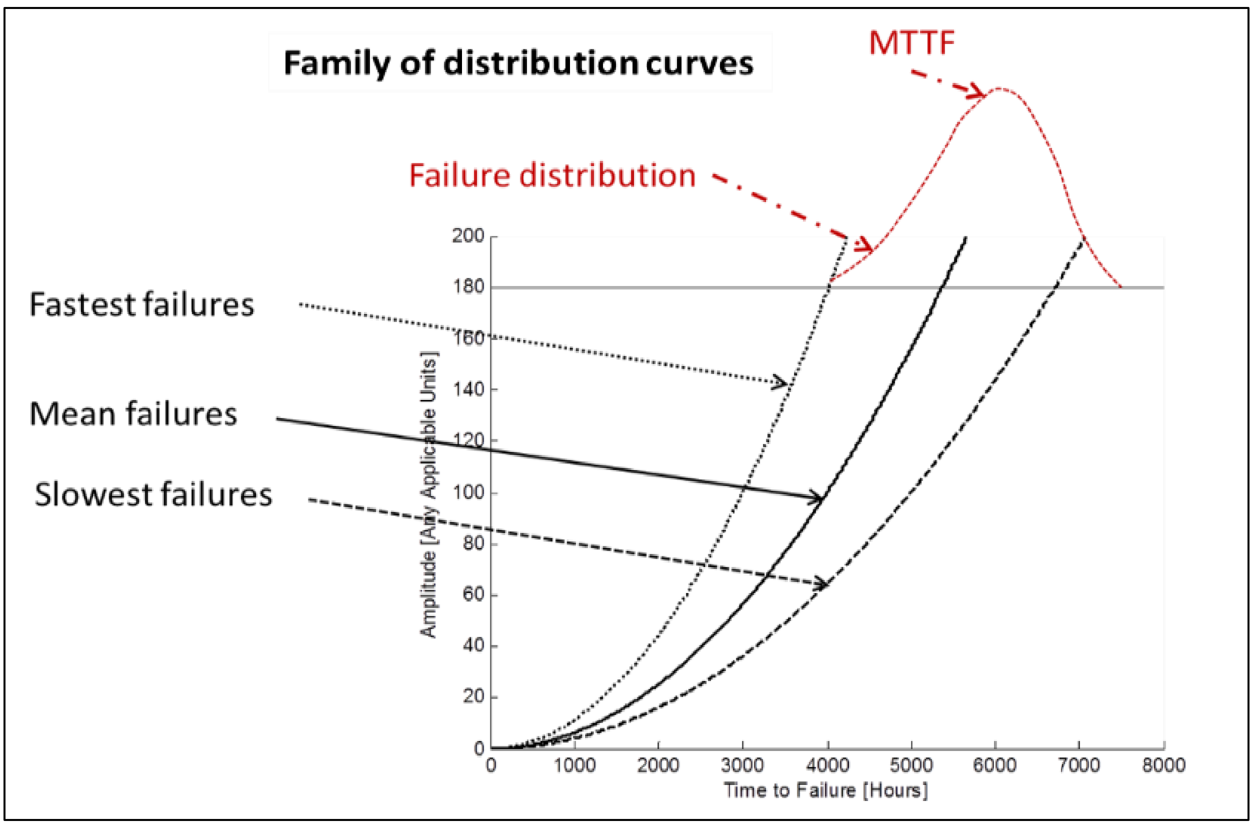


Figure 26 - Example of a failure distribution

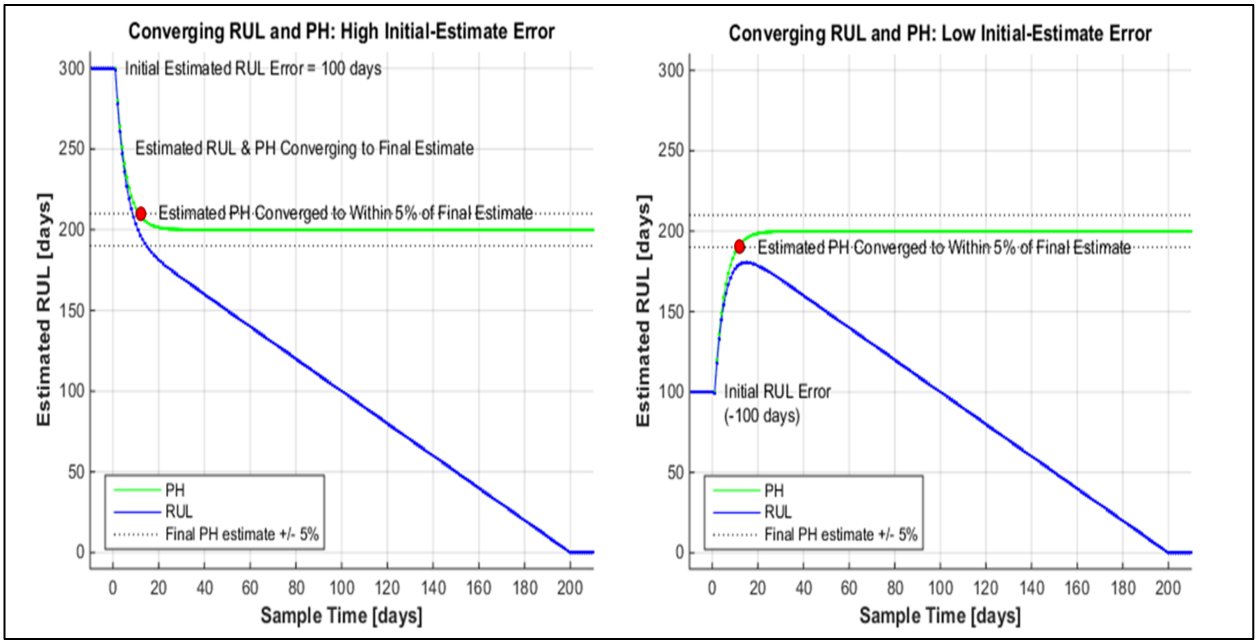


Figure 27 - Idealized plots of RUL and PH estimates given initial values having +/- 50% variation

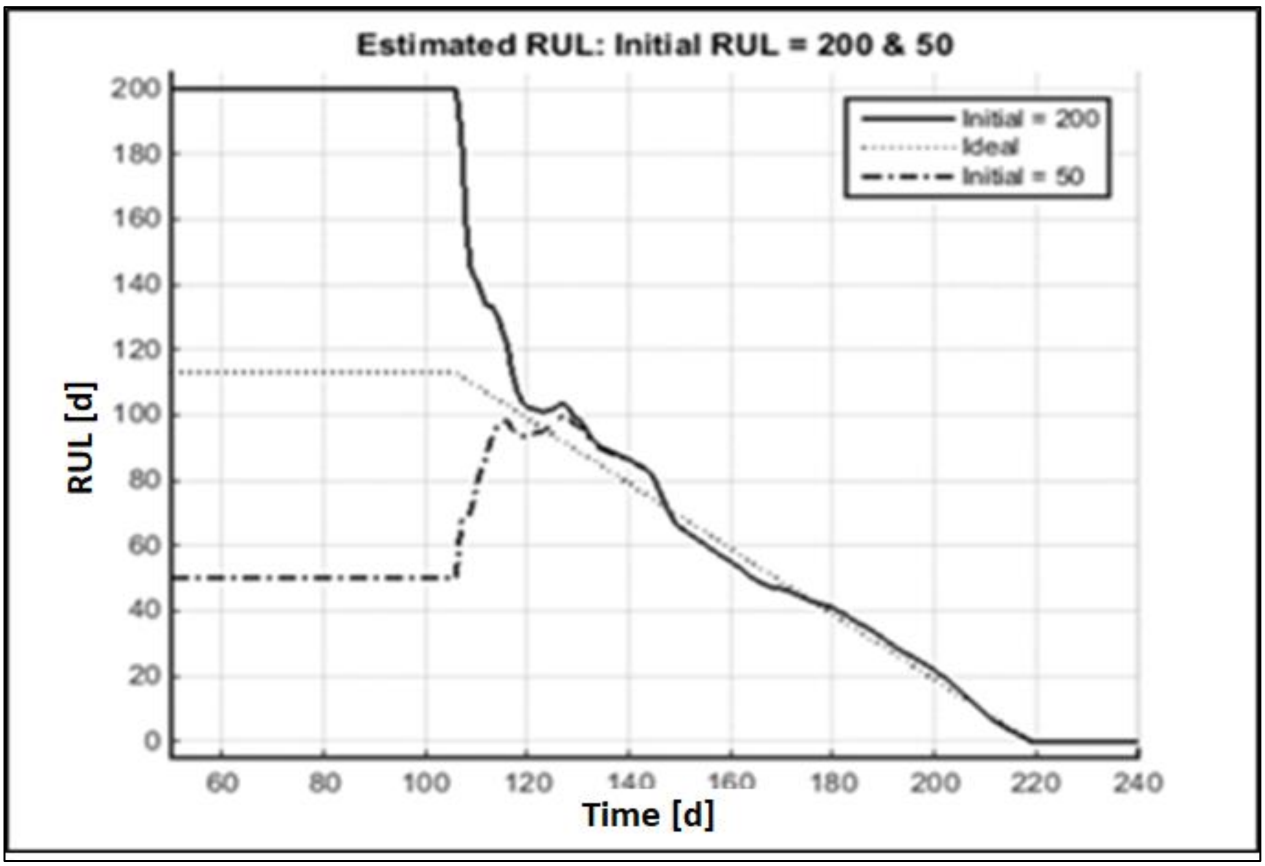


Figure 28 - Example plots of RUL given three values of initial estimates: 200, 115; and 50 days

3.3 Accuracy

The plots in Figure 27 are annotated with the phrase "Estimated PH Converged to Within 5% of Final Estimate:" the '5%' is a value called alpha (α) that is used as a margin of accuracy and, in practice, is not reasonable: 25% and 10% are realistic values for noisy and distorted data. Ideally, accuracy would be relative to true values such as PH_0 and Prognostic Distance (PD_0). For example,

$$\boldsymbol{PD_0 = EOL_0 - BD_0} \quad \textbf{(15)}$$

Except for well-controlled experiments where faults/degradation are injected at specific, known times, true values cannot be exactly known or exactly detected – there will always be a finite, time-to-process-data delay: therefore, ARULE relates accuracy to PD_{MAX}, which can be exactly calculated as follows:

$$\boldsymbol{PD_{MAX} = EOL - BD} \quad \textbf{(16)}$$

But EOL, the time when functional failure occurs, becomes known only after functional failure occurs, a data sample is taken, and a failure condition is detected.

3.3.1 Near-real Time Estimates of Accuracy

For usability, ARULE provides near-real time estimates of the accuracy of prognostic information as each data point is sampled and processed. For each sample at time TS_i:

$$\boldsymbol{EOL_{EST(i)} = EOL_i \pm \Delta t} \quad \textbf{(17)}$$

Where Δt is algorithmically determined by using trending methods and dynamically adapting a signature model to input data. Then,

$$\boldsymbol{EOL_{EST(i)} \neq EOL_i} \quad \textbf{(18)}$$

$$\boldsymbol{PD_{MAX(i)} = EOL_{EST(i)} - BD} \quad \textbf{(19)}$$

$$\boldsymbol{RUL_i = EOL_i - TS_i} \quad \textbf{(20)}$$

$$\boldsymbol{PH_i = TS_i - BD + RUL_i = TS_i - BD + EOL_i - TS_i} \quad \textbf{(21)}$$

$$\boldsymbol{PH_i = EOL_i - BD} \quad \textbf{(22)}$$

In general, $\mathrm{PH_i} \neq \mathrm{PD_{MAX}}$. Initial calculations of PH_i will be higher than or lower than $\mathrm{PD_{MAX}}$ and accuracy can be calculated as:

$$\boldsymbol{accuracy = 100(PH_i/PD_{MAX})} \quad \textbf{(23)}$$

For example, suppose $\mathrm{PH_i} = 0.95PD_{\mathrm{MAX}}$ (95% of maximum PD), then

$$accuracy = 100\,[(0.95)/1] = 95\ \%$$

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

3.3.2 Alpha (α), Margin of Accuracy

Alpha (α), a margin of accuracy, is defined as the difference (in percent) between 100% and the calculated accuracy:

$$\alpha = 100 - accuracy = 100(1 - PH_i/PD_{MAX}) \tag{24}$$

3.3.3 Convergence of Accuracy

Experimentation reveals that convergence of initial estimate errors to actual time of failure is faster when the initial estimate is higher than the actual time of failure. The recommendation is to specify a value for Prognostic Information Time-to-Functional-Failure (PITTFF) in ARULE that is 50% to 100% higher than expected as shown in Figure 29 where ideal input data and PITTFF is 50% higher than the actual PH.

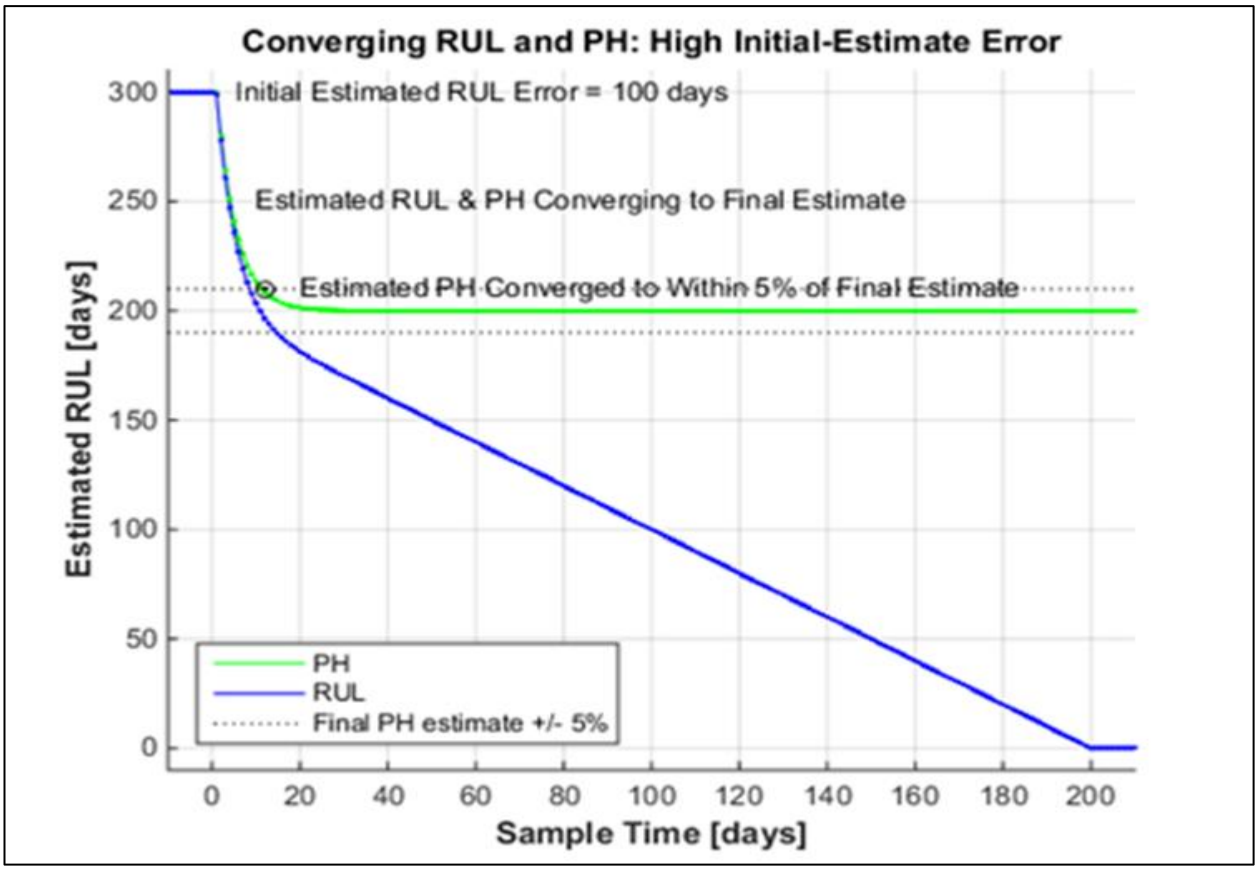


Figure 29 – Example of pseudo–ideal prognostic information. Ideal input FD signature data, non–ideal (300 days) estimate of time–to–failure (200 days) – an initial estimate error of 50 percent

In Ridgetop's recently published IEEE paper, "Reducing Signature Models for Extended Kalman Filtering for Adaptive Prognostic Estimation", Ridgetop engineers demonstrated how accuracy (a) could be evaluated according to Prognostic Horizon (PH) rather than to remaining lifetime (l). By evaluating accuracy with this method, Ridgetop is able to overcome the issue where remaining lifetime is a quantized value that approaches zero, but can never actually reach zero because of sampling intervals, digitization, aperture jitter, measurement uncertainty, and so on. Figure 30 show how accuracy can be evaluated using the RUL and PH Plots, and Table 2 shows the accuracy results for all 20 data sets that utilized in the different real-world examples in Section 6.

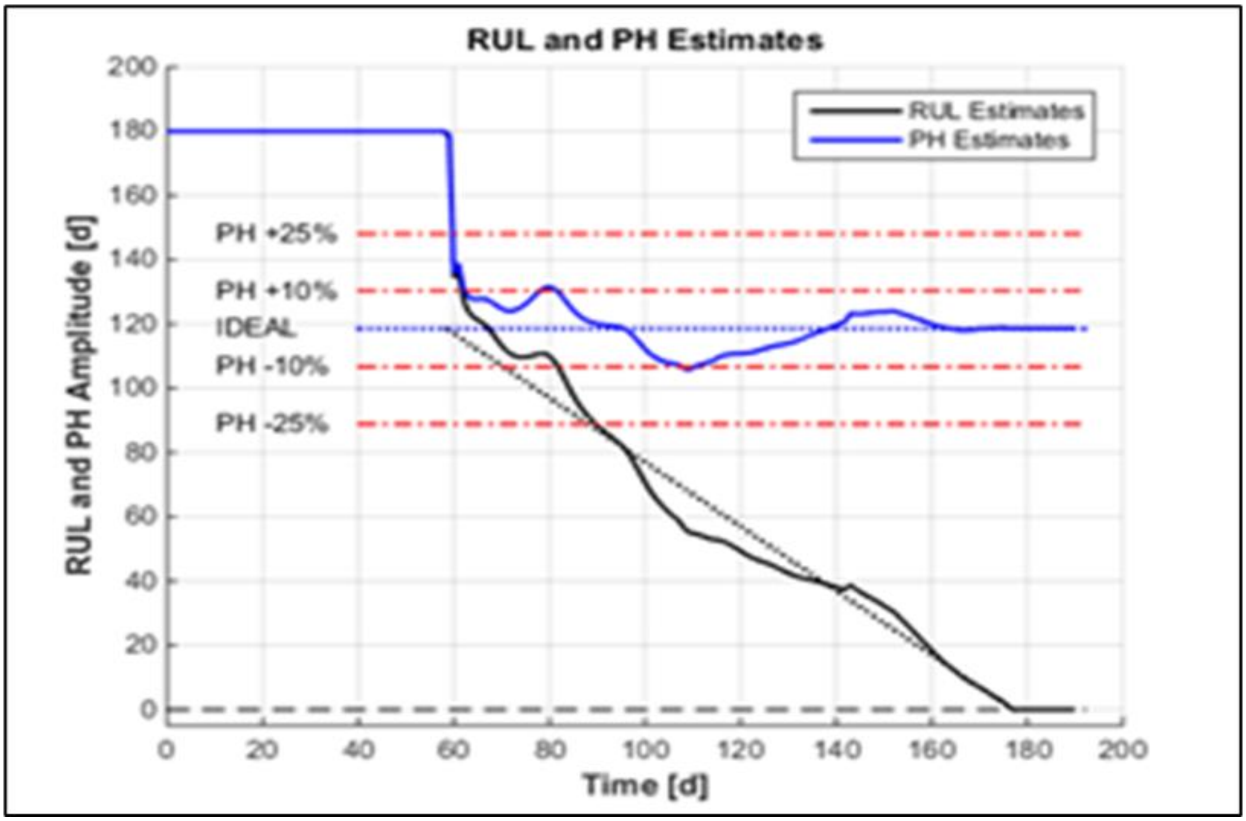


Figure 30 - Plot of RUL and PH estimates where the horizontal dashed lines represent 25% and 10% margins of accuracy

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

Table 2. Accuracy Results: 25% and 10% Margins of PH Accuracy Relative to SoH

Input File * variation	PH (maximum)	SoH when PH estimates within accuracy percent	
		25%	10%
SP-4000-1	114.4	96.2	62.2
SP-4000-1*	109.8	94.2	62.6
SP-4000-2*	117.6	96.2	60.5
BATTIR	245.9	97.2	93.7
BHCHG	41.9	95.2	80.3
CS-CHGWC	32.9	93.1	93.1
CS-CHGBC	47.9	95.5	93.9
VR2400	127.2	88.2	59.1
VR2400*	123.0	88.0	65.2
EMARSPHT	14.4	96.7	96.7
EMASTPHA	16.0	96.4	69.0
Failures	16.6	96.2	75.1
ROTOGB2	53.4	75.0	58.3
VRPCAP	33,324.0	83.7	68.4
BPS03	15.5	97.8	87.6
CLEAK	22,124.0	93.5	78.3
CLEAK*	20,727.0	96.0	93.0
SUBRFA	12.5	91.5	59.4
SUBRFA*	13.0	71.9	71.9
TPWR	28.3	93.8	76.2

3.4 Contributing Factors Related to Inaccuracy

3.4.1 Signature Processing

Signature processing is a source of inaccuracy. Noise margin when trying to mitigate noise:

- A value too low might result in false detection of the onset of degradation.
- A value too high might result in excessive offset in the detection of degradation.

Failure level:

- When set too low, detection of functional failure will occur much sooner than the actual time of failure.
- When set too high, detection of functional failure will occur much later than the actual time of functional failure, and might occur later than actual failure.

3.4.2 External Factors

Inaccuracies could be the result of factors outside of the ARULE Adaptive Prediction Kernel (APK).

- Actual Sampling period (in terms of hours and days rather than microseconds)
- Burst mode sampling at a much higher frequency than the sampling frequency
- Digitization errors due to (analog-to-digital - ADC) and (digital-to-analog - DAC) data conversions
- Rounding errors related to averaging, internal modeling, aperture jitter, and so on
- Time differences between the actual sampling and the system-recorded sampling moments
- Fluctuations in data values due to variations in, for example, the operating environment

3.5 Prognostic Information: Review of RUL, PH, and SOH

A first step in an evaluation process for a prognosing stage is understanding the types of and causes of inaccuracies in prognostic information. The plots in Figure 31 are used to illustrate, for review purposes, selected prognostic terminology: Table 3 on page 39 is a list of the names of prognostic terms plus a summarized description.

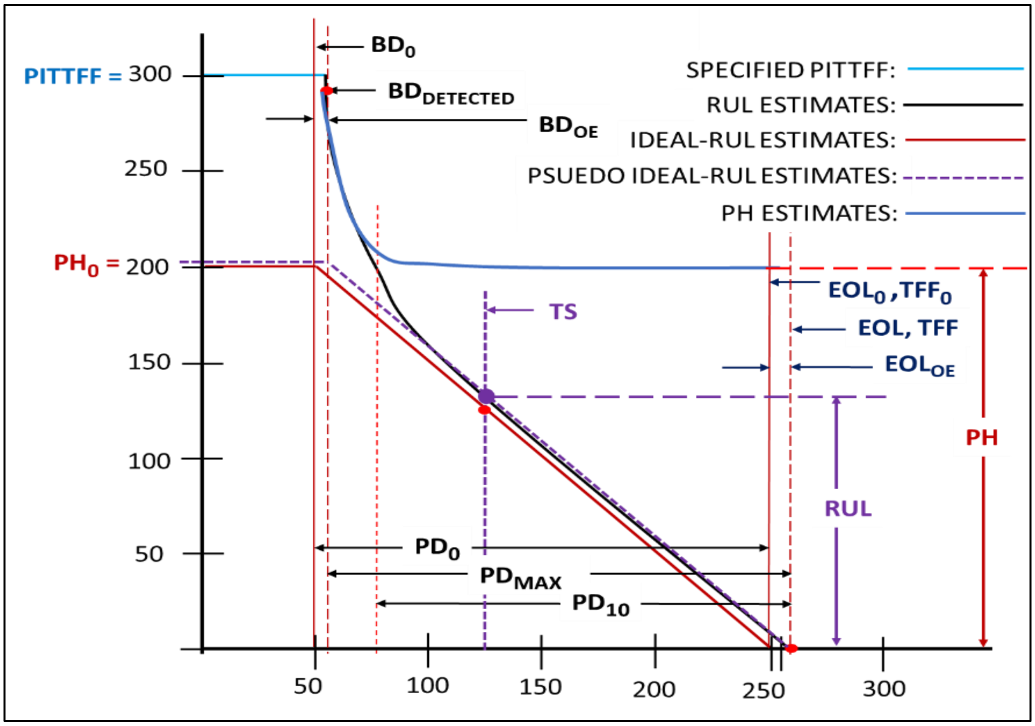


Figure 31 - Ideal and non-ideal plots for terminology and metrics explanation.

3.5.1 RUL

RUL is a distance in time between an estimated future time when functional failure occurs (EOL) and the current time (TS) when the data was sampled.

$$\boldsymbol{RUL_i = EOL_i - TS_i} \quad \textbf{(25)}$$

RUL is very important and useful in prognostics. It is used, for example, to schedule maintenance and repair to avoid an unplanned outage of a system. Another important use is to evaluate the likelihood that a degraded piece of equipment will last long enough to complete a mission within a specified period of time.

3.5.2 PH

One classical definition of PH (prognostic horizon) is the following: PH is the advance distance in time, to a specified level of accuracy, a failure can be predicted.

$$\boldsymbol{PH = time(1 \pm da)} \quad \textbf{(26)}$$

where da is a value, such as 0.05, which defines the required accuracy of the value of PH. The primary issue with that definition is it is relative to a decreasing value. ARULE uses the following:

$$\boldsymbol{PD_{MAX} = EOL_{EST(i)} - BD} \quad \textbf{(27)}$$

$$\boldsymbol{RUL_\alpha = EOL_\alpha - TS_\alpha} \quad \textbf{(28)}$$

$$\boldsymbol{PH_\alpha = TS_\alpha - BD + RUL_\alpha} \quad \textbf{(29)}$$

3.5.3 SOH

SoH information are values ranging from 100 % (full health) to 0 % (no health) and is calculated for every data sample,

$$\boldsymbol{SoH_i = 100(\,RUL_i/(EOL_i - BD)} \tag{30}$$

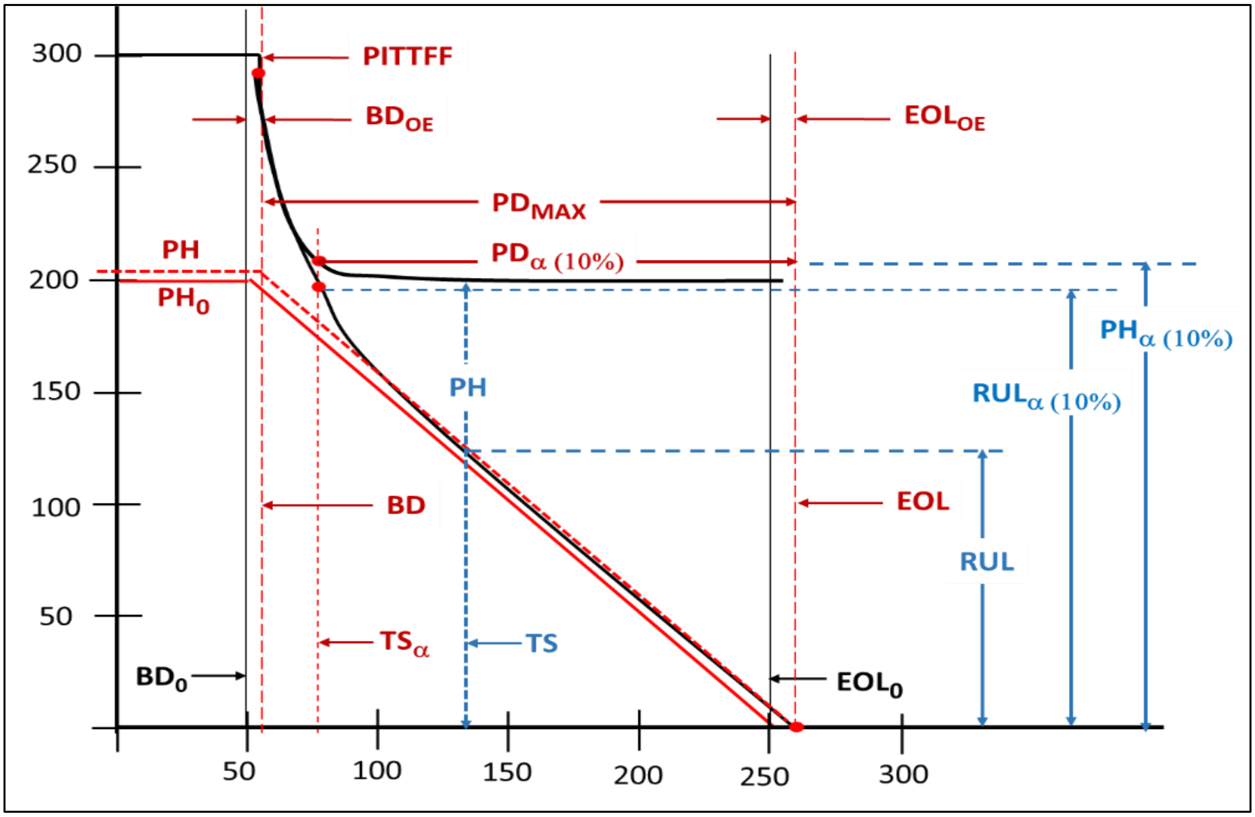


Figure 32 - Example plots to illustrate prediction information (except for SOH)

3.6 Prognostic Events

In prognostics, there are two major events of particular interest: a beginning event and an ending event – the beginning (onset) of degradation and the end of useful life when functional failure occurs.

3.6.1 Beginning of Degradation

- **True Beginning of Degradation** - Is a point in time, BD_0, when degradation of a prognostic target begins. In practice, except by lucky chance, it is not possible to truly detect the exact point in time because of, for example, noise, discrete sampling, and/or digitization of data. Sampling points are much more likely to happen before or after rather than exactly when degradation begins.

- **Detected Beginning of Degradation** - In practice, degradation is detected by processing at a point in time, $BD_{DETECTED}$ (also BD), a sampled data point (top-most red point in Figure 32) that is situated after degradation has begun.

3.6.2 End of Life – Functional Failure

- True End-of-Life (also TTF_0, Time of Functional Failure) - The point in time, EOL_0 or TTF_0, when End of Life (Functional Failure) occurs. In practice, except by lucky chance, it is not possible to detect this point in time because of, for example, noise, discrete sampling, digitization of data, and signature smoothing (lowers signal level).
- **End-of-Life** - In practice, End-of-Life is detected by processing at a point in time, EOL (also, $EOL_{DETECTED}$ or TFF), a sampled data point after the time corresponding to the failure level, as illustrated by the bottom-most, red-colored data point Figure 32. In one important respect, this is 'undesirable' because failure is reported after, instead of before, the event. One way to 'mitigate' that is to specify a lower failure level.

3.7 Prognostic Distance (PD)

Classically, PD is the time needed to make a failure prediction and take action, such as repair or shutdown, which can be restated as follows:

$$PD = minimum\ required\ RUL$$

That classical definition is qualitative and furthermore, it includes an indefinite, non-quantitative event (take action) that is not related to the accuracy of a prognosing stage and its prediction algorithms. A quantifiable definition of PD is needed.

3.7.1 True Prognostic Distance

True PD is the distance in time between the true onset of degradation and the true time of functional failure.

$$\boldsymbol{PD_0 = EOL_0 - BD_0} \quad \textbf{(28)}$$

In practice, PD_0 can only be measurably determined by using a well-controlled experiment in which the onset of degradation is forced at a known time and functional failure occurs as the result of an injected fault at a known time. Determining PD values, such PD_0, in a controlled, well-design experiment is useful for evaluation purposes of (1) the effects of sensor sampling rates, noise filtering, windowing and feature extraction, and digitization; (2) the effects of data conditioning to process input CBD and produce output FD signature data, with or without additional noise filtration and/or mitigation, to a prognosing stage; and (3) the efficacy and accuracy of a prognosing stage, such as the ARULE APK, to perform specified signature transforms and noise mitigation to provide prognostic information.

3.7.2 Maximum Prognostic Distance

The maximum PD is the distance in time between detection of degradation and the detection of functional failure. The value of PD_{MAX}, when compared to the value of PD_0, can be used as a primary performance metric of the sensing stage. Optimally, the difference between the two values should be no larger than one or two sampling periods and, at the same time, be within the desired or specified margin of accuracy.

Unacceptable differences in value are not due the deficiencies in the design and operation of ARULE: instead, corrective actions need to be applied to the sensing and data processing stages: (1) sensing stage - primarily noise filtering, sampling methods, and sampling rates; (2) data processing – primarily data conditioning and extraction, domain transforms, fusion of data with data and data with models, and input specifications to a prognosing stage.

3.7.3 Beginning of Degradation, Offset

Another inaccuracy is caused by an offset between the true time of the onset of degradation and the time when degradation is detected: $\mathrm{BD_{OE}}$.

The difference in time between BD_0 and $BD_{DETECTED}$ is an offset. The magnitude is dependent on the following: the magnitude of unfiltered noise in the data – the higher the magnitude, the higher the value of Noise Margin (NM) required to mitigate noise and the larger offset; the time period of periodic sampling – higher sampling rates have smaller time periods, which lowers the offset; and measurement uncertainty, including inaccuracies due to quantization and rounding during data conversions.

3.7.4 End of Life, Offset

Another inaccuracy is caused by an offset between the true time of functional failure and detection of functional failure: $\mathrm{EOL_{OE}}$. The difference in time between $\mathrm{EOL_0}$ and $BD_{DETECTED}$ is an offset. The magnitude of the offset is dependent on many factors, primarily the magnitude of unfiltered noise in the data – the higher the magnitude, the higher the value of NM required to mitigate noise and larger the offset.

3.7.5 Methods to Mitigate Offset Error

The following methods may be used to mitigate offset errors: (1) use a sensor and/or extract feature method(s) that filters more noise; (2) reduce the value of noise margin; (3) increase the sampling rate; (4) use data converters that have a higher effective number of bits (ENOB); (5) use measurement devices having a higher resolution; and (6) use adaptive application-specific calibration.

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

3.8 Summary of Prognostic Terminology and Performance Metrics

Table 3 is a summary list of prognostic terms, their names, and a brief description or explanation. More detailed information was previously presented in this chapter.

Table 3. List of Prognostic Terms, Names, and Description

Term	Name – Example Figure	Description/Explanation
$\boldsymbol{\alpha}$	Alpha (accuracy)	• Used with PD, PH, RUL, and SoH. • When estimates are within α of actual
$\mathbf{BD_0}$	True BD	• Actual time when degradation begins
$\mathbf{BD}$ $\mathbf{BD_{DETECTED}}$	Estimated beginning of degradation	• Estimated because of measurement uncertainty, noise margin, and sampling
$\mathbf{BD_{OE}}$	BD offset $BD_{OE} = BD_{DETECTED} - BD_0$	• The inaccuracy (time) between actual and estimated beginning of degradation
Convergence	Convergence of estimates to within a specified margin (actual +/- margin)	• Transition from highly inaccurate values of RUL to within a specified accuracy
$\mathbf{EOL_0}$ $\mathbf{TFF_0}$	True EOL Time of Functional Failure	• Actual time when end of life (functional failure) occurs
EOL $\mathbf{EOL_{DETECTED}}$ **TFF**	Estimated time of End of Life Detected time of End of Life Estimated Time of Functional Failure)	• Estimated because of measurement uncertainty, noise margin, sampling, level shifting due to conditioning/transforms
$\mathbf{EOL}_{\boldsymbol{MAX(i)}}$	Estimated EOL at time after the time of a sample (TS_i)	• Used to calculate an estimated value for PD_{MAX} to produce estimates of accuracy
$\mathbf{EOL_{OE}}$	EOL offset $EOL_{OE} = EOL_{DETECTED} - EOL_0$	• Time between actual end of life and time when functional failure is detected
FF	Functional Failure	State in which a prognostic target no longer operates within specifications
NM	Noise Margin	• Shift signature downward by NM value to mitigate noise and measurement uncertainty

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

Term	Name – Example Figure	Description/Explanation
PD	Estimated prognostic distance $PD = EOL - TS$	• Estimated time between functional failure and a time of a sample
$\mathbf{PD_0}$	True Maximum Prognostic Distance $PD_0 = EOL_0 - BD_0$	• Time between the actual beginning of degradation and the time when functional failure occurs
$\mathbf{PD_{MAX}}$	Estimated Max. Prognostic Distance $PD_{MAX} = EOL_{MAX} - BD$	• Used in lieu of $\mathbf{PD_0}$ because BD and EOL are estimated
PH	Prognostic horizon, For relative time (BD = 0) $PH_i = (TS_i - \mathrm{BD}) + RUL_i$	• Estimated time after BD when functional failure occurs
$\mathbf{PH_\alpha}$	Subsequent PH estimates within $\alpha \leq$ $[abs\,(PD_{MAX} - PH)/PD_{MAX}]\,100$	• When the percent difference between $\mathbf{PD_{MAX}}$ and $\mathbf{PH}$ is within α percent
PITTFF	Prognostic Initial Time-to-Functional-Failure used as an initial RUL estimate absent detection of degradation	• A specified value to be used for estimated RUL in the absence of detected degradation
RUL	Remaining useful life $RUL = EOL - TS$	• The distance between EOL (estimated) and the sample time
$\mathbf{RUL_\alpha}$	RUL for PH_α	• The value of RUL when $\mathbf{PH_\alpha}$ is achieved
SoH	State of Health)	• Estimated health (in percent) of a prognostic target
$\mathbf{SoH_\alpha}$	SoH value for $\mathbf{PH_\alpha}$	• SoH value corresponding to $\mathbf{PH_\alpha}$
TB	Burst-mode period	• Time period during which data is sampled at a high-rate
TP	Sample period)	• Time between samples
TS	Sample time	• Time when data is sampled (taken): includes count, record, etc.

4. ARULE™ Installation

4.1 System Requirements

The following system requirements have been tested to ensure proper software installation and operation:

1. PC with Windows 10 Professional 64–bit operating system.
2. PC with at least 4GB of RAM installed.
3. Screen resolution of 1024x768.

Important Note: There is no specialized test equipment needed to run the software application, but the program does require some software dependencies from National Instruments. All required program dependencies are included and installed with the installation wizard. It is also recommended that the Windows PC has a minimum display resolution of 1920 x 1080 with the scale and layout settings set to 100%. The software is still operational if these specifications are not met, but the user may experience scaling issues.

4.2 Installing the Software for the First Time

Important Note: The dates and versions in these instructions and views are for illustrative purposes: they might not be the same as those for your particular install.

1. Download and extract the installation folder **ARULE_01.27.2023** from the download source as shown in Figure 33. Download source could be through Email, USB, Disk, Dropbox, etc. Also note that the installation folder may be in a .7z or .zip format and can be unzipped with the built-in Windows File Utility or the free .7z software that can be download from the following link:
   - Link for .7z File Utility: www.7-zip.org

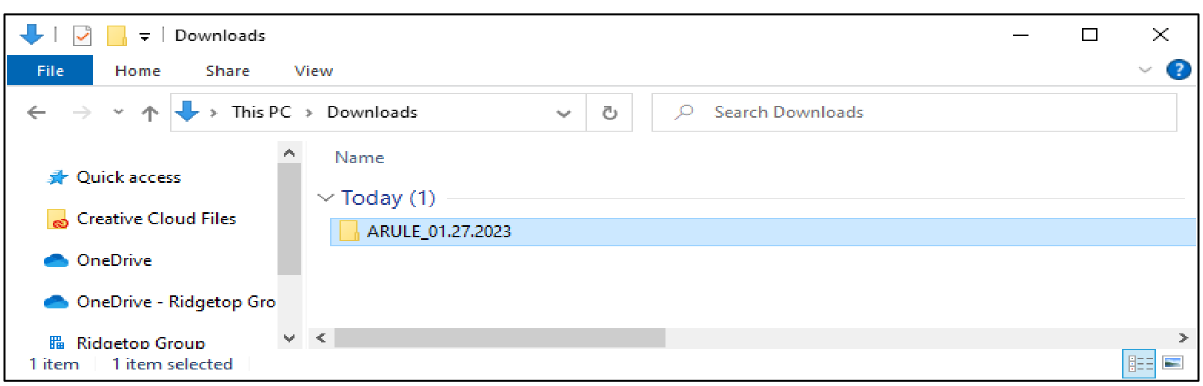


Figure 33 – ARULE™ installation folder saved in This PC > Downloads

2. Navigate to the following directory in the Windows File Explorer: **...\Downloads\ ARULE_01.27.2023.**

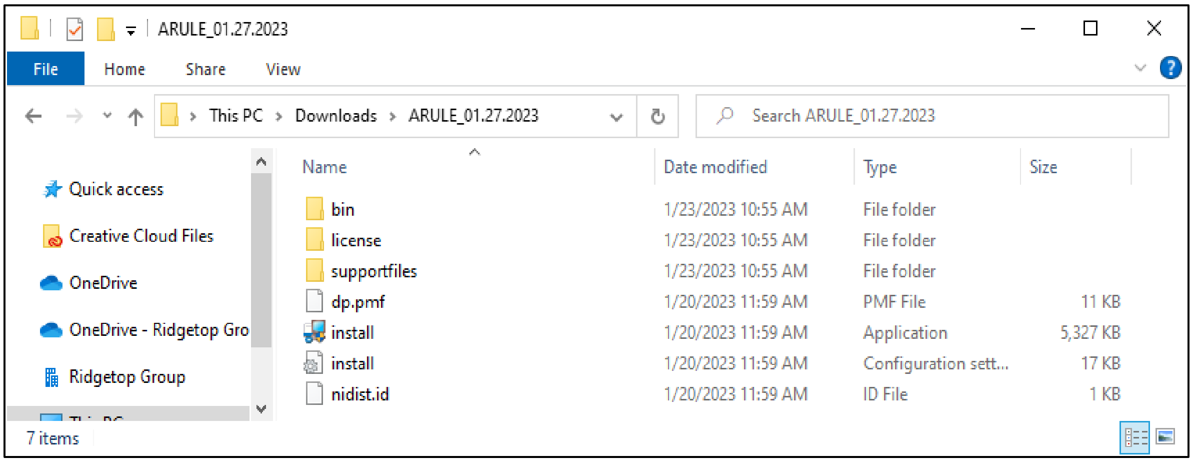


Figure 34 – Folder is located at ...\Downloads\ARULE_01.27.2023

3. Right click on the install.exe and select Run as administrator.
4. Verify that installation wizard initializes as shown in Figure 35.

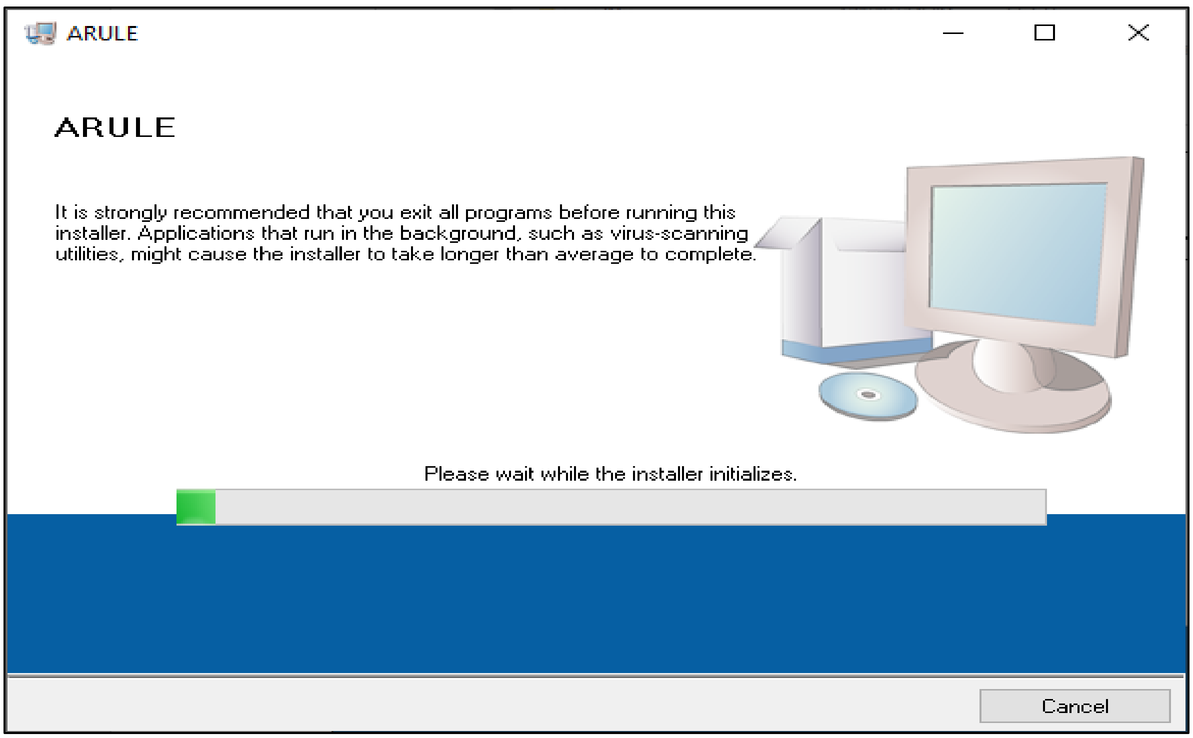


Figure 35 – ARULE™ installation wizard initializing

**3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com**

5. Follow the Setup Wizard's installation prompts as shown in Figure 36 - Figure 40:

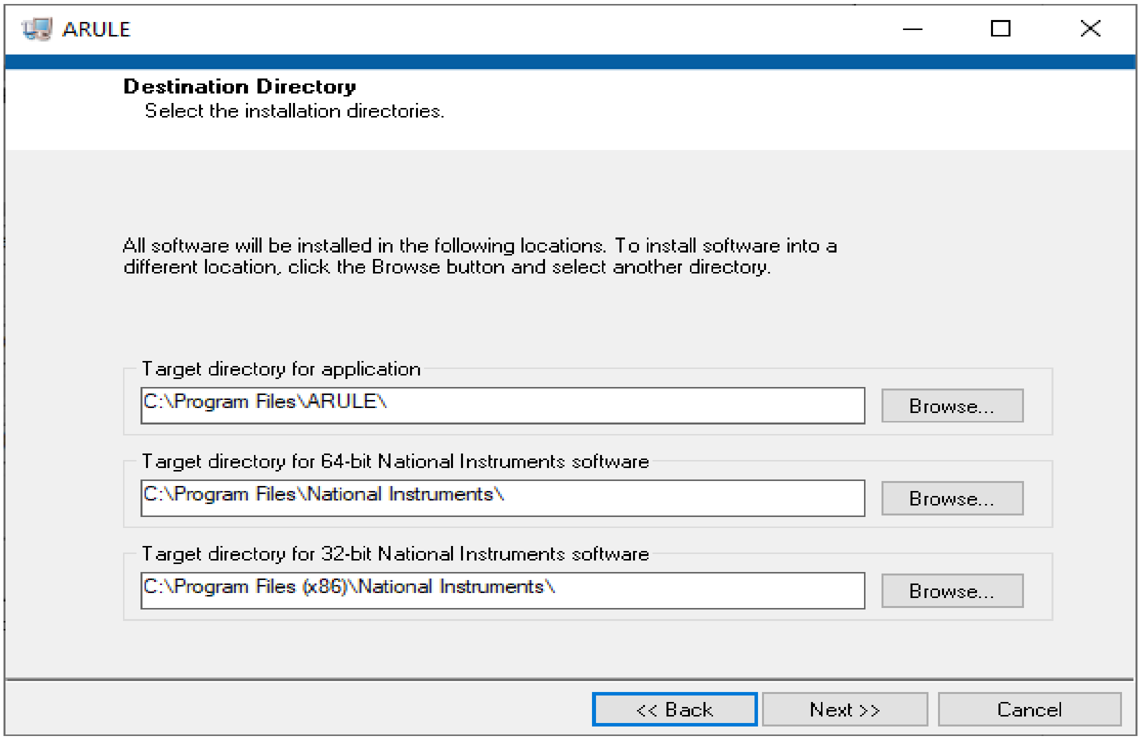


Figure 36 – View of the installation wizard for the ARULE™ installation directory. Click Next >> to continue

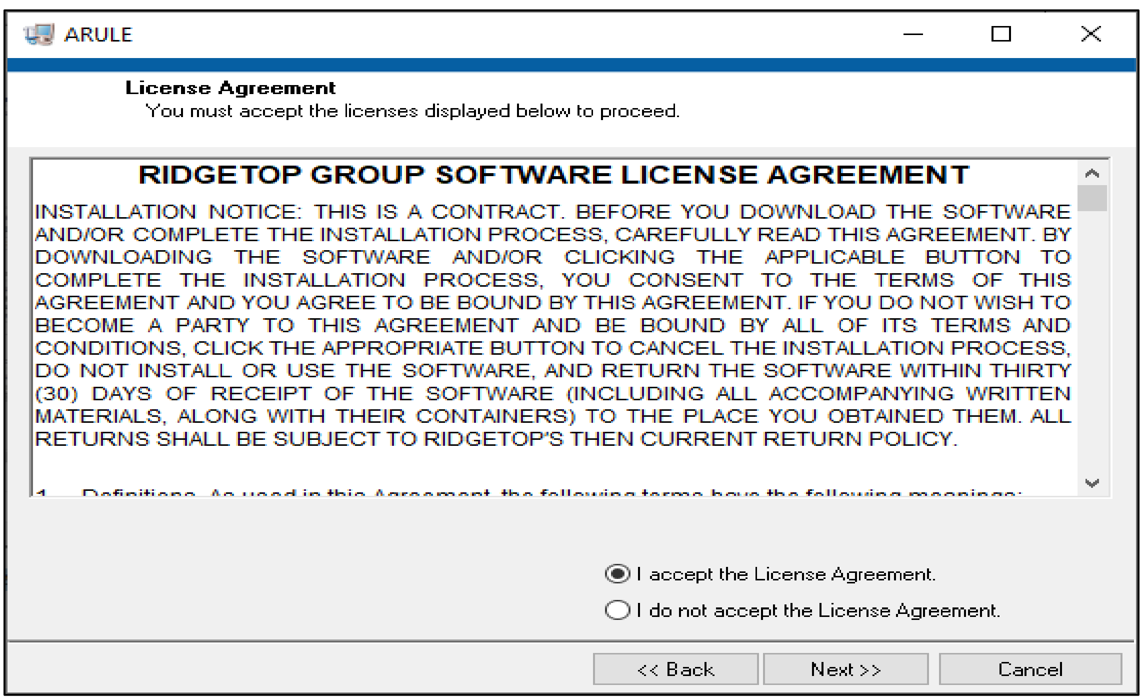


Figure 37 – View of the installation wizard that prompts the user to select the National Instruments Software License Agreement. Click Next>> to continue

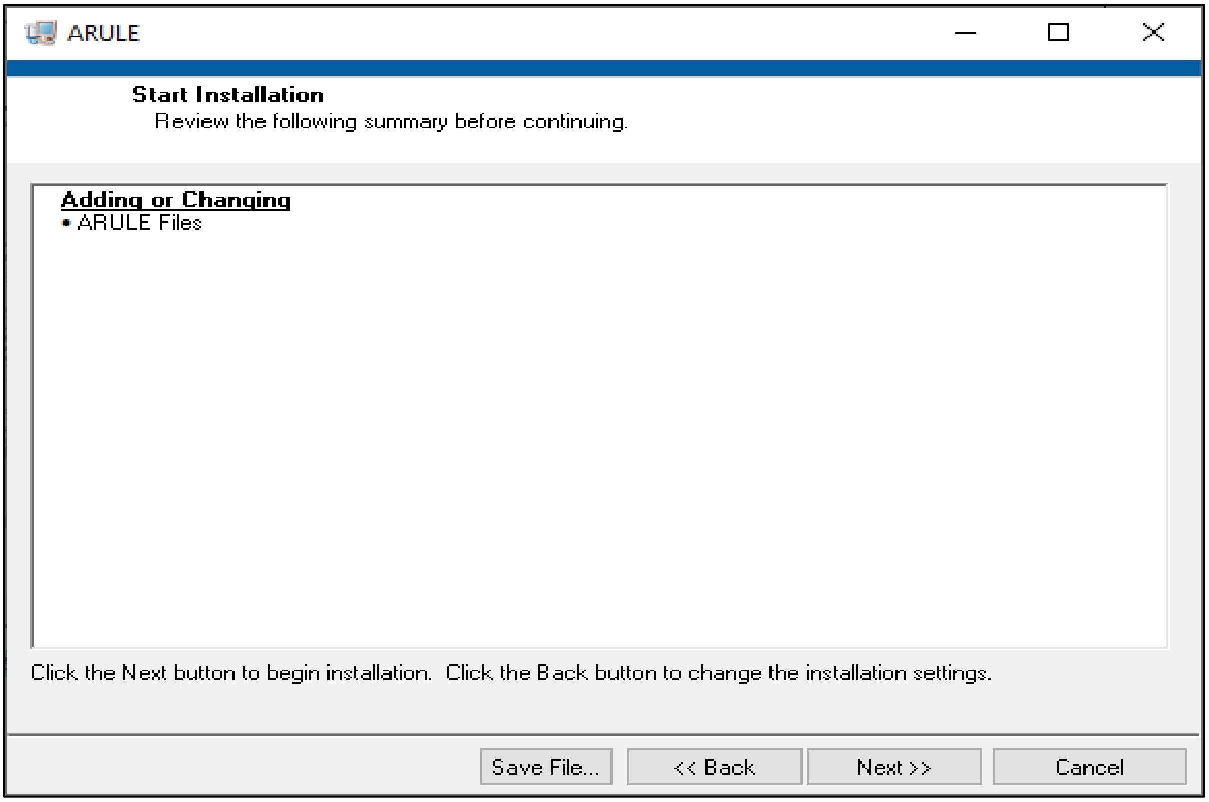


Figure 38 – View of the installation wizard to start the installation. Click Next >> to continue

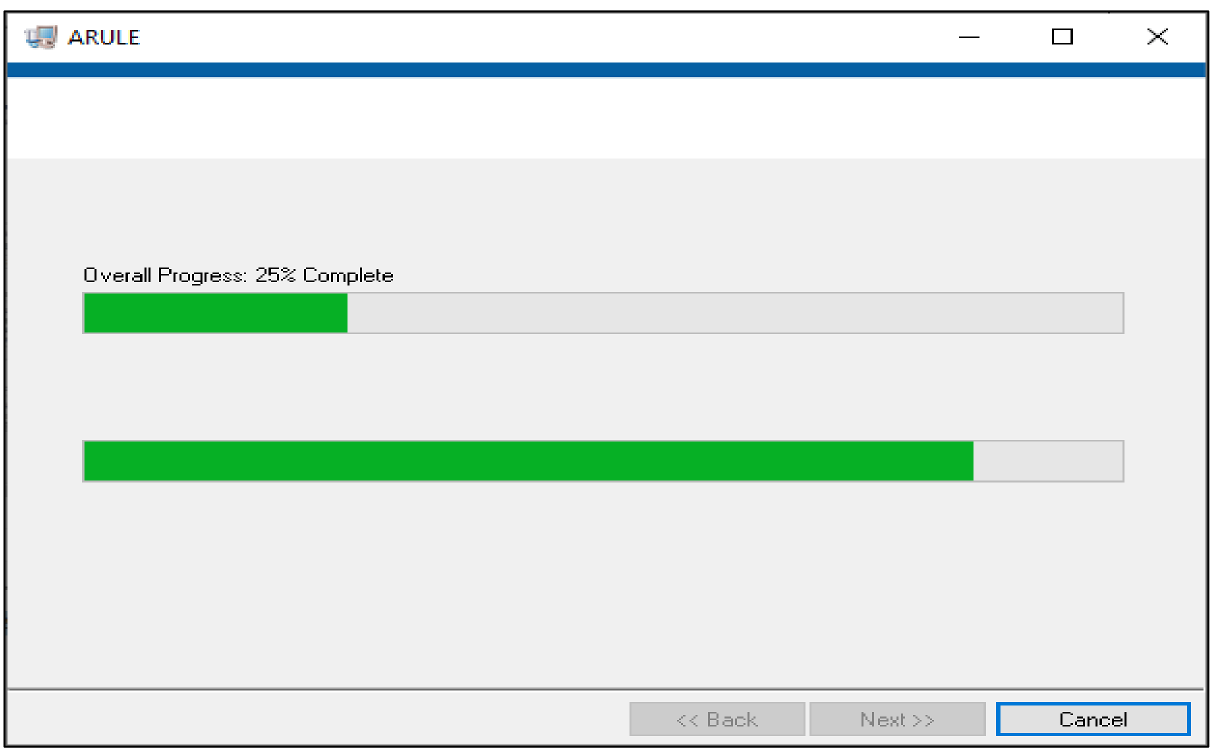


Figure 39 – View of the installation wizard installing the software

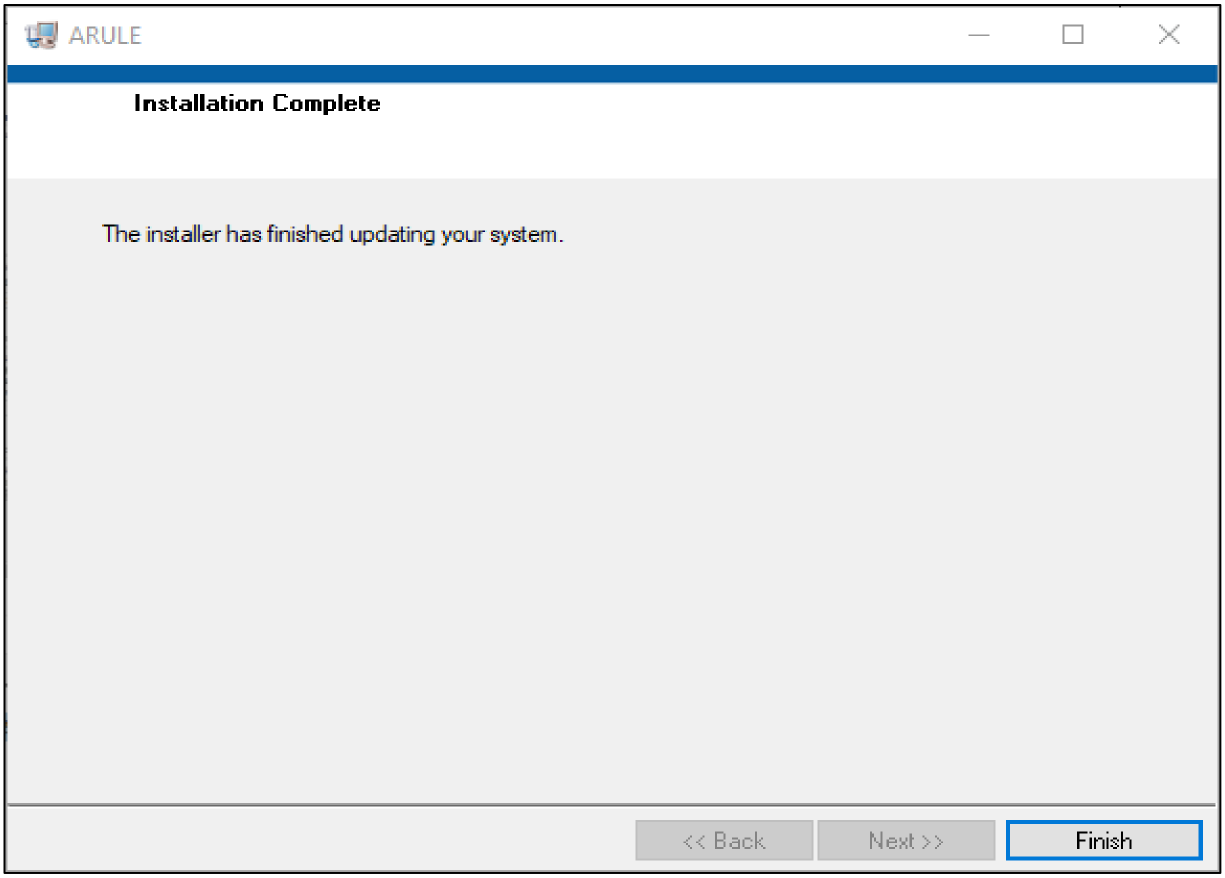


Figure 40 – View of the installation wizard when the installation has completed

6. Select **Finish** and exit the installation wizard.
7. Restart your computer to complete the installation procedure.

4.3 Activating License and Verifying Software Installation

1. Open the ARULE™ software by using the shortcut in the Windows Start Menu or by opening a File Explorer and navigating to the default installation directory. The default directory is located at the following path: **C:\Program Files\ARULE.**

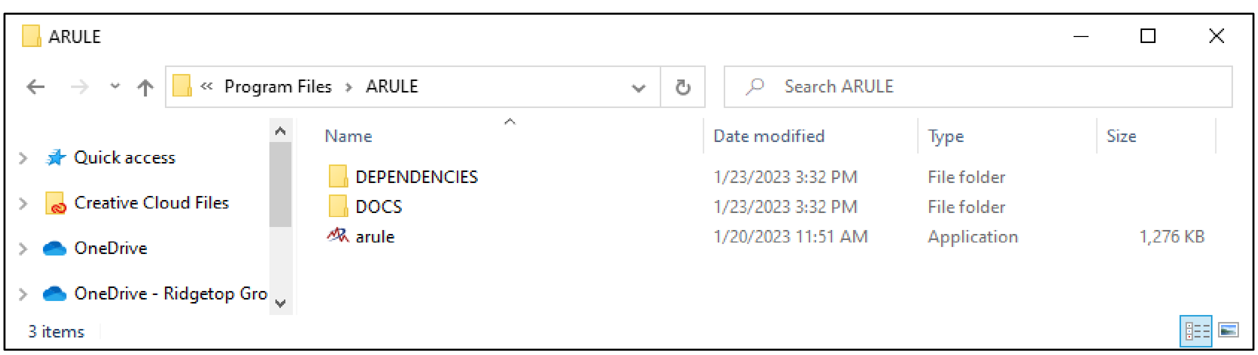


Figure 41 – Installation directory of ARULE™

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

2. Open the **arule.exe** application by either a double click or right click and select **Run as Administrator**.
3. If an active license is not found the program will prompt the user to retrieve a license from the server. To do this click the **GET NEW LICENSE** button as shown in Figure 42.

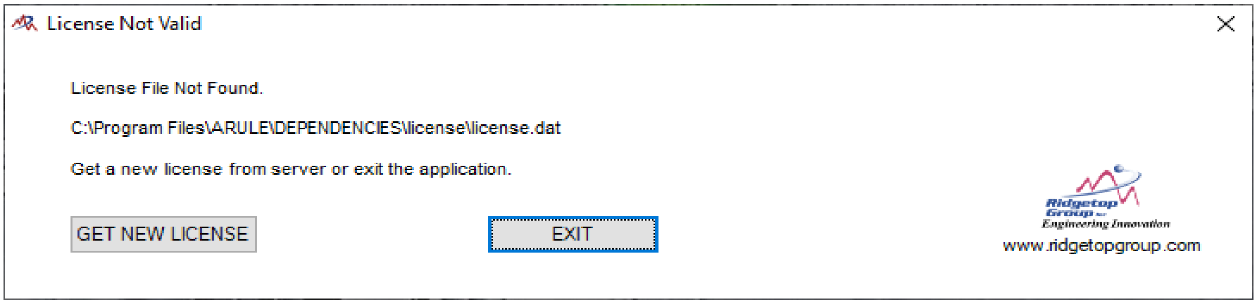


Figure 42 – License verification prompt asking user to retrieve a license from the server

4. Login to the license server with the username and password combination that was provided through email. If you do not have a username and password, please contact a Ridgetop Group representative.

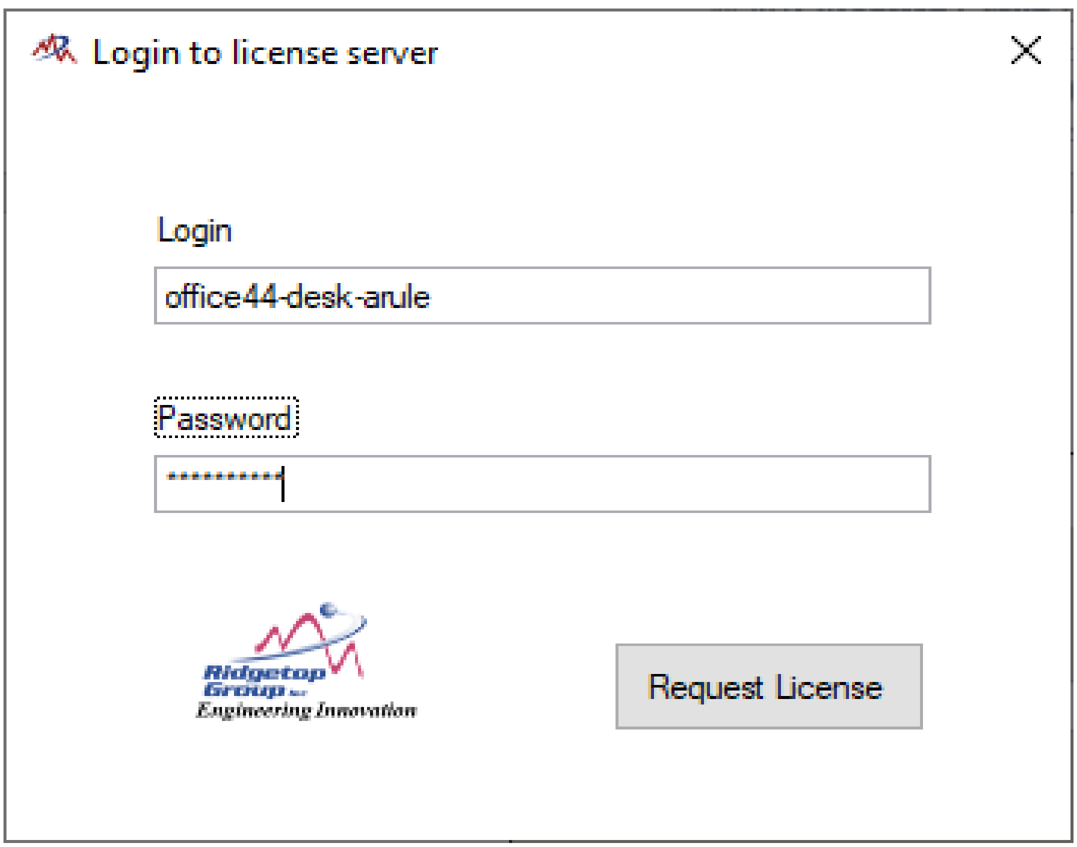


Figure 43 – Logging into the license server

5. Enter your credentials and click the **Request License** button.
6. Verify that the software license is activated as shown in Figure 44. If the login credentials were not entered correctly or if the license is expired, then the user will be prompted to get a new license.

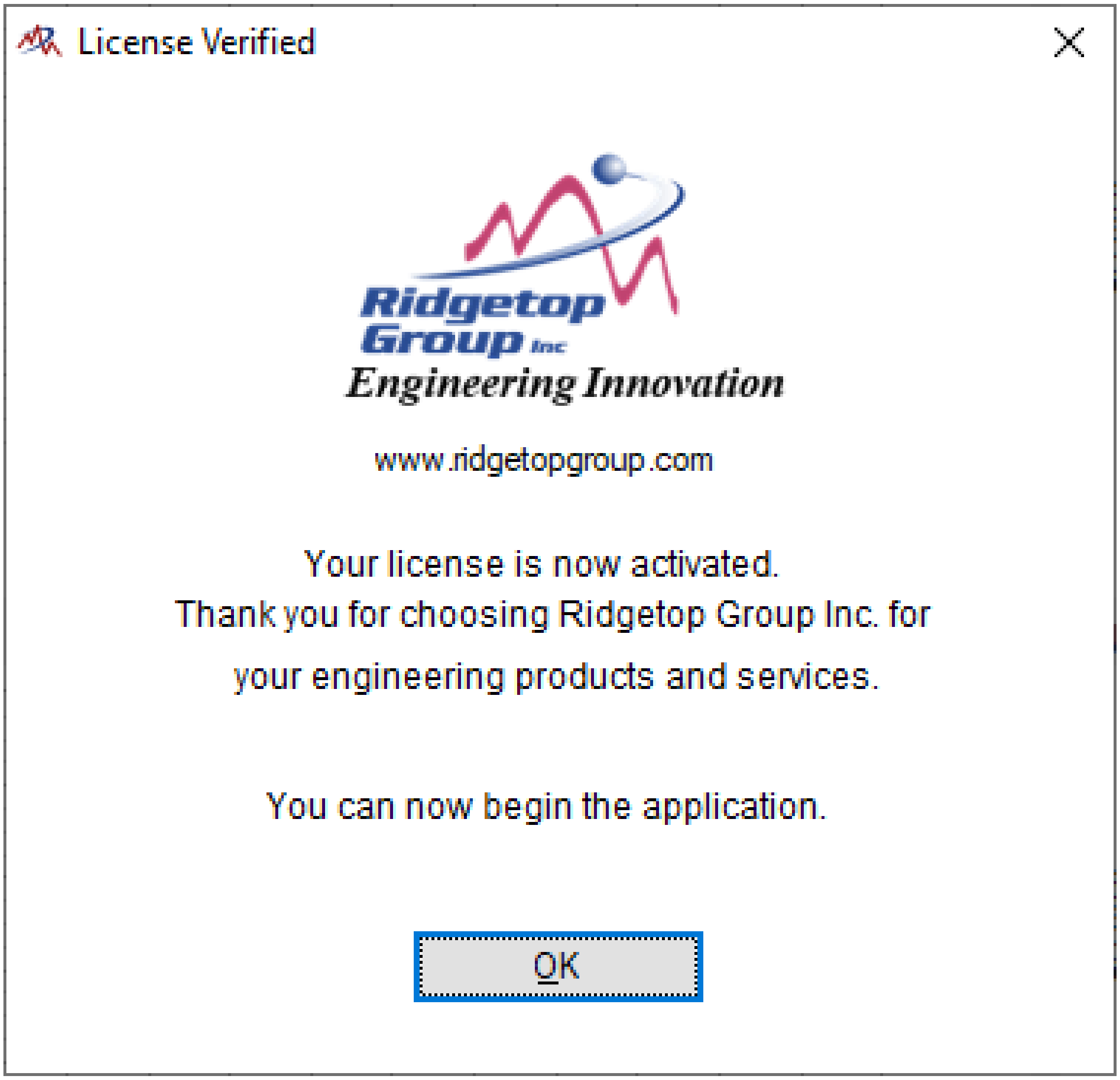


Figure 44 – Message that software license is activated and installed correctly

7. Click **OK** to continue and run ARULE™.
8. Upon first start, ARULE™ will prompt the user to select the default folder where the ARULE directory needs to be placed. Using the File Explorer prompt, select the location where you want to put this folder and click the **Select Folder** button.
   Important Note: It is recommended to place this folder in This PC > Downloads

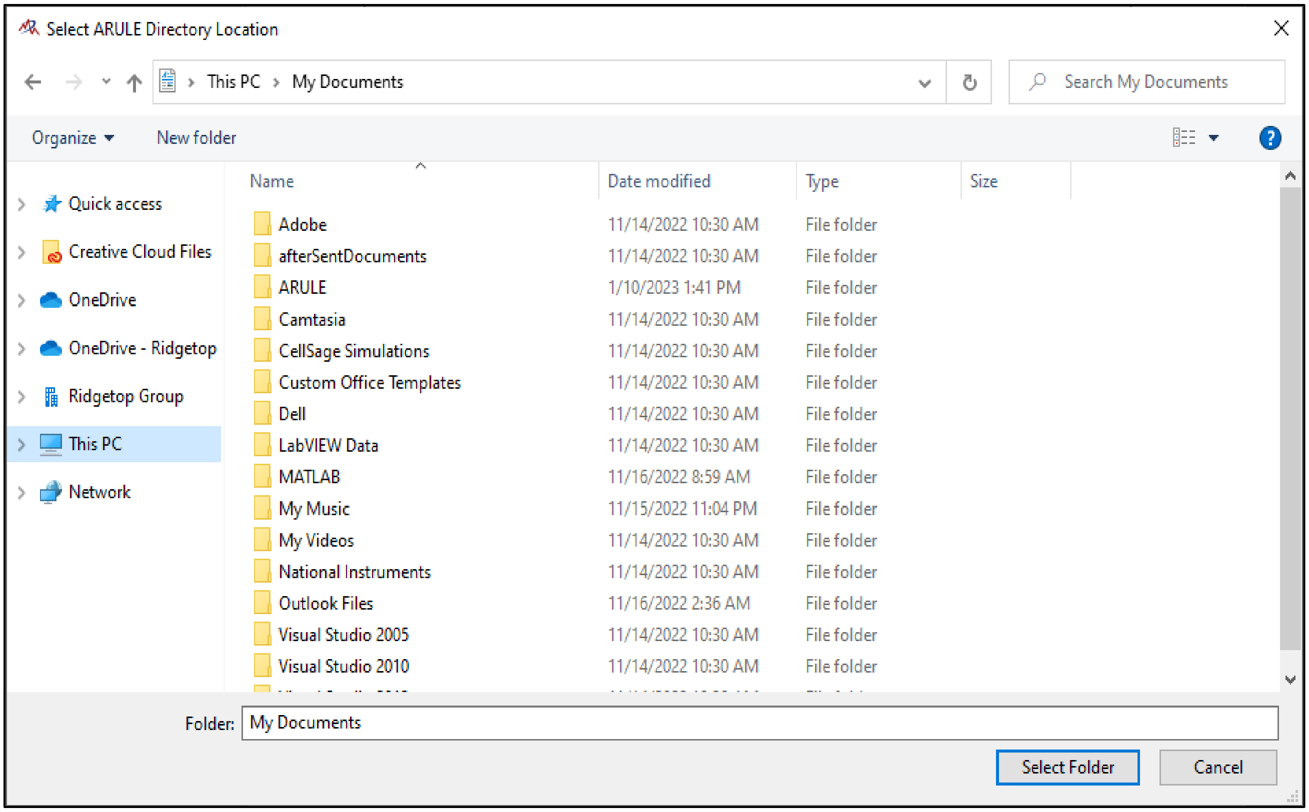


Figure 45 – Selecting the ARULE Directory location

9. After selecting the ARULE folder, ARULE™ will open in the following window.

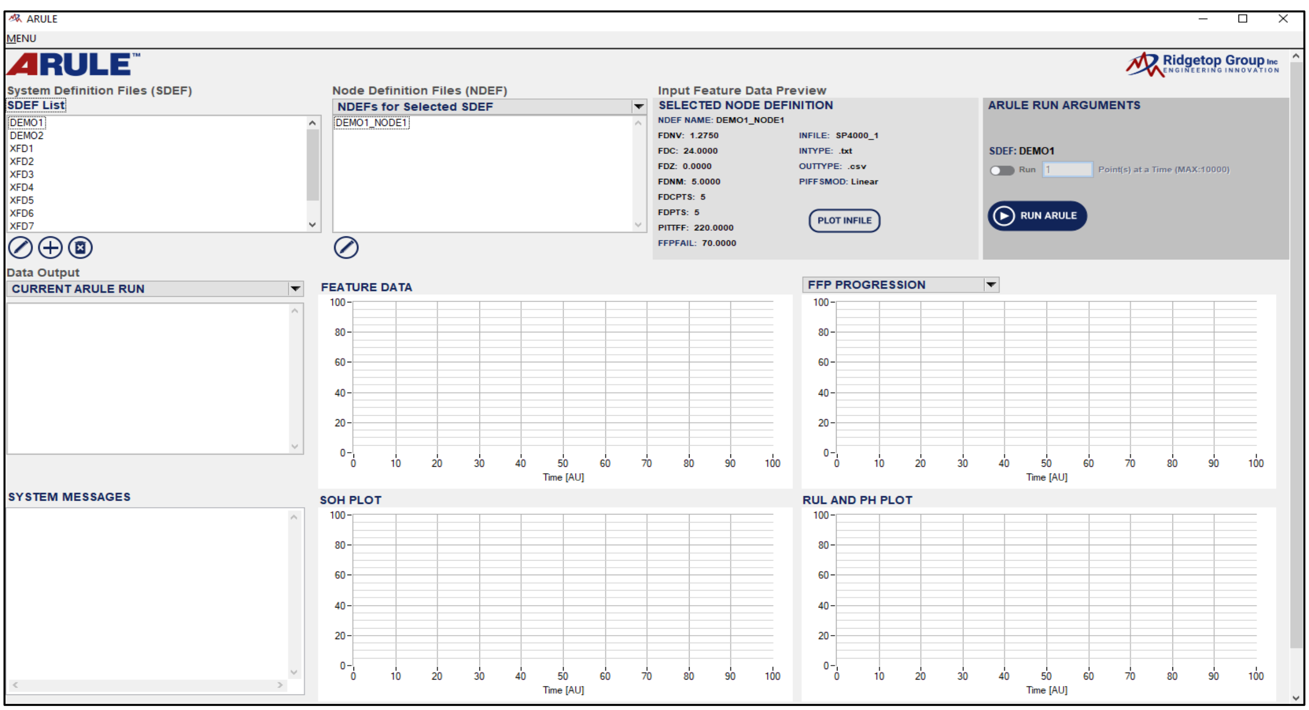


Figure 46 – View of the ARULE™ software being opened after license installation

4.4 ARULE™ Views and Display Scale and Layout

The views used by the ARULE™ GUI (Graphical User Interface) have been designed for computer displays set to a Scale and Layout of 100%: **Settings –> Systems –> Display –> Scale and Layout –> 100% (Recommended)**: see Figure 47.

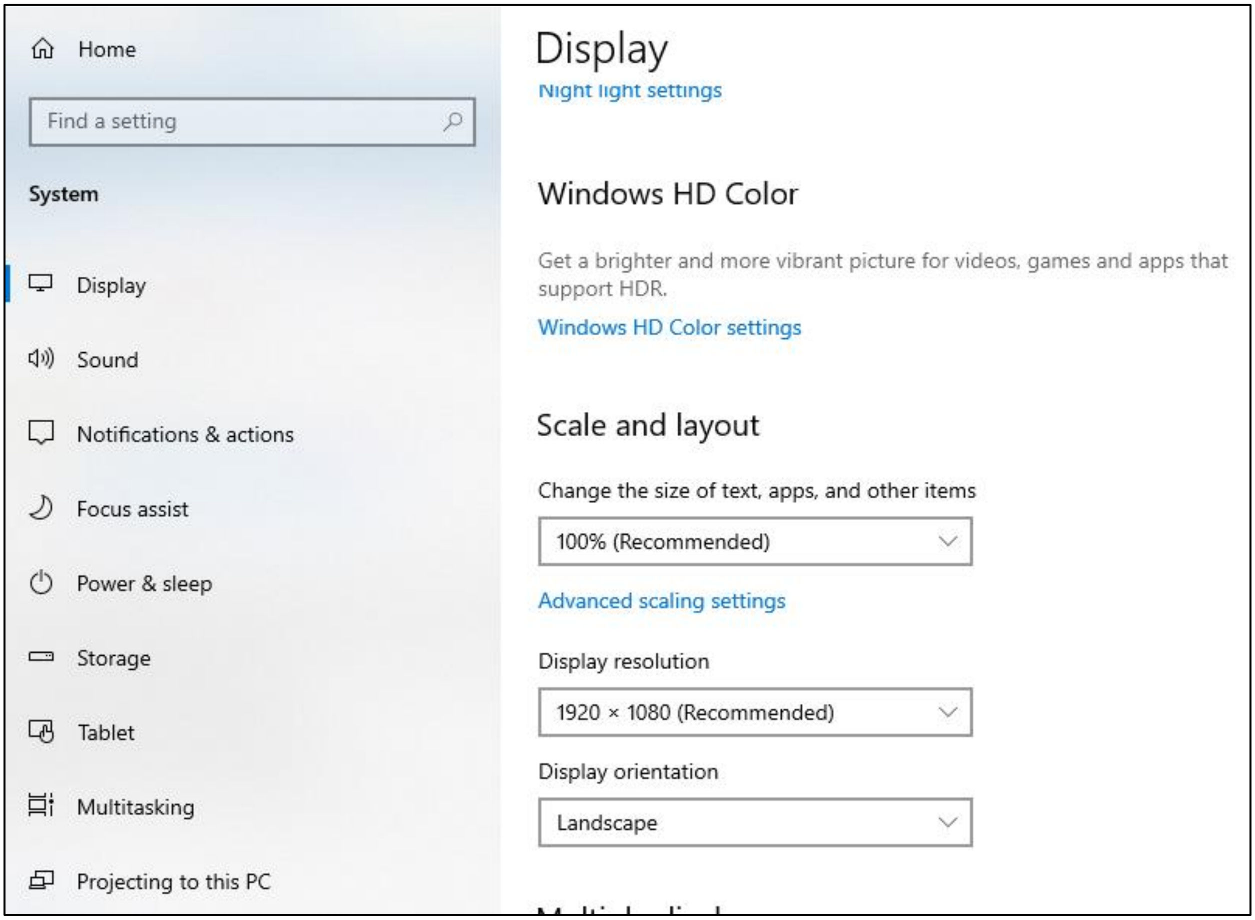


Figure 47 – Settings, System, Display, Scale and Layout: recommended values

5. ARULE™ User Interface

5.1 ARULE™ GUI Overview

The ARULE™ GUI comprises of the System Definition Input and Data Output, collectively in a single window. The programming logic follows a sequential flow of operations, where there are six main sections in the ARULE™ GUI. A visual for the single window setup is shown in Figure 48.

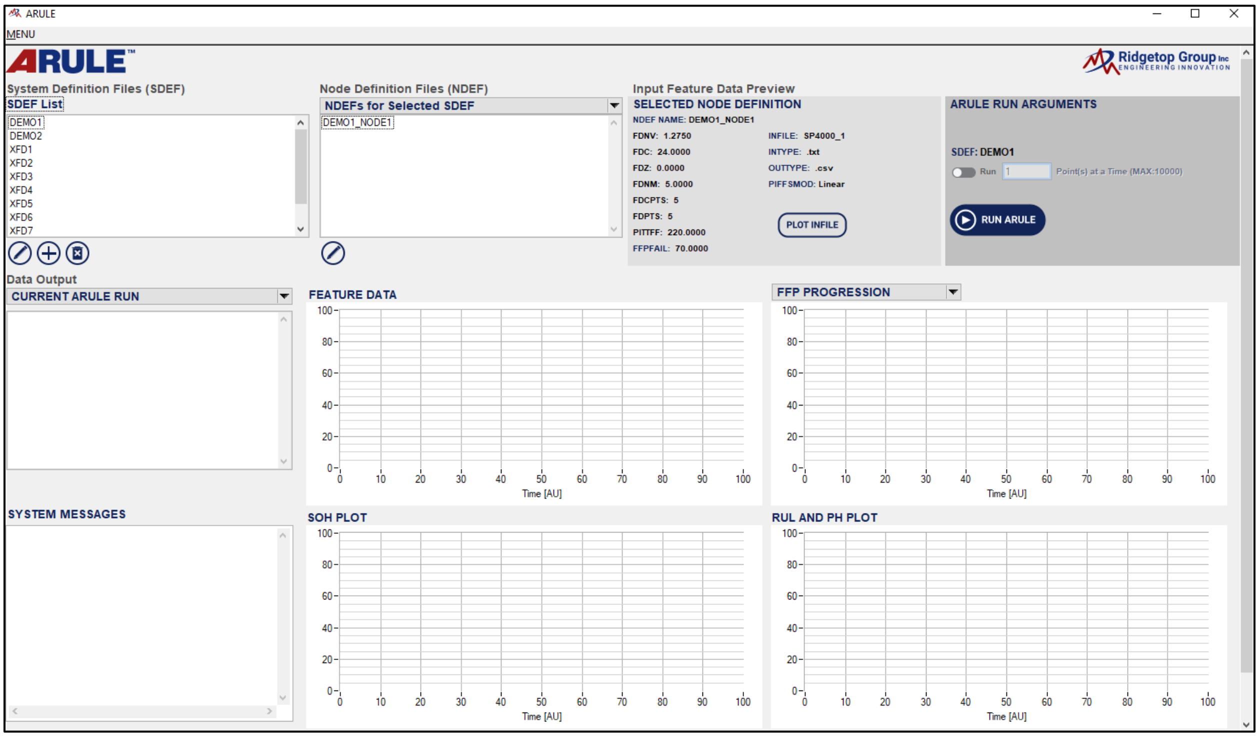


Figure 48 – Single Window GUI: System Definition Input and Data Output

An outline of the 5 main program sections is as follows:

<u>Single Window GUI Sections:</u>

System Definition Input:

1. **System Definition Files (SDEF)** – A list of available System Definition (SDEF) files.
2. **Node Definition Files (NDEF)** – A dropdown option for a list of Node Definition (NDEF) files for the selected SDEF file, or a dropdown option for all available Node Definition (NDEF) files.
3. **Input Feature Data Preview** – A visual of file contents for selected NDEF file and corresponding keyword parameters.

4. **ARULE Run Arguments** – User specified program run arguments to process a specified number of data points or use the default option to process all input data points.

Data Output

5. **Data Output –** Two dropdown options that will display a scroll box containing a list of data outputs for either the CURRENT ARULE RUN or all previously SAVED ARULE RUNs.
6. **SYSTEM MESSAGES** – A text preview box for program messages.

5.2 System Definition Input

The System Definition Input section of the ARULE™ GUI, is where the user will select, create, and edit the System Definition files, Node Definition files, and specify the ARULE™ run arguments. This section offers a combination of scroll boxes, user forms, a **PLOT INFILE** button, and a **Toggle** switch for the ARULE run arguments. The PLOT INFILE button will provide a preview of the Condition–Based Data input plot that is linked to a data input file that is specified in the selected Node Definition file, which can be in either .csv or .txt format: see

Figure 49.

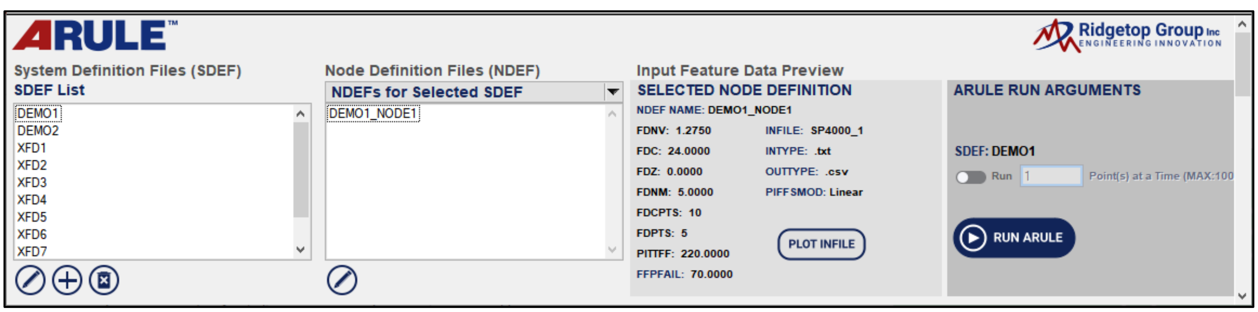


Figure 49 – Visual of the System Definition Input

5.2.1 System Definition (SDEF) Files

The **SDEF List** is where the user can define, edit, and delete System Definition (SDEF) files. An SDEF file is defines a system of nodes that each have a specific node number ID (NDNUMID) and a corresponding Node Definition (NDEF) file name (NDFNAME). Each NDEF file contains key information and values for a particular system node. An SDEF file can contain up to ten nodes (i.e. NDEFs) for a given system. An example SDEF filename that contains one system node is **DEMO1.txt**. (see Figure 50). Upon right–clicking on the required SDEF file (in this case, **DEMO1**), the user can open it in the editor, locate it in File Explorer, delete it, sort it among other files, and find a particular file.

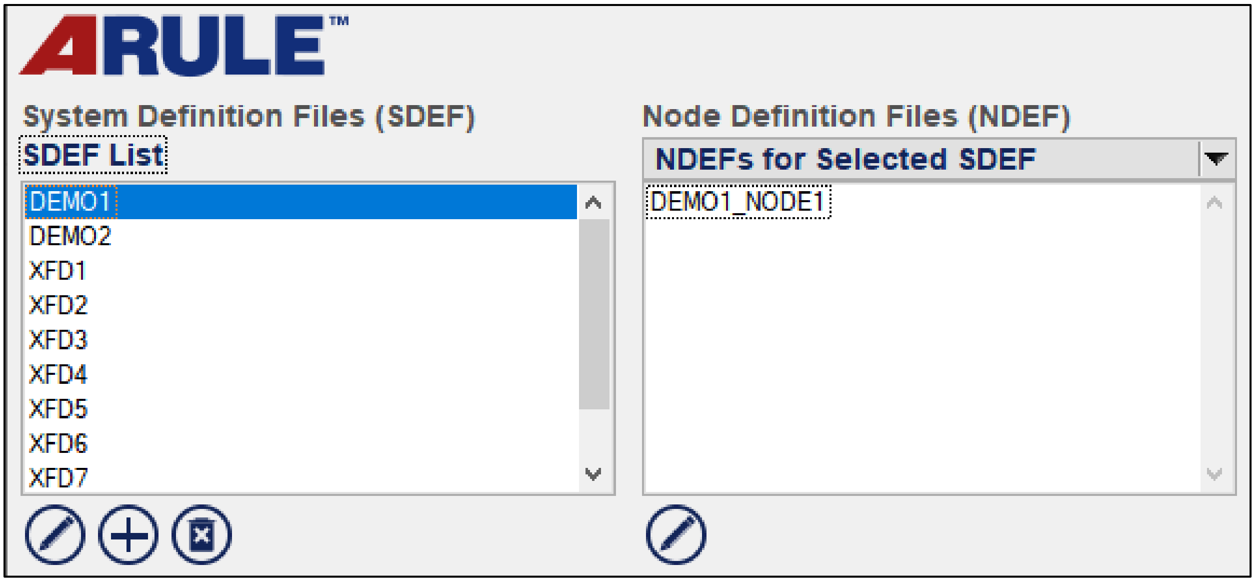


Figure 50 – Visual of the SDEF List section for an SDEF file on the GUI

ARULE™ has the following prebuilt list of SDEF files in a scroll box in the SDEF List section. Each of these SDEF examples will be covered in Section 6 of this User Guide as they are based on real–world applications.

- DEMO1.txt
- DEMO2.txt
- XFD1.txt
- XFD2.txt
- XFD3.txt
- XFD4.txt
- …
- XFD9.txt

SDEF files can be created/edited 1 of 2 ways. The first method is to click on the **Circle Plus** Button ⊕ (to create a new SDEF file) or click on the **Circle Pencil** Button (to edit an existing SDEF file) located on the lower–left side of the **SDEF List.** Here the user has text inputs to specify the SDEF file name and **+/X** buttons to add/remove its corresponding NDEF files. The program then automatically assigns the node ID and NDEF file for the node(s) in that SDEF file. The program also supports freeform editing in an external text editor program such as Notepad, WordPad, Notepad++, and so on. This second editing method is enabled by right–clicking on the selected SDEF file in the scroll box of the SDEF List section and selecting **Open in Editor**. Section 7 will cover the specific procedure for creating a new SDEF and NDEF file that points to a user defined data set. An example of the System Definition User Form, and the corresponding .txt file in the external Notepad program are shown in Figure 51 and Figure 52, respectively.

Important Note: ARULE™ software does not support nor recommend simultaneously editing definition files with the user form and an external program.

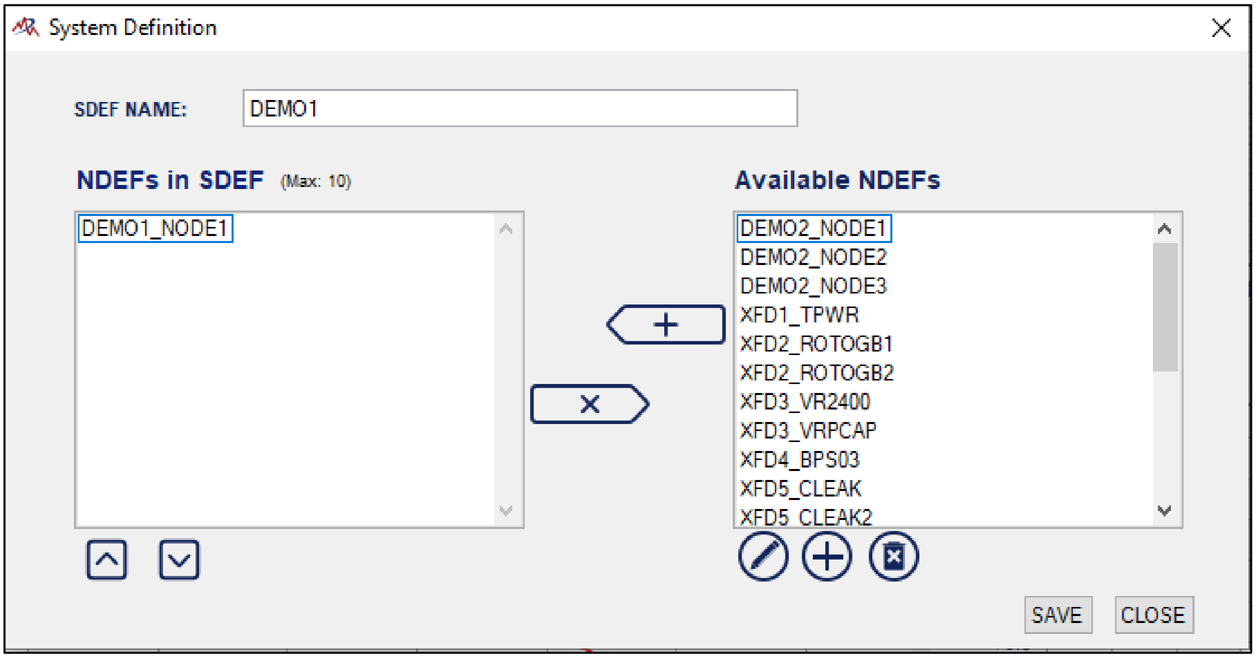
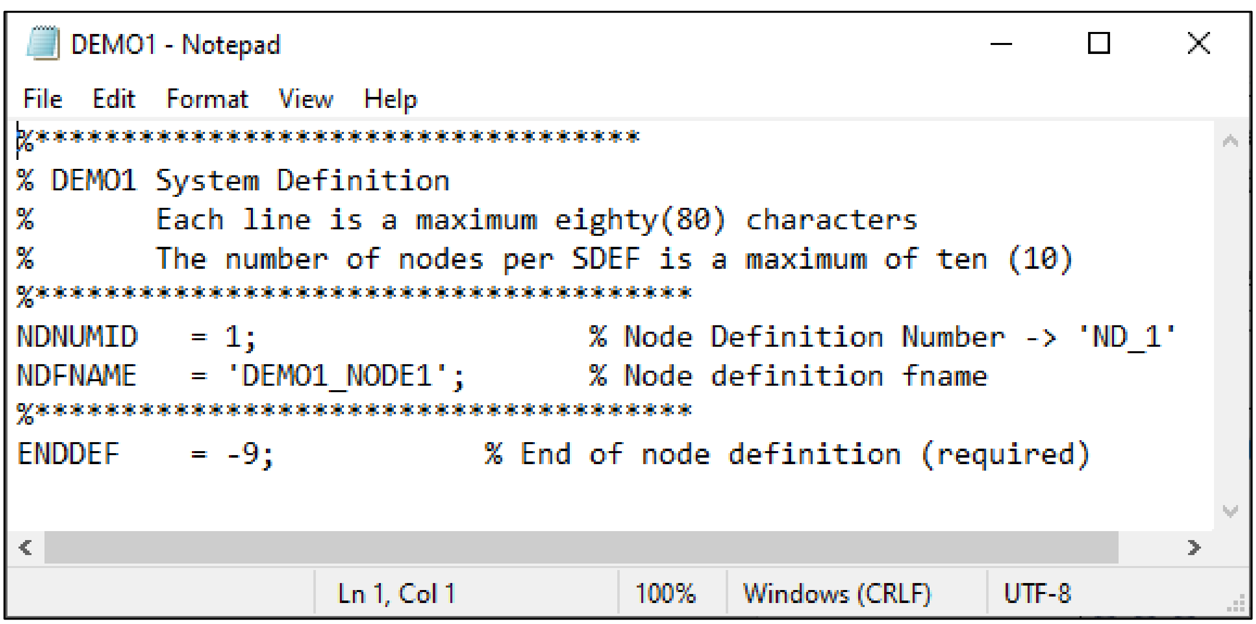


Figure 51 – Visual of the System Definition User Form




```
%************************************
% DEMO1 System Definition
%       Each line is a maximum eighty(80) characters
%       The number of nodes per SDEF is a maximum of ten (10)
%**************************************
NDNUMID    = 1;                % Node Definition Number -> 'ND_1'
NDFNAME    = 'DEMO1_NODE1';    % Node definition fname
%**************************************
ENDDEF     = -9;           % End of node definition (required)
```

Figure 52 – The corresponding SDEF file in .txt format

Here, the first node is specified to have a NDNUMID of 1, and the detailed specifications for this node are in the NDEF file that is named **DEMO1_NODE1**.

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

Important Note: There is no requirement that a specified NDNUMID, a string value, to have any correlation to its relative position (First, Second, ... nth) position in an SDEF file.

5.2.2 Node Definition (NDEF) Files

The Node Definition (NDEF) Files section of the GUI has a scroll box for two drop-down options associated with **NDEFs for Selected SDEF** or **All NDEF** files. See a visual for both options in Figure 54 and Figure 54. This section is where the user can add/edit and preview each NDEF file that is specified in the SDEF file. An NDEF file will allow the user to define values for all keywords and data processing parameters for a given system node.

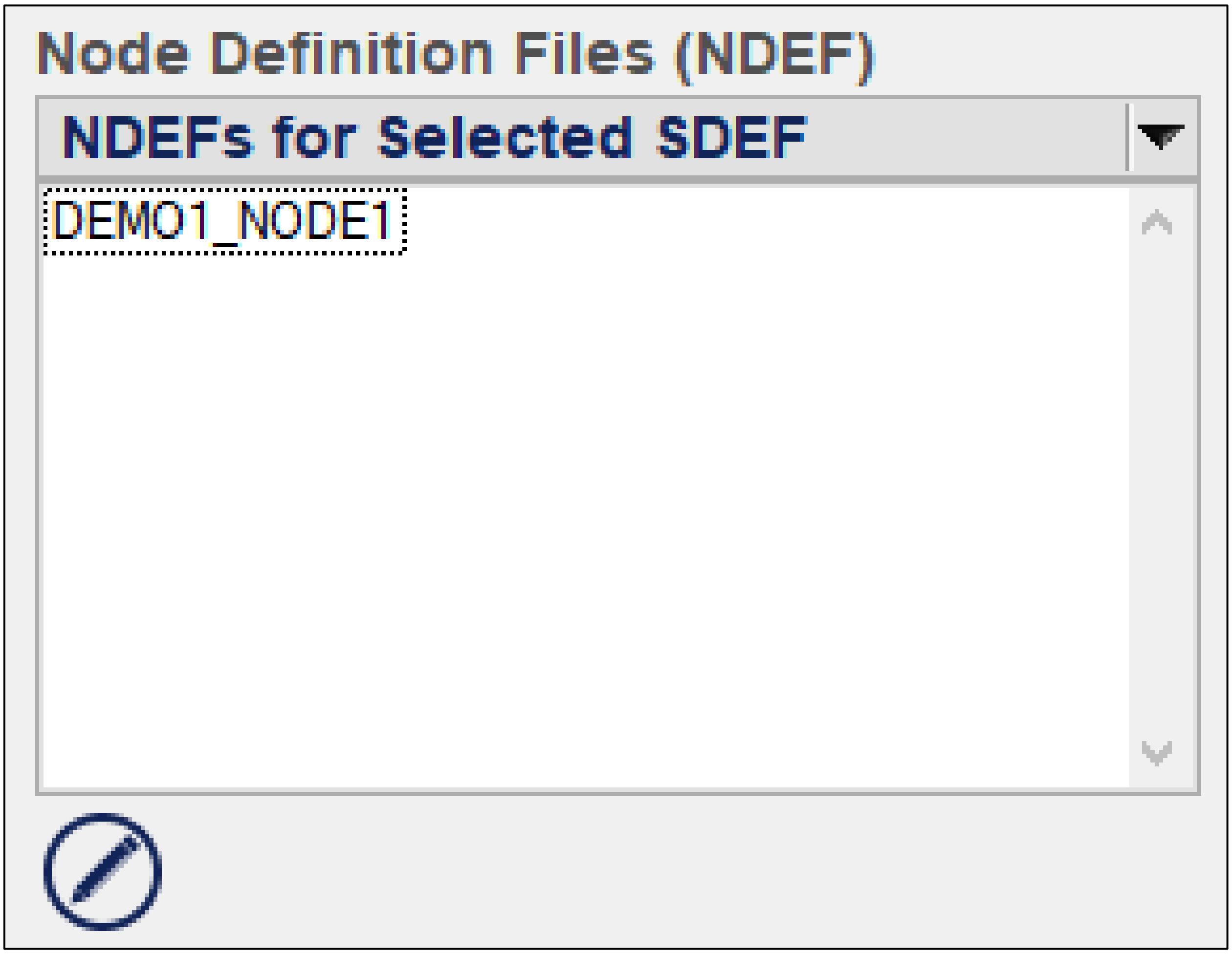


Figure 53 – Visual of the NDEF file scroll box on the main window

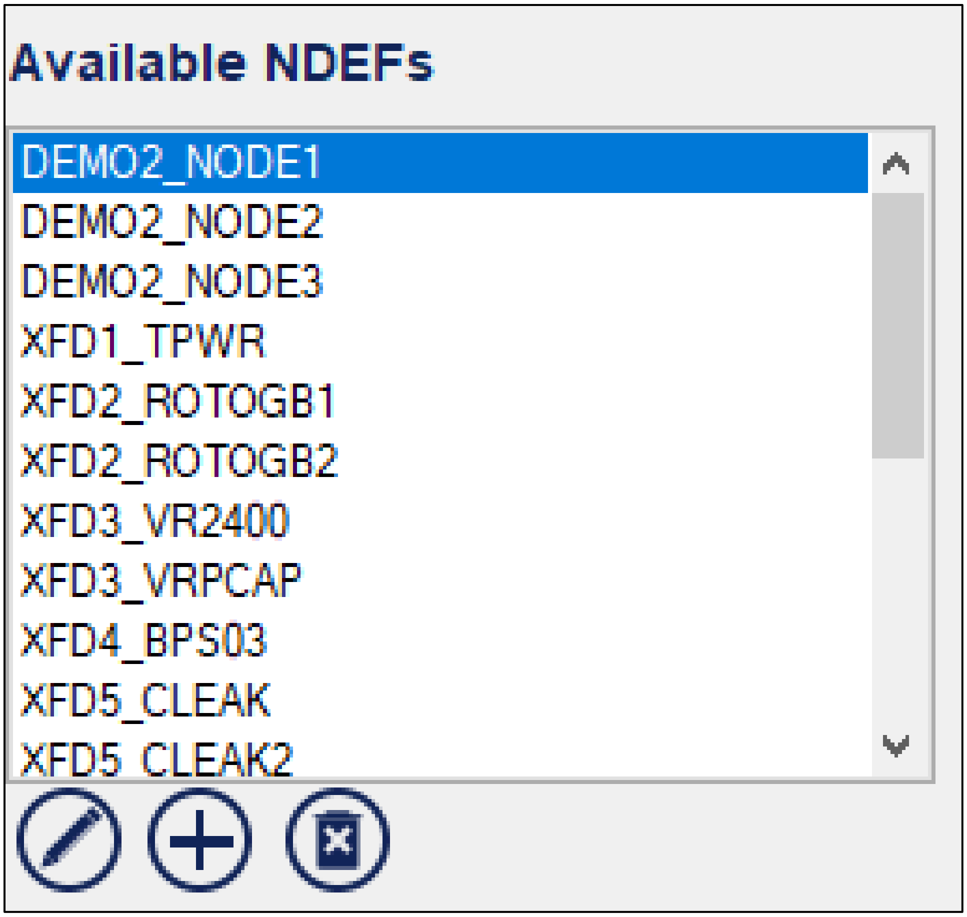


Figure 54 – Visual of the NDEF file scroll box

To add/edit the NDEF file the user can click on the **Circle Plus** Button (to create a new NDEF file) or click on the **Circle Pencil** Button (to edit an existing NDEF file). Both icons are located on the lower–left side of the **Node Definition Files** scroll box and will open the NDEF file in a separate pop–up defined as the Node Definition User Form. Here, the user form allow the user to specify the Node Definition name, the Condition–Based Input Data filename, and various other operational parameters. Alternatively, the user can also edit the NDEF in a text editor by right clicking on the selected NDEF file in the **NDEFs for Selected SDEF/ All NDEFs** section and selecting **Open in Editor**. See Section 4.2.3 for more information on this editing method.

Important Note: Ridgetop recommends that end users edit their NDEF files using the user forms, as it protects against typing errors for those that are not familiar with the BNF syntax outlined in Section 2.4 of this User Guide. Also note that the ARULE™ software does not support nor recommend simultaneously editing definition files with the user form and an external program.

5.2.3 Node Definition User Form

A visual for the Node Definition User Form is shown in Figure 55.

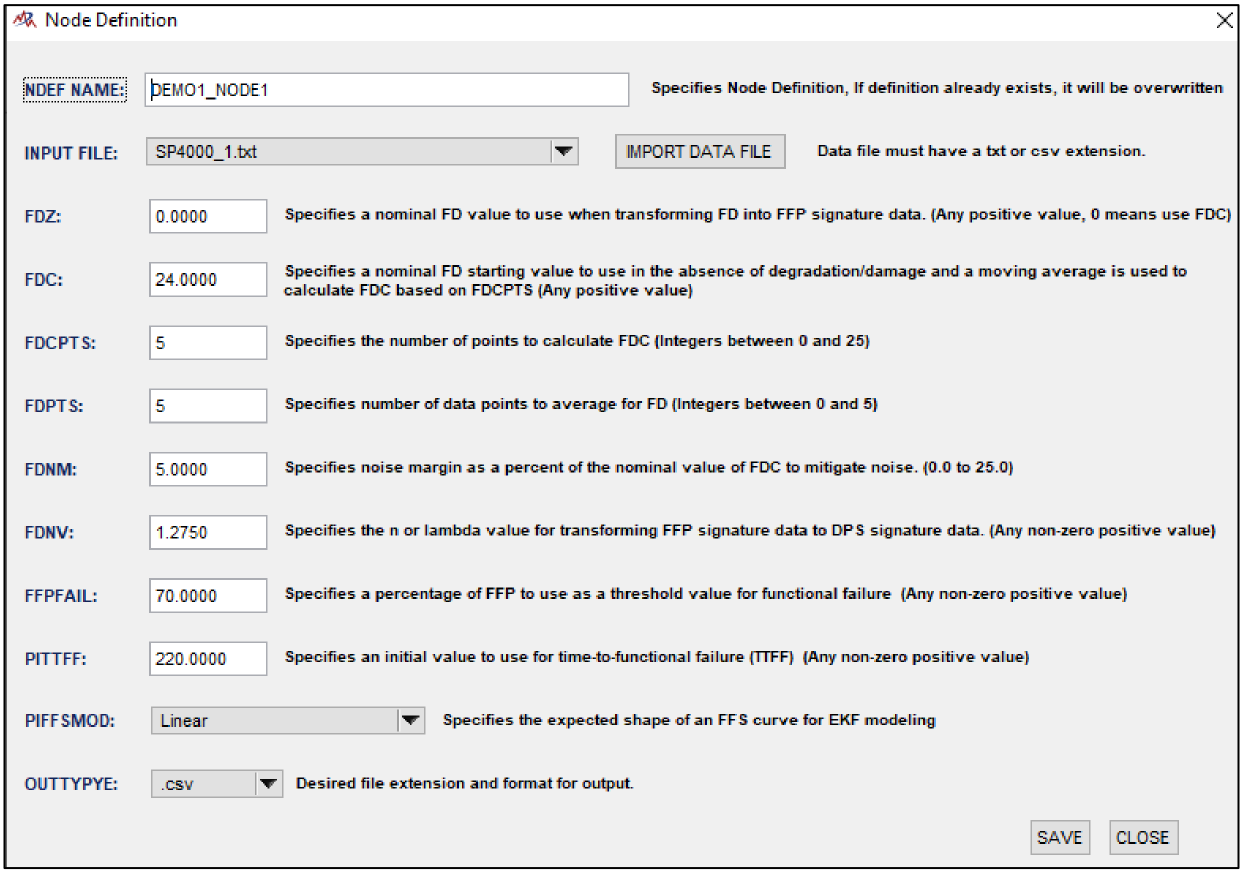


Figure 55 – Visual of the Node Definition User Form

The Node Definition User Form displays the NDEF file name that was specified in the SDEF file. The user form also allows the user to select/specify the input data file using the **IMPORT DATA FILE** button and the Windows File Explorer. Once an input file is selected, the program then automatically sets the INFILE and INTYPE keywords in the NDEF file. For this example, the **DEMO1_NODE1** file (NDEF file) references **SP4000–1.txt**, which is located in following default directory: **C:\Users*username*\My Documents\ARULE\DATA\DINP**

Important Note: The input data file and type can be specified as either .csv or .txt; and, for usability, the prebuilt NDEF files in ARULE™ are named using the following convention: SDN_IFN(n), where SDN is the filename of the SDEF that uses the NDEF file, IFN is the filename of the input file specified in the NDEF, and (n) is used when two NDEF files use the same input file in a single SDEF. DEMO1_SP4000–1 is an example of the filename for a prebuilt NDEF. XFD1_TPWR.txt and XFD1_TPWRB.txt are NDEF examples that use the same input file for the XFD1 SDEF.

The Node Definition User Form then categorizes NDEF values and parameters based on Feature Data, Data Conditioning, Degradation signatures, and Prognostics. There is a total of 9 parameters, (not including the name of input file) that control how the condition–based data (CBD) input is processed, conditioned, and transformed into a Fault to Failure Signature (FFS) data set. A brief outline of these data parameters is detailed below.

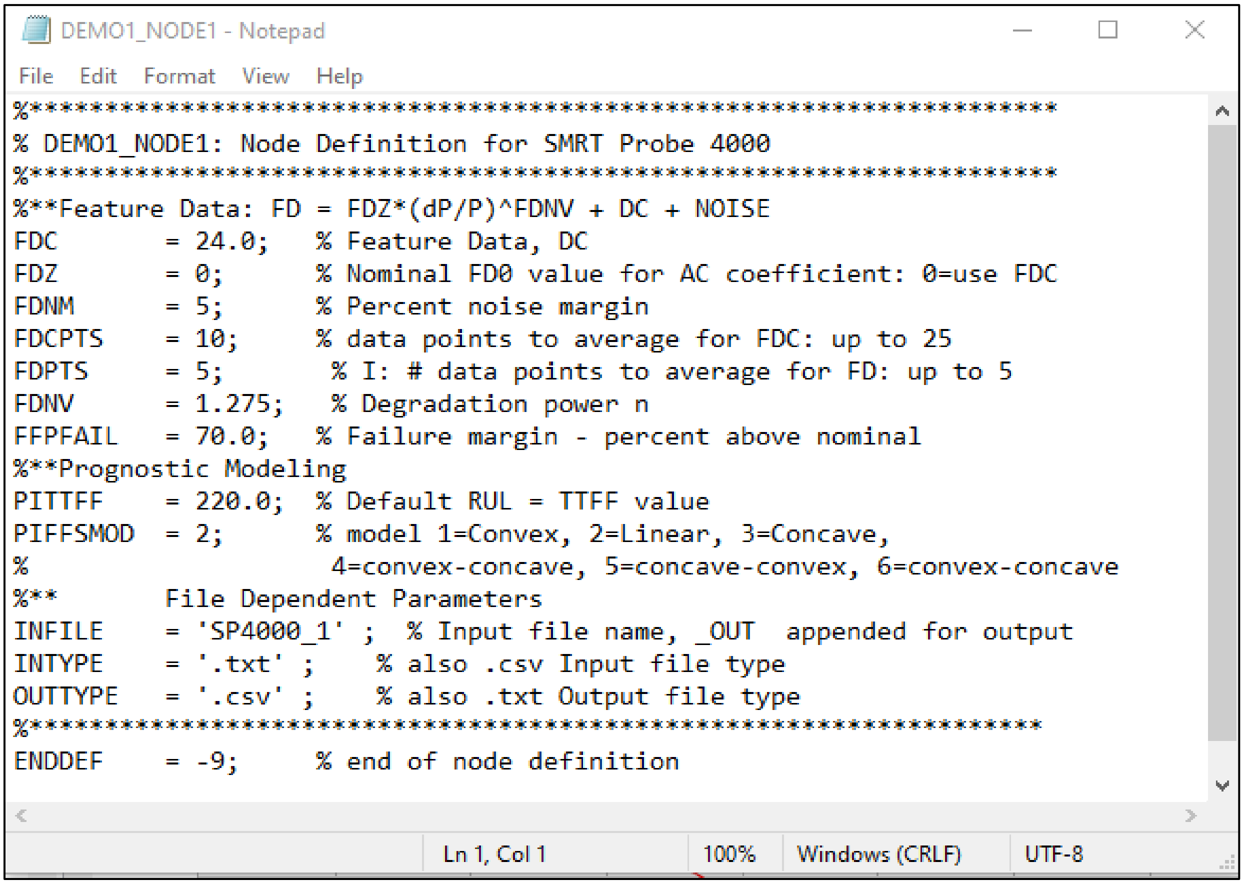
Feature Data, Data Conditioning, Degradation Signatures, and Prognostics Parameters:

- **FDZ** – Feature Data Nominal Value as observed in your data input file.
    - Any positive value and sets FD NOM in data output file.
    - 0.0 (Use FDC Value)
- **FDC** – Feature Data Nominal Value in the absence of degradation/damage.
    - Any positive value and is used to continuously calculate FD NOM based on FDCPTS.
    - 1.0 (default)
- **FDCPTS** – Number of data points to calculate FDC.
    - Integer value 0.0 to 25.0
    - 10.0 (default)
- **FDNM** – FD Noise Margin as a percent of the Nominal Value of FDC.
    - 0.0 to 25.0
    - 5.0 (default)
- **FDNV** – n or lambda Value for transforming FFP Signature data to DPS Signature data.
    - Any non–zero positive value
    - 1.0 (default)
- **FDPTS** – Number of points to average for FD.
    - 1 to 5
    - 5 (default)
- **FFPFAIL** – FFP value at which functional failure occurs.
    - Any non–zero positive value
    - 70.0 (default)
- **PITTFF** – Initial value to use for initial value to use for Time–to–Functional Failure (TTFF): becomes RUL and PH in the absence of degradation.
    - Any non–zero positive value
    - 200.0 (default)
- **PIFFSMOD** – Functional Failure Signature Model (FFSMOD) to use when processing input FFS data shown in Figure 22.

- Linear (default)
- 1: concave, 2: linear, 3: convex, 4: convex–concave, 5: concave–convex;

5.2.4 Raw NDEF File in Notepad

An example of all NDEF parameters being displayed in a raw .txt file with Notepad is shown in Figure 56.




```
%*********************************************************************
% DEMO1_NODE1: Node Definition for SMRT Probe 4000
%*********************************************************************
%**Feature Data: FD = FDZ*(dP/P)^FDNV + DC + NOISE
FDC        = 24.0;   % Feature Data, DC
FDZ        = 0;      % Nominal FD0 value for AC coefficient: 0=use FDC
FDNM       = 5;      % Percent noise margin
FDCPTS     = 10;     % data points to average for FDC: up to 25
FDPTS      = 5;       % I: # data points to average for FD: up to 5
FDNV       = 1.275;   % Degradation power n
FFPFAIL    = 70.0;   % Failure margin - percent above nominal
%**Prognostic Modeling
PITTFF     = 220.0;  % Default RUL = TTFF value
PIFFSMOD   = 2;      % model 1=Convex, 2=Linear, 3=Concave,
%                     4=convex-concave, 5=concave-convex, 6=convex-concave
%**        File Dependent Parameters
INFILE     = 'SP4000_1' ;  % Input file name, _OUT  appended for output
INTYPE     = '.txt' ;    % also .csv Input file type
OUTTYPE    = '.csv' ;    % also .txt Output file type
%********************************************************************
ENDDEF     = -9;     % end of node definition
```

Figure 56 – Example of an NDEF being edited in raw .txt by Notepad

Important Note: Ridgetop Group recommends that the GUI user forms be used to edit and save NDEF and SDEF files.

5.2.5 Input Feature Data Preview

This section displays the Condition–Based Data input plot for a selected NDEF file. This can be achieved by clicking on the **PLOT INFILE** button in the **SELECTED NODE DEFINITION** under the **Input Feature Data Preview** section as shown in Figure 57, to open the Input Data Plot Preview tab.

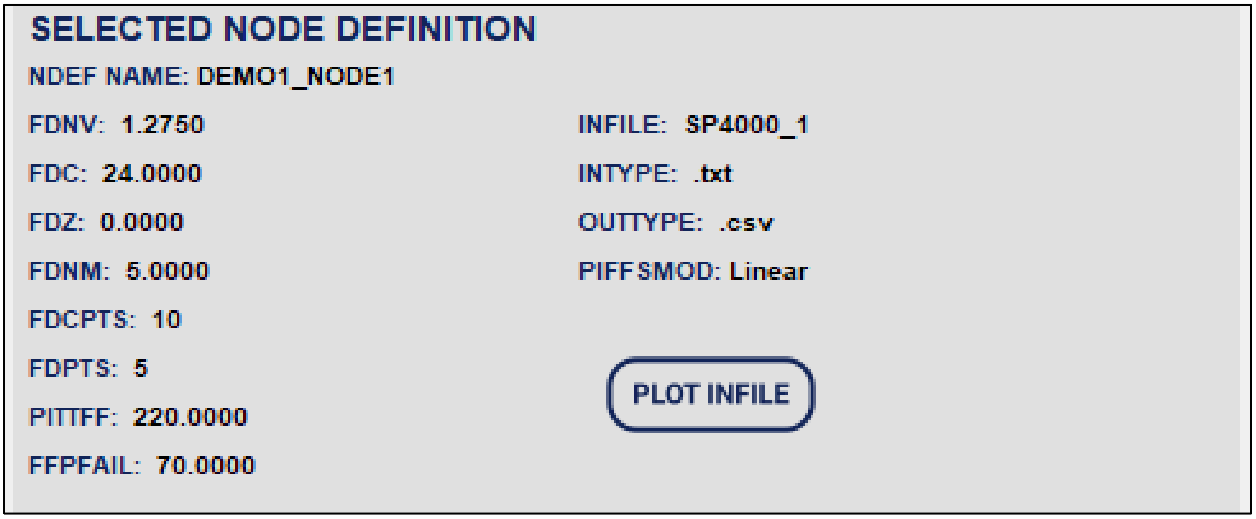


Figure 57 – Plot Input File feature in the SELECTED NODE DEFINITION section

Figure 58 provides a visual for the **SP4000–1.txt** input data file after clicking the **PLOT INFILE** button:

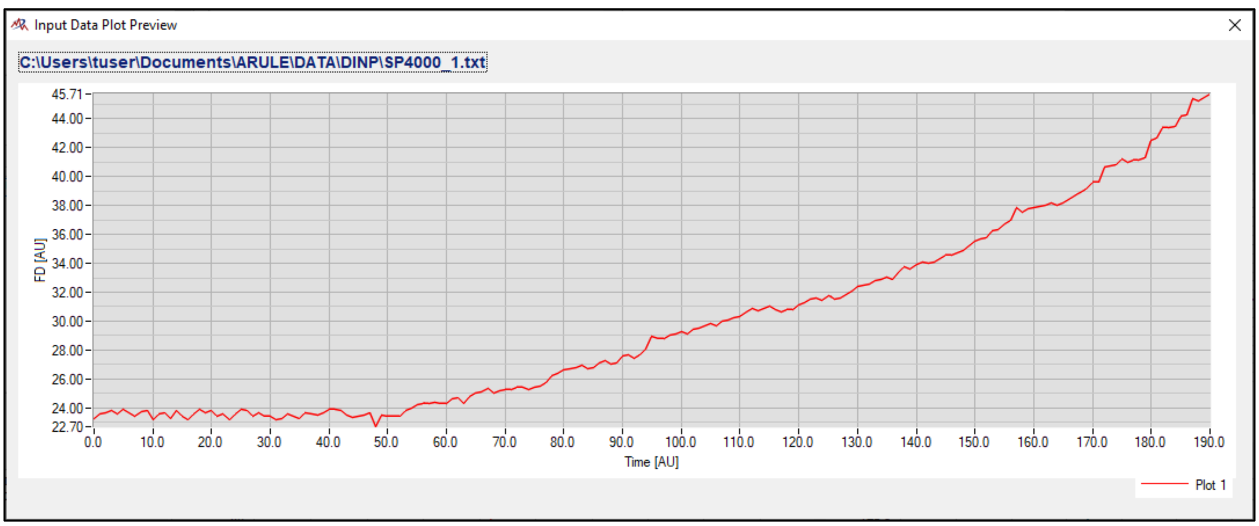


Figure 58 – Visual of CBD Input Data Plot Preview tab

The blue line at the top of this plot preview shows the full path for the CBD input data file that was defined in the NDEF file.

5.2.6 ARULE™ Run Arguments

This section of the ARULE™ software program will show a label for the SDEF file that will be processed by the ARULE™ APK, and allows the user to specify 2 input argument before attempting a prognostic estimation run.

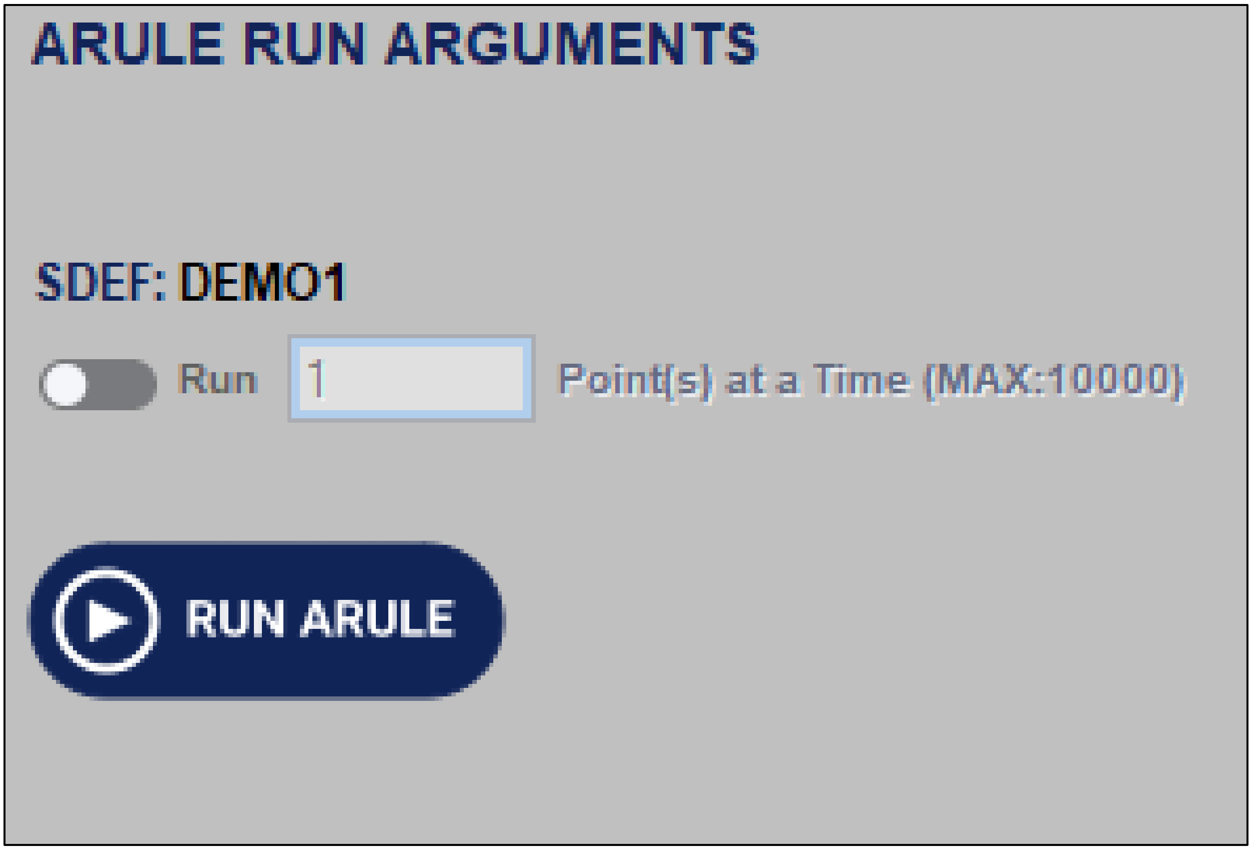


Figure 59 – ARULE™ Run Arguments section

- Argument 1 – is the name of the SDEF file you selected, and is displayed as a label.
- Argument 2 – is a Toggle Switch that enables Check Point Restart (CPT) mode to process a specified number of data points.
    - When the Toggle Switch is disabled, the program is in the Default Processing Mode and will process all data points for the input data files that were defined in the NDEF file.
    - When the Toggle Switch is enabled, the program is in Check Point Restart (CPT) Mode. In CPT mode, the user has the ability to process N# of data points between 1–10000 in the text input as shown in Figure 60.

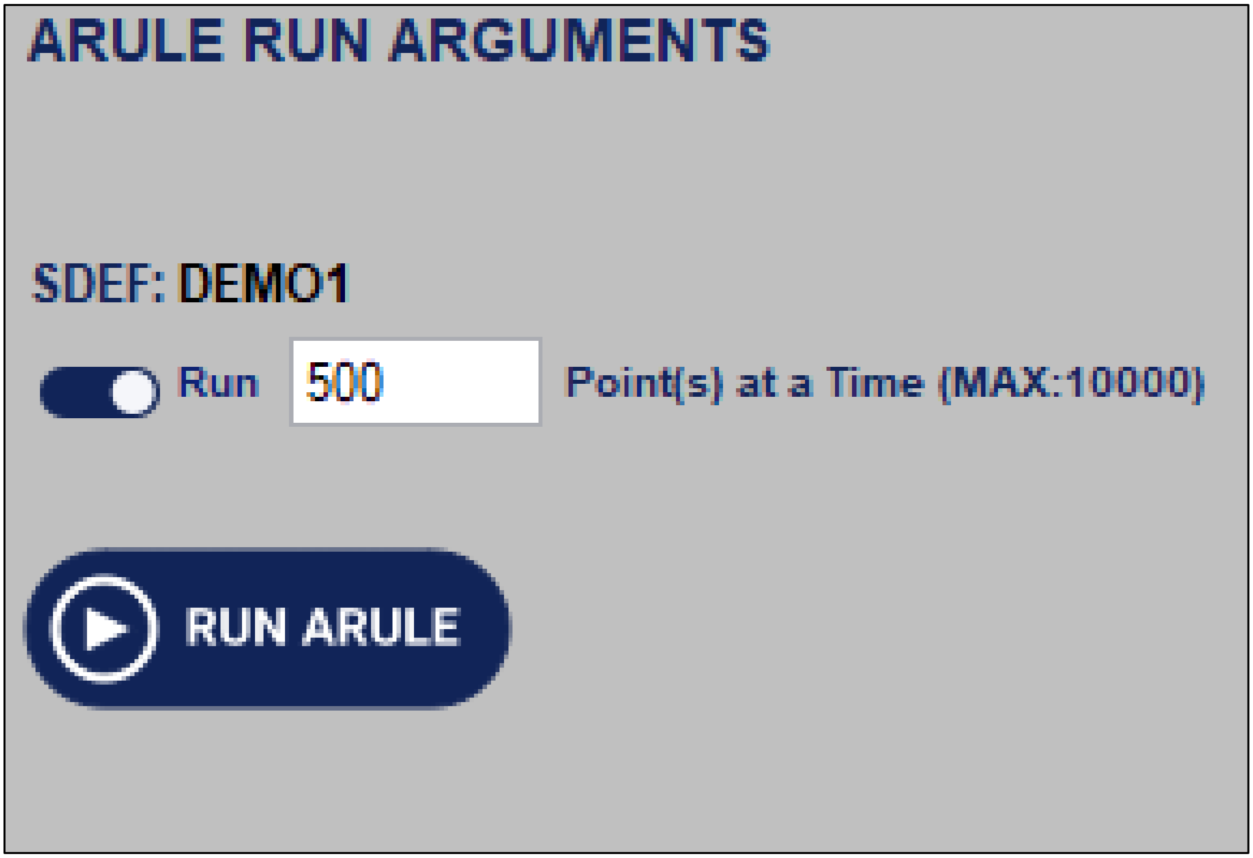


Figure 60 – Example of enabling CPT mode and wishing to process 500 data points

Enabling CPT Mode demonstrates the ability to call the ARULE™ APK multiple times to process the same data set at different instances in time. When CPT Mode is enabled, you many continue to process another set of input data points; or you may return to change the number of input data points to process; or you may choose to exit CPT Mode by processing a different SDEF and NDEF selection.

System messages will also appear in the lower left corner of the GUI, and specific reason codes can be reviewed the log files here: **C:\ARULE™\DATA\LOG.**

Upon completing a successful prognostic estimation run with the ARULE™, there will be a **"ARULE APK program has ended"** message shown in the **SYSTEM MESSAGES** section as in Figure 61 and the program outputs the key prognostic plots.

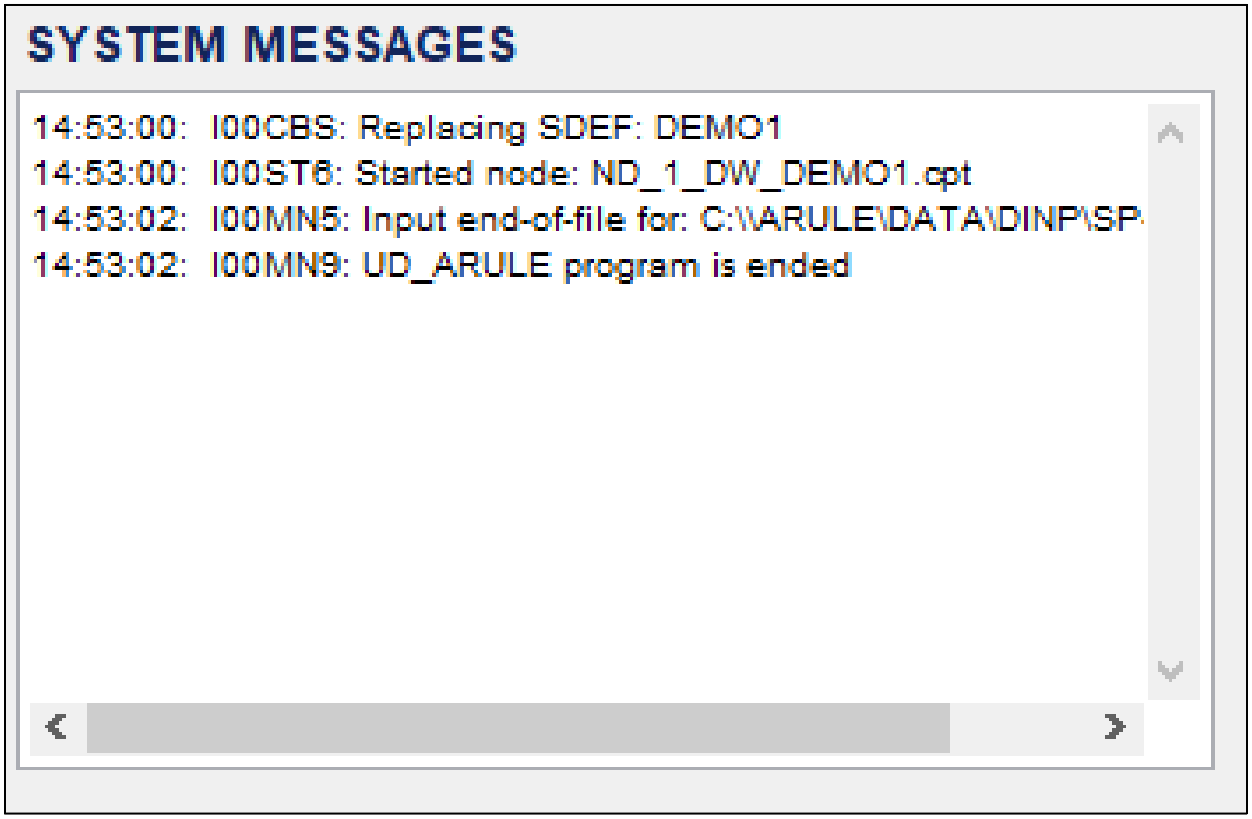


Figure 61 – Example of a system message: successful processing

5.3 Data Output

The Data Output section on the main window (Figure 62) shows a graphical representation of data output files for all processed data sets that were referenced in the NDEF file. When the data output files are selected using either the **CURRENT ARULE RUN or SAVED ARULE RUN** dropdown options, then four default data output graphs will populate for the selected data output file: 1) **FEATURE DATA**; 2) **FFP PROGRESSION**; 3) **SOH PLOT**; and 4) **RUL AND PH PLOT**. Note that the second graph has a drop-down option to toggle from the **FFP PROGRESSION** plot to the **DPS, FFS,** and **FFIN PROGRESSION** plots. These plots represent the different signature transforms within ARULE as covered in Section 2.5.

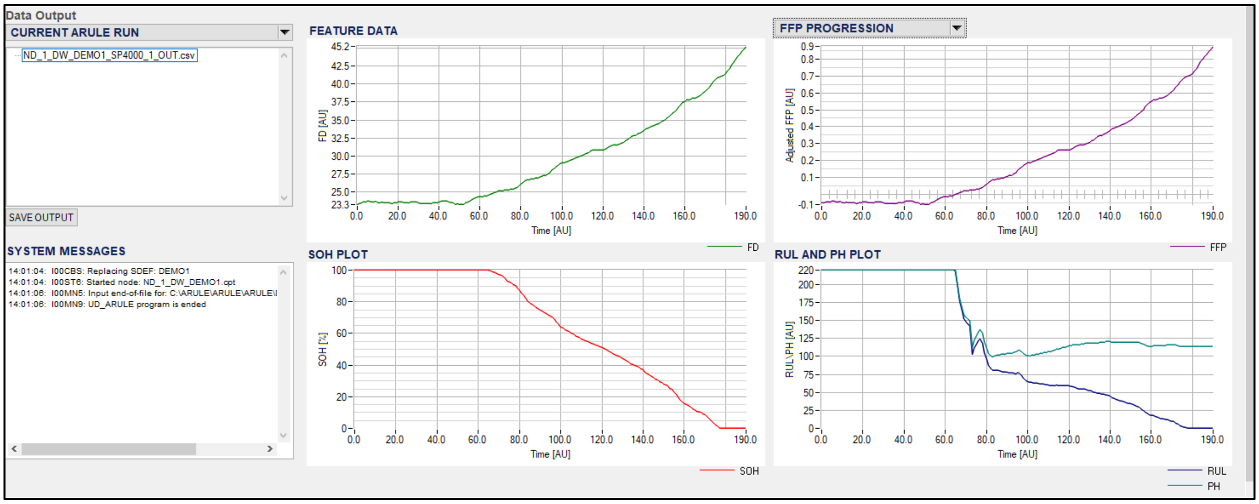


Figure 62 – Example of a Data Output section

5.3.1 Data Output Dropdown Options

This section of the GUI displays two drop-down options and a corresponding scroll box that contains the data output for the CURRENT ARULE RUN or SAVED ARULE RUNS that were produced from each ARULE™ prognostic estimation run. Each call of the program will produce data output files in either .txt or .csv format as defined in the processed NDEF file.

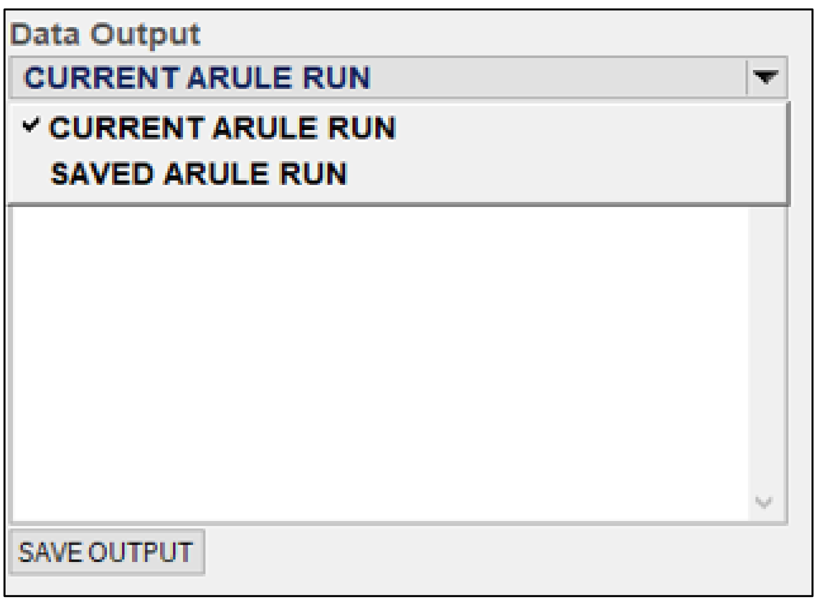


Figure 63 – View of Data Output dropdown options.

The user also has the option to save the data output information to a user specified data output directory by clicking on the **SAVE OUTPUT** button. By default, these files will be downloaded/saved in a default directory in a timestamped folder as shown in Figure 65.

All saved ARULE runs are stored in the following default directory: **C:\Users*username*\My Documents\ARULE\DATA\SAVE.**

Important Note: The defined user path was set during the original installation process and can be reset at any time using the Menu option documented in Section 4.10.

All output data and plot information can be saved for future analysis or export into another program. As previously covered, this is achieved by Right Clicking on the saved output file to open the directory using the Windows File Explorer (See Figure 64 – Figure 65)

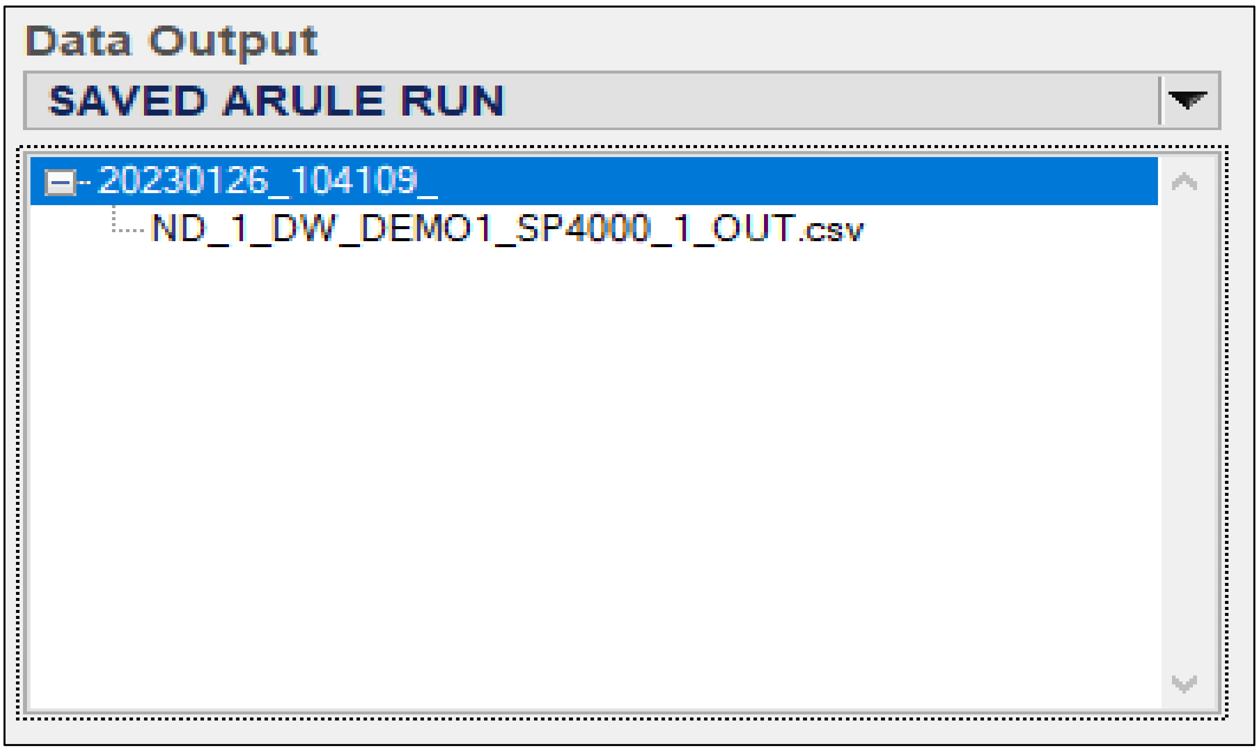


Figure 64 – View of dropdown option for SAVED ARULE RUN

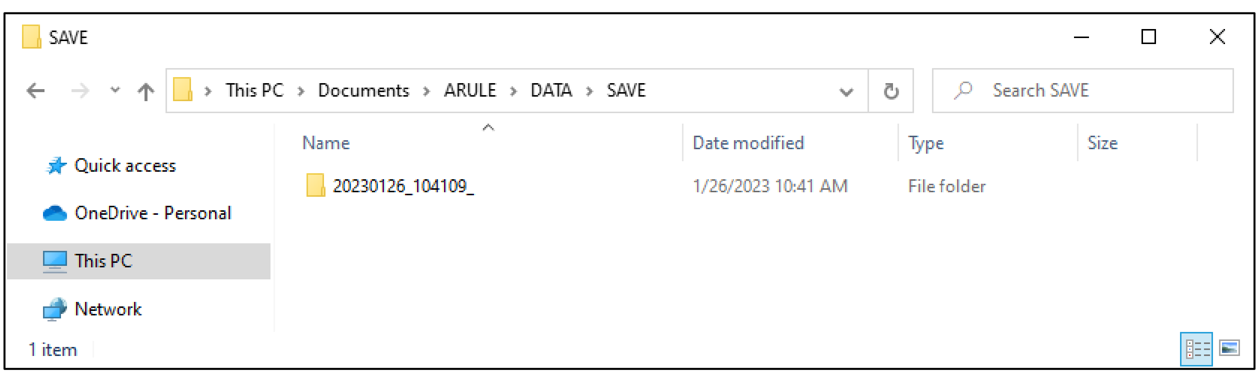


Figure 65 – Example of downloaded plot information in a time stamped folder

5.3.2 Data Outputs in .csv File

The Input-Output (IO) Bus of the ARULE API is the primary means by which an application program provides input data to ARULE and the means by which ARULE returns prognostic information to the application. RUL, SOH, and PH values are prognostic information useful in determining state of health and for predicting a future time of failure. The other values are provided for the following reasons: (1) support evaluation, including verification and validation, of ARULE results and performance; (2) support the creation of visualization aids, such as plots, by an application; and (3) support problem determination of incorrect parameter and other unexpected conditions (return and reason codes). The IO Bus is also used to provide return and reason codes for verification and validation of specified parameter values and for problem determination, accuracy evaluation and analysis, and performance evaluation by an application program. Table 4 lists the identifier (ID), a short description and other information for the values in the ARULE APK IO Bus.

Table 4. Data Output Values

ID	DESCRIPTION	Notes
FLAG	Flag for internal program operation.	-
DT	Data Time	1, 6
DA	Data Amplitude	1
RUL	Remaining Useful Life	1
PH	Prognostic Horizon	1, 2
SOH	State of Health	1
BD	Begin of Degradation	1, 3
EOL	End of Life	1, 3
FDNOM	FD-nominal value (Set by FDZ or calculated by FDC & FDCPTS)	1
FD	Feature Data	1, 5
FFP	FFP signature data	1, 5
DPS	DPS signature data	1, 5
FFS	FFS signature data	-
FFIN	Functional Failure input data to ARULE prognosis	-
RC0	Highest Return Code for this data point	7
RS0	Reason for RC0	**Table 5**
RC1	First non-zero RC	7
RS1	Reason for indicated RC	

ID	DESCRIPTION	Notes
		Table 5
FLAG	Reserved flag for internal program operation.	-

Note 1: All values, except SOH, are arbitrary units of measure. It is the responsibility of the application to keep track of, for example, that a DT value of 8.5 means 8.5 hours, days, and so on.

Note 2: SOH values are percent values, for example, a SOH value of 92 means 92% healthy.

Note 3: -99 value indicates not detected

Note 4: -99 value indicates not estimated

Note 5: Data after conditioning/smoothing and prior to next signature transform

Note 6: Time value for input data must be larger than previous time value

Note 7: Return code values are 0 (No errors/no exceptions); 4 (warning); 8 (error – ARULE is unable to continue processing data); 12 (severe error – possible programming error, ARULE is unable to continue processing)

Table 5. Reasons: Value, Descriptions, Action, and Remedy

RS	DESCRIPTION	ACTION	REMEDY
2	not assigned	none	none
4	FDNM is a negative value or greater than 25.0	Changed to 5.0	Specify a positive value between 0 and 25.0
6	not assigned	none	none
8	FDC/FDZ is a negative value	Changed to -FDZ	Specify a positive value
10	Not assigned	none	none
12	Calculated value of FDC is not within FDCVAR limits	Specified FDZ value is used	Change FDZ and/or FDZVAR as applicable
14	Not assigned	none	none
16	Not assigned	none	none
18	Not assigned	none	none
20	FDXPTS is a negative value or greater than 25	Changed to -FDXPTS and/or to 25	Specify a positive integer value up to 25
22	Not assigned	none	none
24	not assigned	none	none
26	Not assigned	none	none

28	not assigned	none	none
30	FFPFAIL is not greater than 0.0	Changed to 70.0	Specify a non-zero positive value
32	not assigned	none	none
34	not assigned	none	none
36	PITTFF is not greater than 0.0	Changed to 200.0	Specify a non-zero positive value
38	PIFFSMOD is not a recognized model number	Changed to 2	Specify a valid FFS-model number
40	Not assigned	none	none
99	Functional Failure has occurred	Data processing continues	Repair/replace the prognostic target
100	Value for input data time is not increasing	Data processing is ended	Correct the incorrect value of time
110	Value is not a number	Data processing is ended	Correct the incorrect value
900 **9nn**	Unexpected operating condition	Program operation is ended	Report the problem to your system administrator

Important Note: As required, additional reason codes might be added.

5.3.3 System Messages

This section of the GUI displays a text box that populates with any system messages, error codes, and reason codes that may occur during program operation.

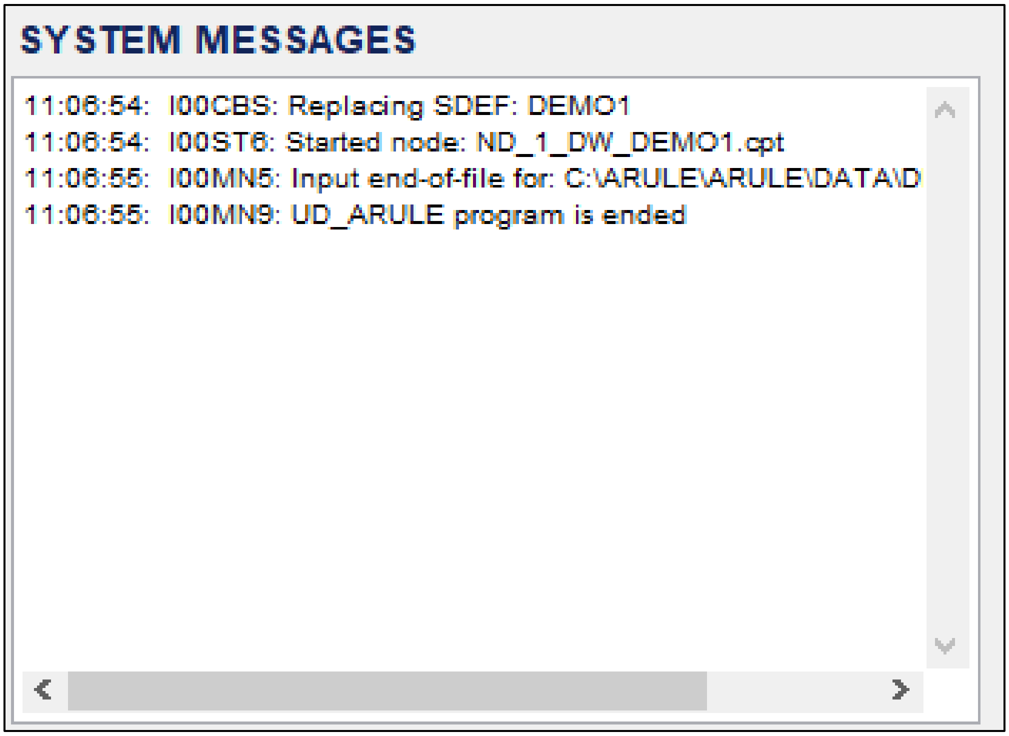


Figure 66 – View of System Messages

5.4 ARULE™ Global Features

5.4.1 Global Menu Options

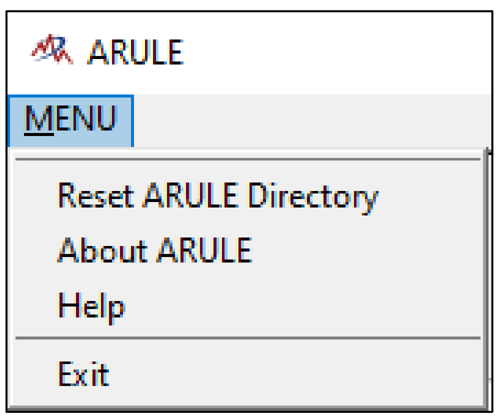


Figure 67 – View of Global Menu Options

There are four (4) global menu options in the ARULE™ GUI as shown in Figure 67. An overview of each option is outlined as follows:

- **Reset ARULE Directory**: Firstly, this opens up a pop-up window confirming with the user that they want to reset the ARULE directory or not, as shown in Figure 68. The user should click **Yes** to proceed or **No** to prohibit this action. This prompts to a new

File Explorer window where the user can specify the new location for the default folder where the ARULE directory needs to be placed. (Figure 69).

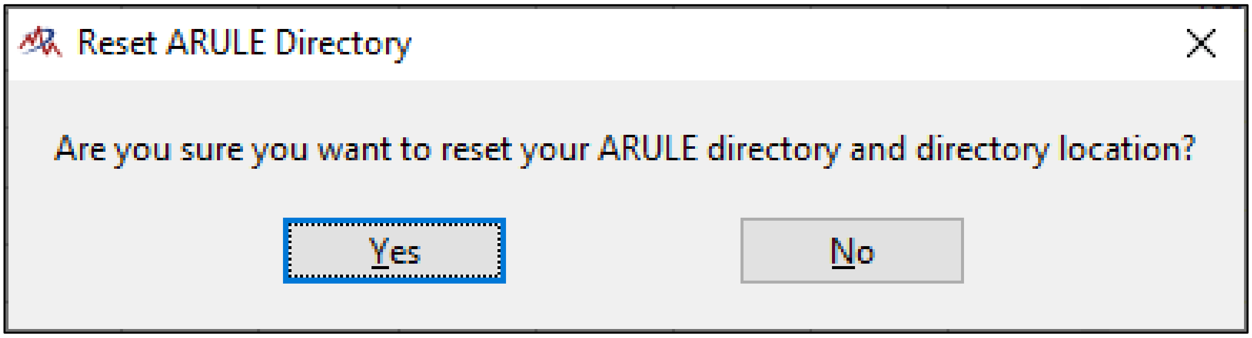


Figure 68 – First pop-up window when the Reset ARULE Directory option is selected

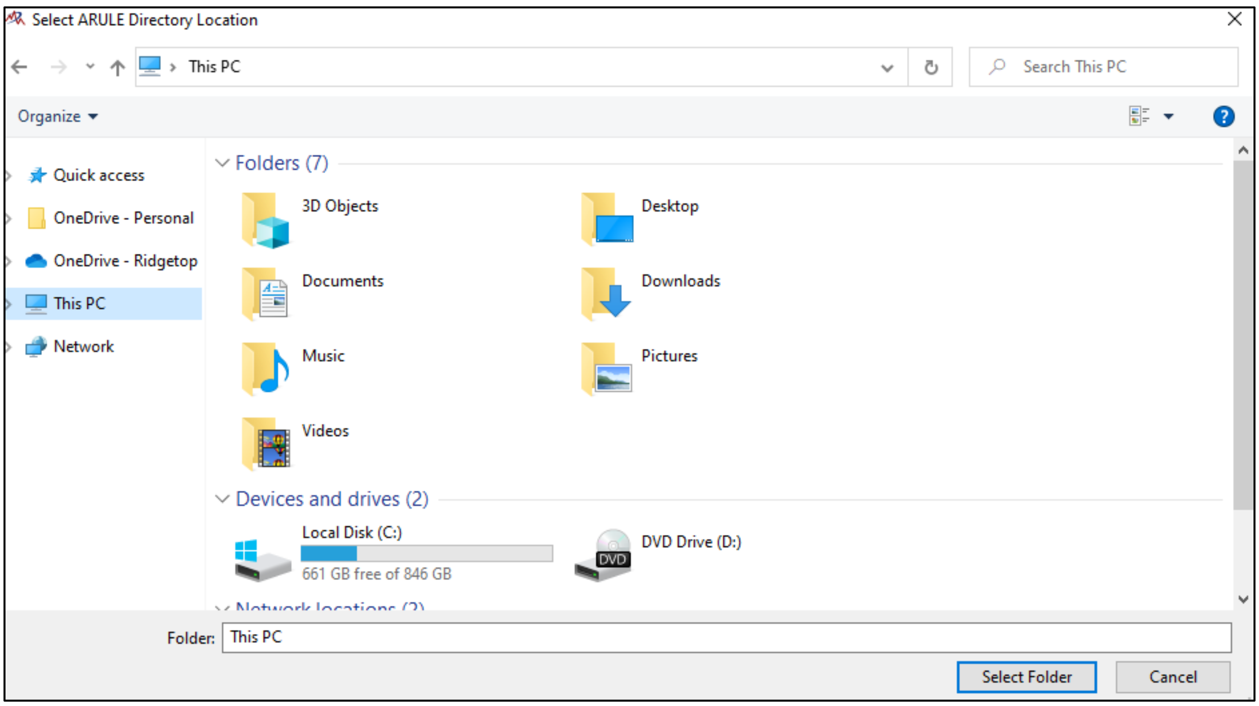


Figure 69 – File Explorer window to specify the new location for the ARULE directory

- **About ARULE**: This generates a pop–up window that tells you what version of ARULE™ is installed (Figure 70).

Figure 70 – View of About ARULE option when selected from the Menu

- **Help**: This opens a hyperlink to the User Guide.
- **Exit:** This prompts you to close and exit the program as shown in Figure 71.

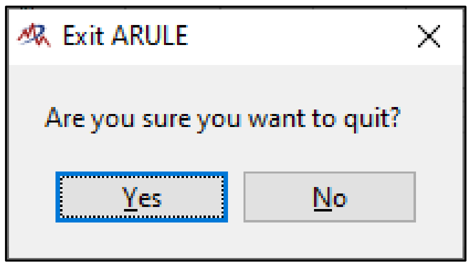


Figure 71 – Prompt to exit the ARULE™ application

6. Real World Examples

Node definitions are used to process condition–based Feature Data (FD) and the major steps in creating and using an NDEF file in ARULE™ are the following:

1. Acquire a set of example FD that progresses from no degradation to functional failure.
2. Create an initial set of node definitions that define:
   a. A characteristic curve applicable to the example data.
   b. Initial points where degradation is detectable and where functional failure is likely.
3. Run ARULE™ to process the example FD.
4. Evaluate the results.
5. As necessary, update node definition(s) and repeat steps three and four.

Step two in the above process comprises transforming condition–based data (CBD), also referenced as Feature Data (FD), into Fault to Failure Signature (FFS) data by selecting and defining NDEF keywords and values. The NDEF parameters then allow ARULE™ to transform FD signature data into FFP signature data, then into DPS data, and then into FFS data that is processed to produce prognostic information. The flow diagram shown in Figure 72 and the models shown in Table 6 summarizes the process of selecting applicable NDEF keywords and specifying values for those keywords. Further information is found in Section 2.

Transform $\{CBD\}$ into an $\{FFP\}$ signature
$FFP_i = ((FD_i - FD_0)/FD_0) \ - NM)$

Perform signal conditioning of the $\{FFP\}$ signature
Select degradation function curve to $\{FFP\}$ curve
$FFP_i = g(dP_i, P_0) - NM$

Select corresponding $f(FFP_i)$
Transform $\{FFP\}$ signature to $\{DPS\}$ data
$\{DPS\} = \{f(FFP_i)\}$

Plot to verify and evaluate transform results
Define a functional-failure threshold
Select corresponding $h(FL_{FFP})$ and calculate
$FL_{DPS} = h(FL_{FFP})$

Use FL_{DPS} to transform $\{DPS\}$ to an {FFS}
{FFS} = $(\{DPS\}/FL_{DPS})100$

Plot and verify transformation to {FFS}
Adjust variables in models as needed

Figure 72 – Flow–diagram for transforming CBD signature data into FFS data

ARULE™ is packaged with sets of input data as well as SDEF and NDEF files that are used as examples for illustrative, instructional, and guidance purposes. The data used in those examples were originally collected from experiments, simulations, published works, internal and funded projects, and so on. Sometimes data from multiple collections and sources were combined to form more complete data sets; and sometimes dense data collections, such as those gathered from continuously running sensors, were transformed to less data collections. In prognostic applications, data should be collected periodically with high sampling frequencies. For example, the data plotted in Figure 98, page 87, comprises 109 data points extracted from an original data set of over 300,000,000 data points (500+ hours of about 160 data points per second).

Table 6. Table of FFP, DPS, and FFS Transformation Models

Increasing Signature Function	FFP Signature Model	FFP-to-DPS Transform Model	FFP-based FL to DPS-based FL Model
Set 1 Power	$(dP_i/P_0)^n$	$(FFP_i)^{1/n}$	$(FL_{FFP})^{1/n}$

6.1 DEMO1

The primary purpose of DEMO1 is to illustrate “batch–mode” processing of a single input data file to produce prognostic estimates of RUL, PH, and SoH. The input data file is in .txt format and represents the resonant frequency response from a PC power supply shown in Figure 73.

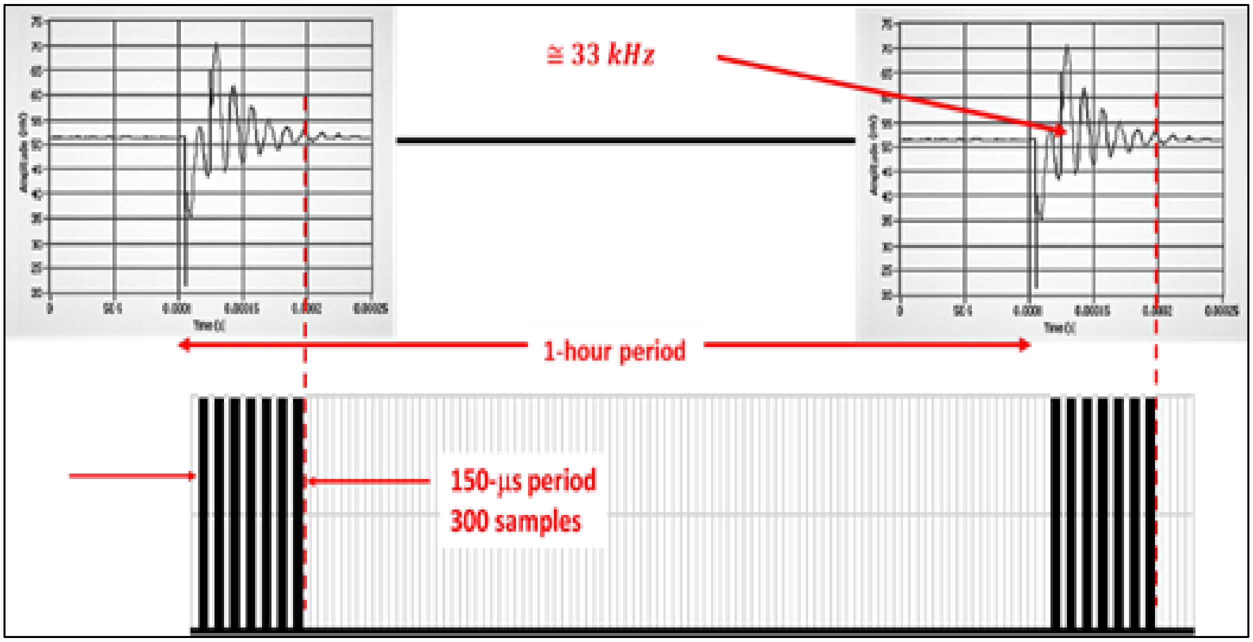


Figure 73 – Example of high–rate burst–mode sampling at a low–rate periodic–mode sampling

The data set was collected using Ridgetop's SMRT Probe 4000 sensing device which is a key component in the Sentinel Power family of prognostic solutions for power supply systems.

6.1.1 DEMO1: Definitions

As shown in Figure 74, the SDEF file for this example points to a node definition number of 1, and an NDEF file named **DEMO1_NODE1**. Figure 75 shows the NDEF parameters and keywords used for DEMO1.

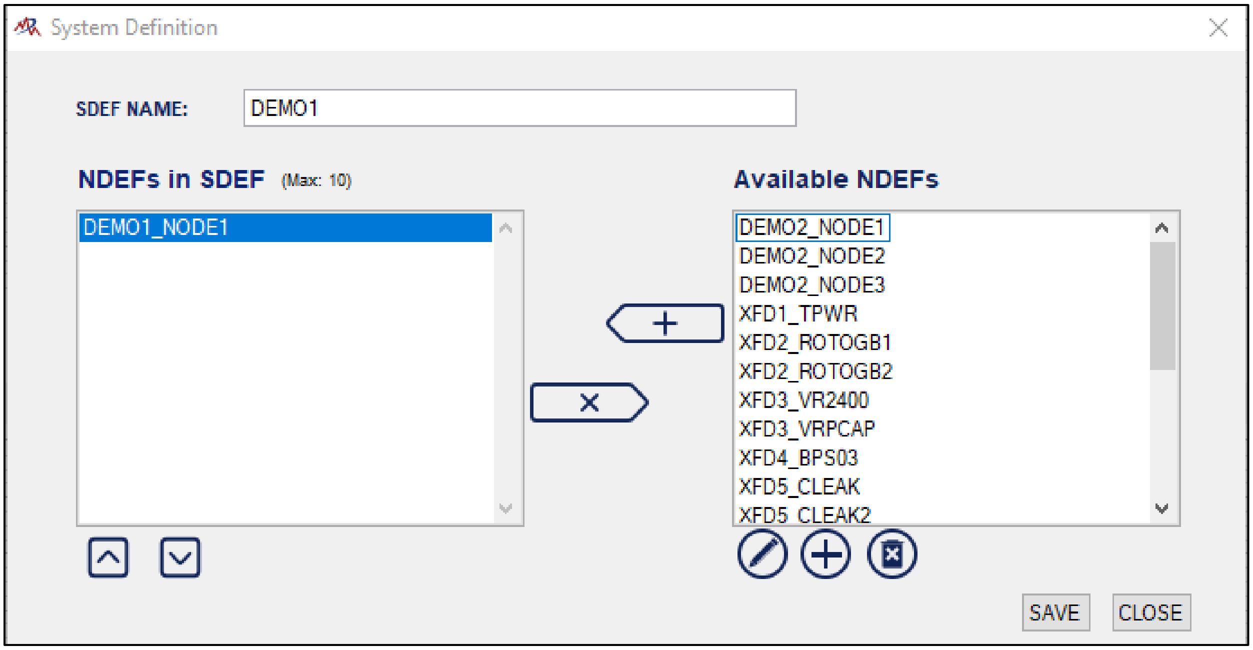


Figure 74 – DEMO1 System Definition (SDEF) User Form

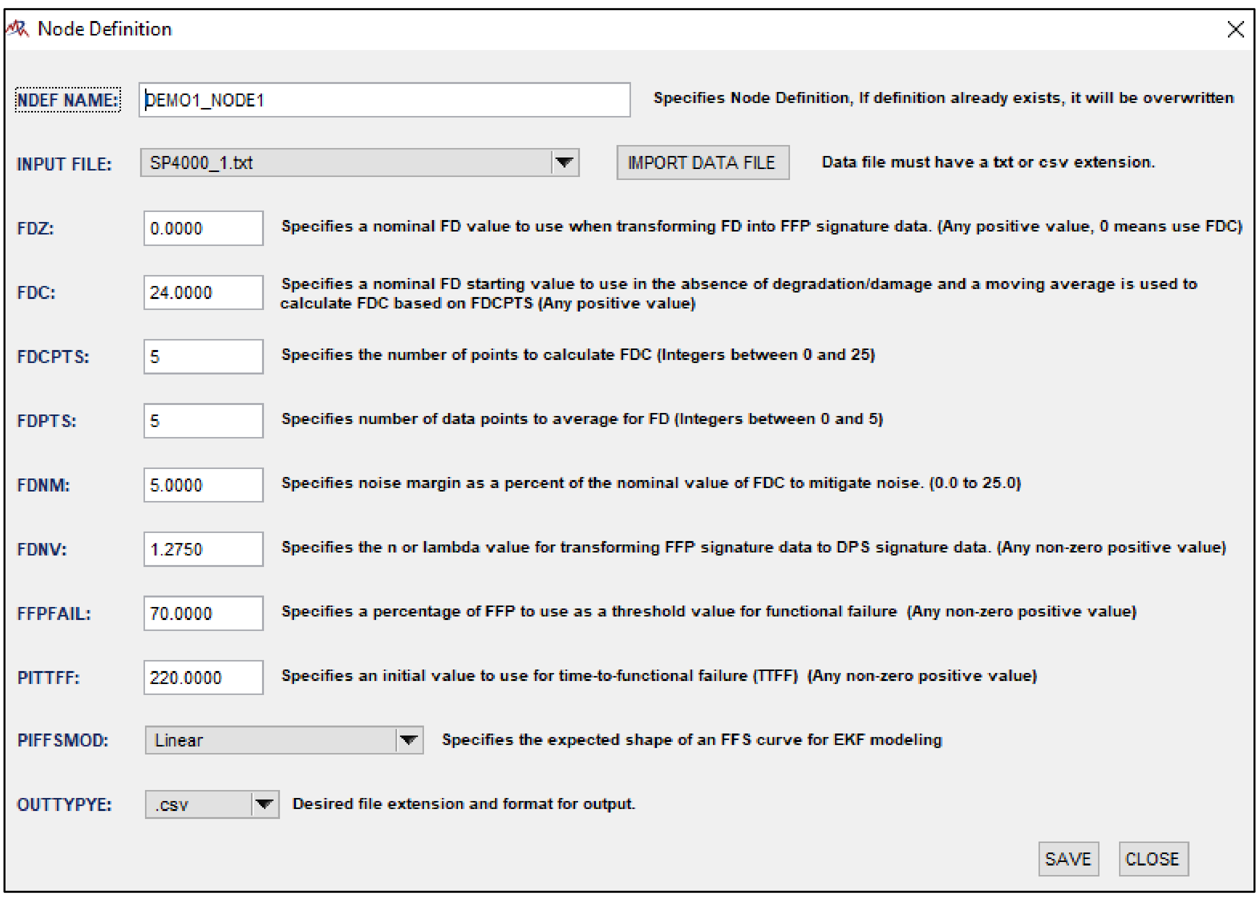


Figure 75 – DEMO1 Node Definition User Form

6.1.2 DEMO1: Input Feature Data

Figure 76 shows the input data plotted in the Input Data Plot Preview of the GUI.

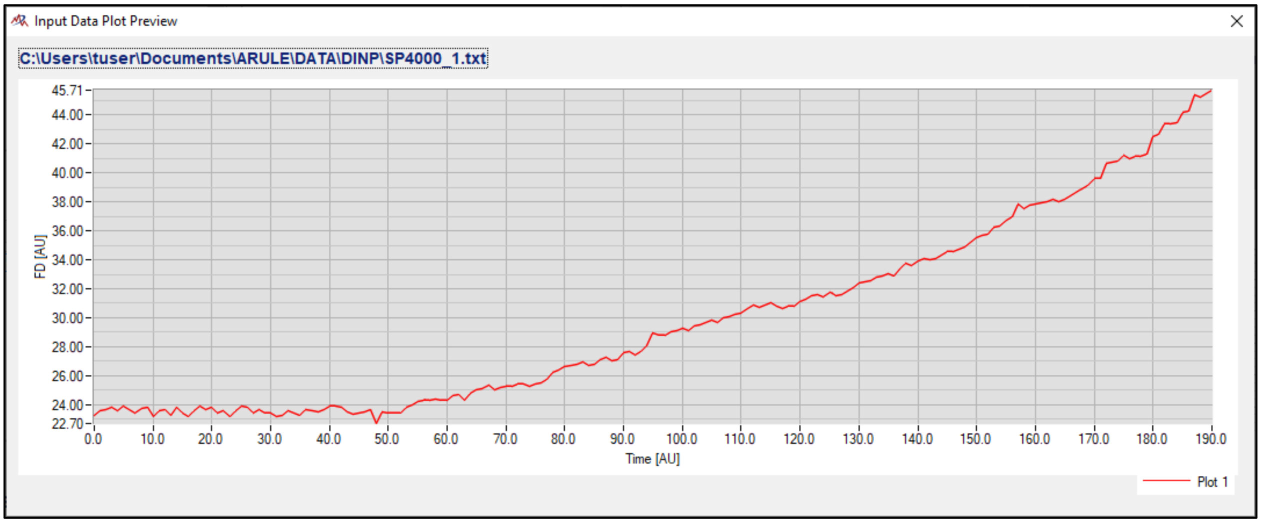


Figure 76 – DEMO1 Input Feature Data and ARULE RUN ARGUMENTS section

6.1.3 DEMO1: Data Outputs and Prognostic Information

Figure 77 shows the plots of the prognostic information: FD, FFS Progression, RUL, PH, and SoH.

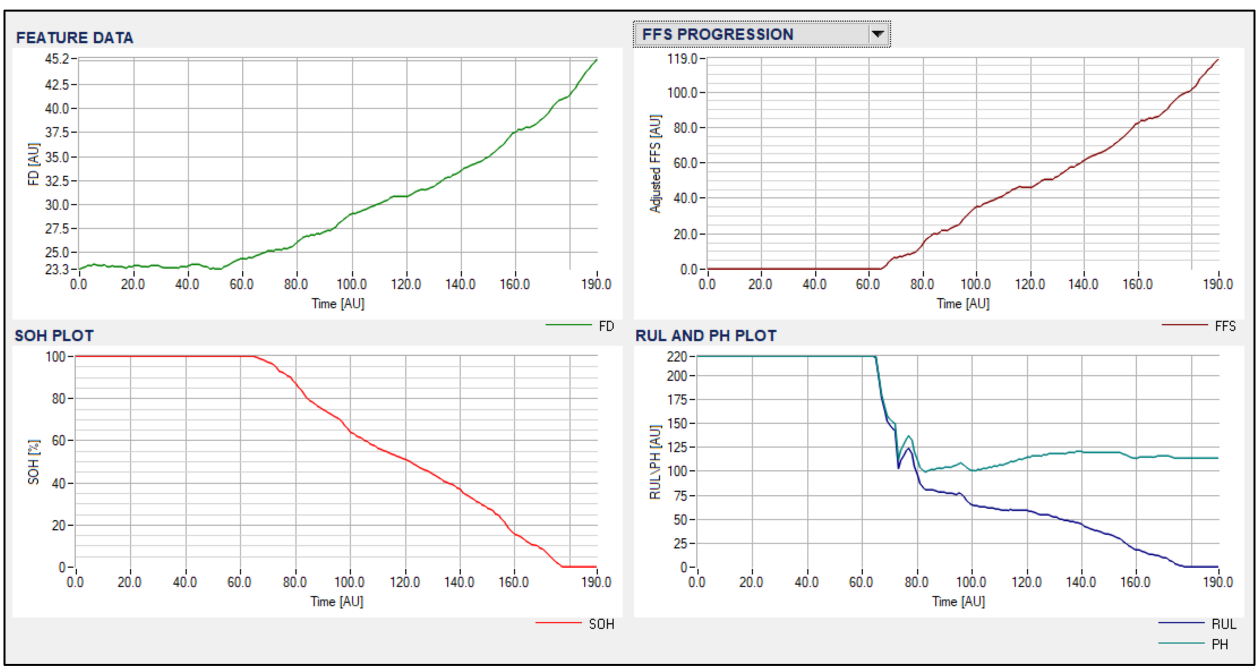


Figure 77 – Data Output section for DEMO1

6.1.4 DEMO1: Discussion

Power supplies supply power, voltage, and current, to a load and, as the load draws current, the magnitude of the output voltage changes. The magnitude of the change is dependent on multiple factors such as the magnitude of the current, the rate at which the current is drawn, the amount of capacitance and inductance used in the output filter circuit of the power supply. Capacitors have intrinsic series resistance that results in voltage changes called ripple voltage and a common design technique is to use many capacitors in parallel to reduce the total value of that intrinsic resistance. A common mode of failure is observed with the dielectric material of those capacitors: a capacitor fails short, high current results, and the capacitor connectors overheat and fail open – the net result is a loss of total capacitance, which causes the resonant frequency of the filter to increase.

A method to detect such failure is to attach a sensor, such as Ridgetop's SMRT Probe 4000, to the output of a power supply. When the sensor is commanded to sample, it introduces a small abrupt change in load that results in a damped–ringing response at the resonant frequency of the output filter of the power supply. The captured change in resonant frequency is shown in Figure 76. The plot is noisy because of thermal noise, switching noise, and distortion noise introduced by the feedback circuity of the power supply. Despite the noise, the results of processing the very irregular, noisy data are evaluated as excellent: refer back to Figure 77.

6.2 DEMO2

The primary purpose of DEMO2 is to illustrate "batch–mode" processing of three separate input data files of different types to produce prognostic estimates of RUL, PH, and SoH. The first and second input data sets are in .txt format with varying FDNV and FFPFAIL parameter values, and the third input data set is in .csv format. The data values are not identical, and are representative to the resonant frequency data analyzed in DEMO1.

6.2.1 DEMO2: Definitions

As shown in Figure 78, the SDEF file for this example points to two node definitions: **DEMO2_ NODE1**, **DEMO2_NODE2**, and **DEMO2_NODE3**. Figure 79 shows the NDEF parameters, keywords, and the input data used for DEMO2: the three node definitions and the three input data from SP4000_1.txt and SP4000_2.csv. These data are not the same, as evidenced by the plots in Figure 82 and in Figure 83.

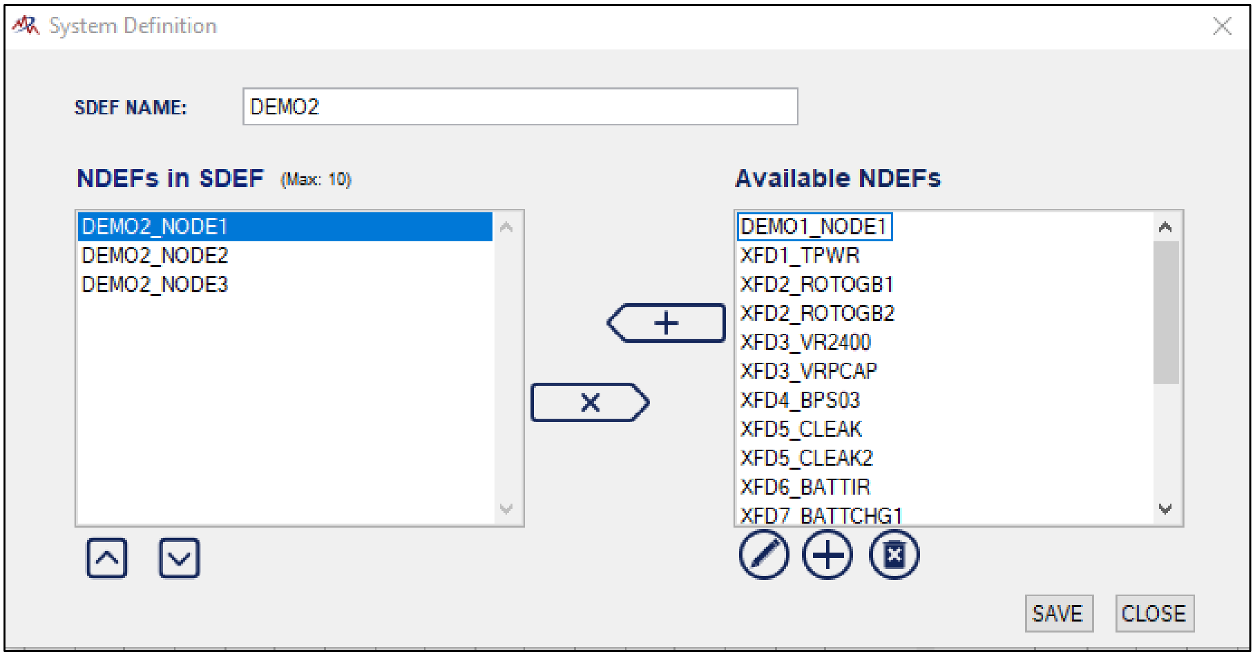


Figure 78 – DEMO2 System Definition (SDEF) User Form

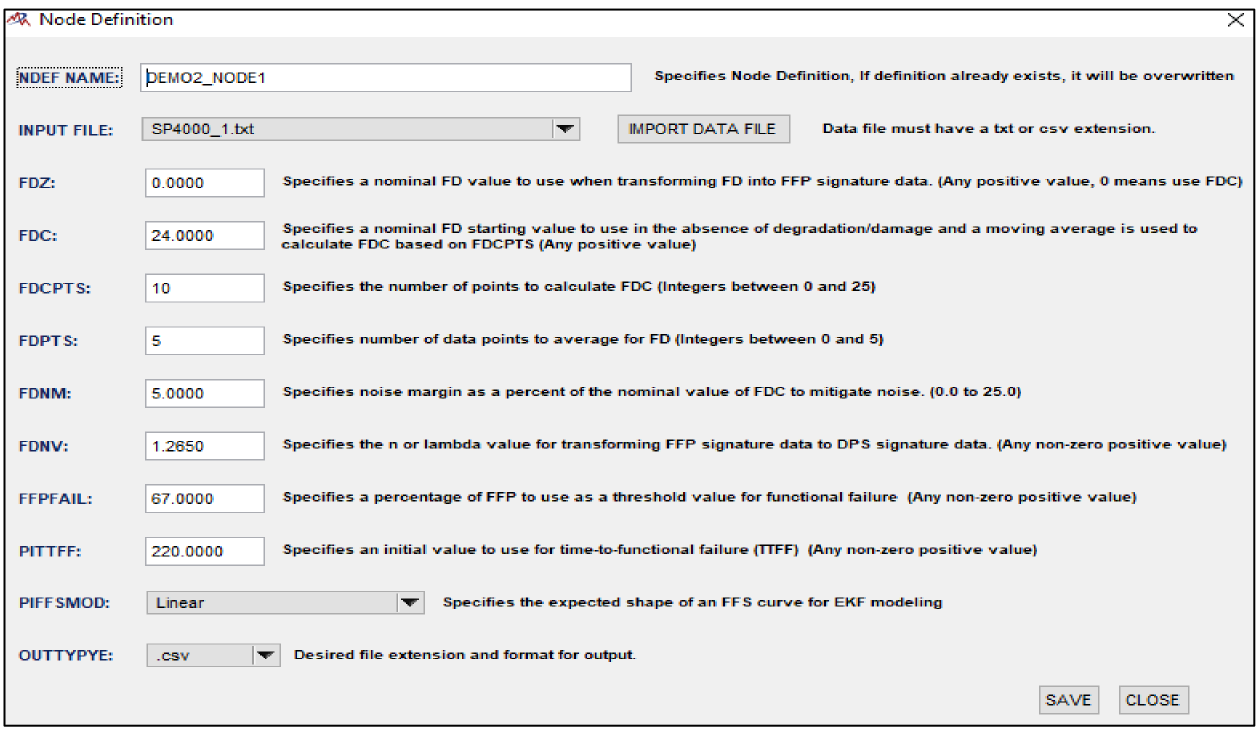


Figure 79 – Node Definition (NDEF) User Form for the First Node in DEMO2

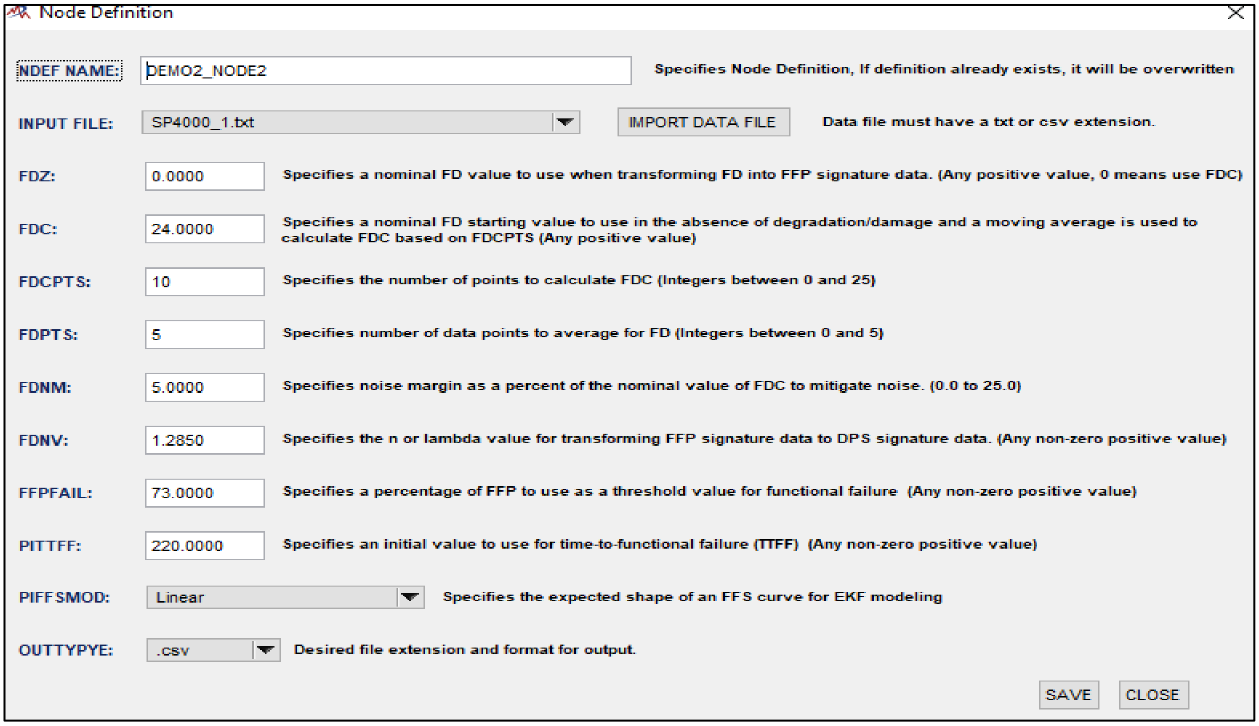


Figure 80 – Node Definition (NDEF) User Form for the Second Node in DEMO2

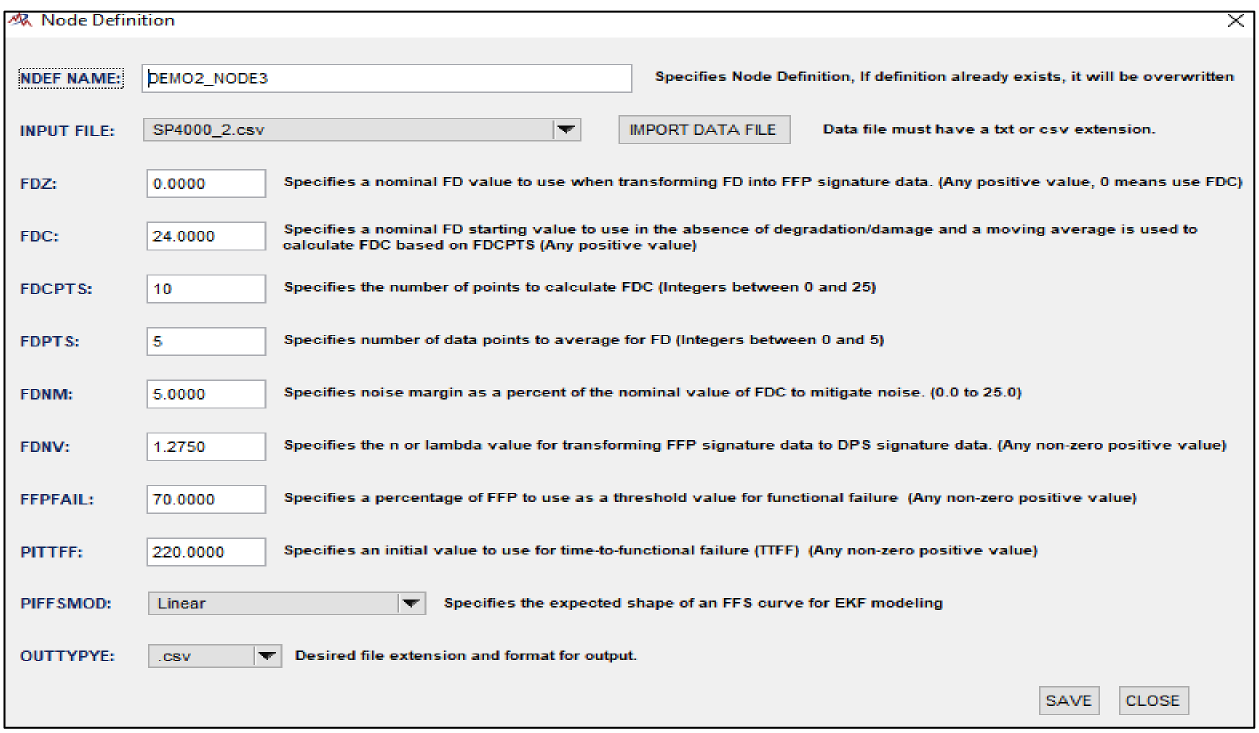


Figure 81 – Node Definition (NDEF) User Form for Third Node in DEMO2

6.2.2 DEMO2: Input Feature Data

Figure 82 shows the input data set in .txt format for the First and Second Node, and Figure 83 shows the input data set in .csv format for the Third Node. Note that data sets are not identical in values and FDNV & FFPFAIL parameters are varied in the case of the First and Second node.

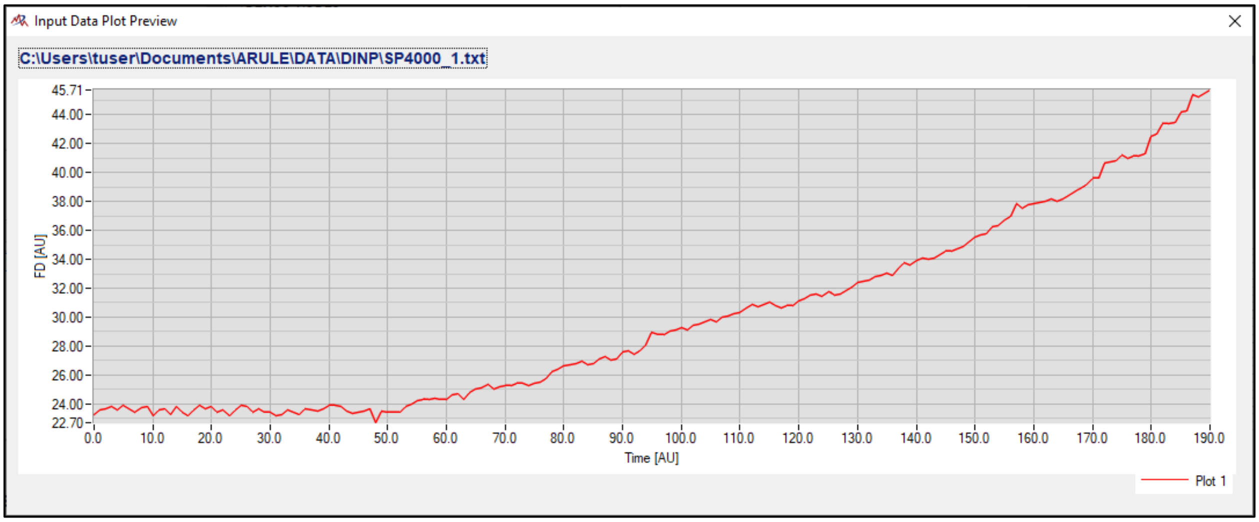


Figure 82 – DEMO2 Input Feature Data for the First and Second Node

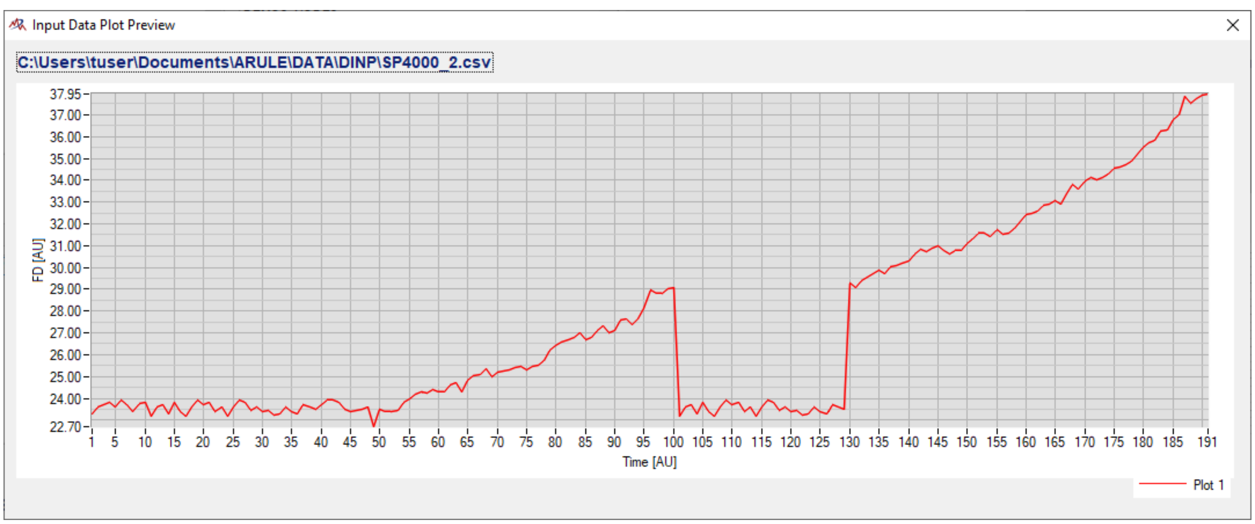


Figure 83 – DEMO2 Input Feature Data and for Third Node

6.2.3 DEMO2: Data Outputs and Prognostic Information

Figure 84, Figure 85, and Figure 86 show the data outputs and prognostic information for the First, Second, and Third Node respectively. These data outputs include the FD, FFS Progression, along with SoH, RUL, and PH. As expected, these plots are almost identical in the case of the First and Second node, but the Third node has different results.

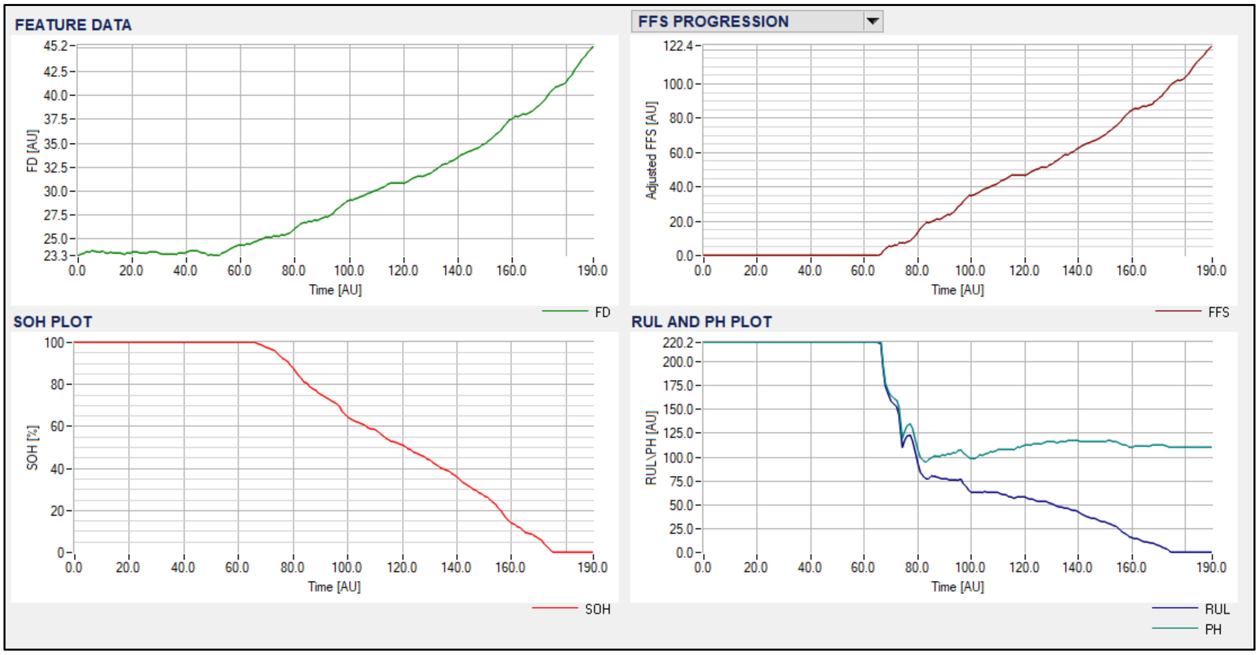


Figure 84 – Data Output for the First Node in DEMO2

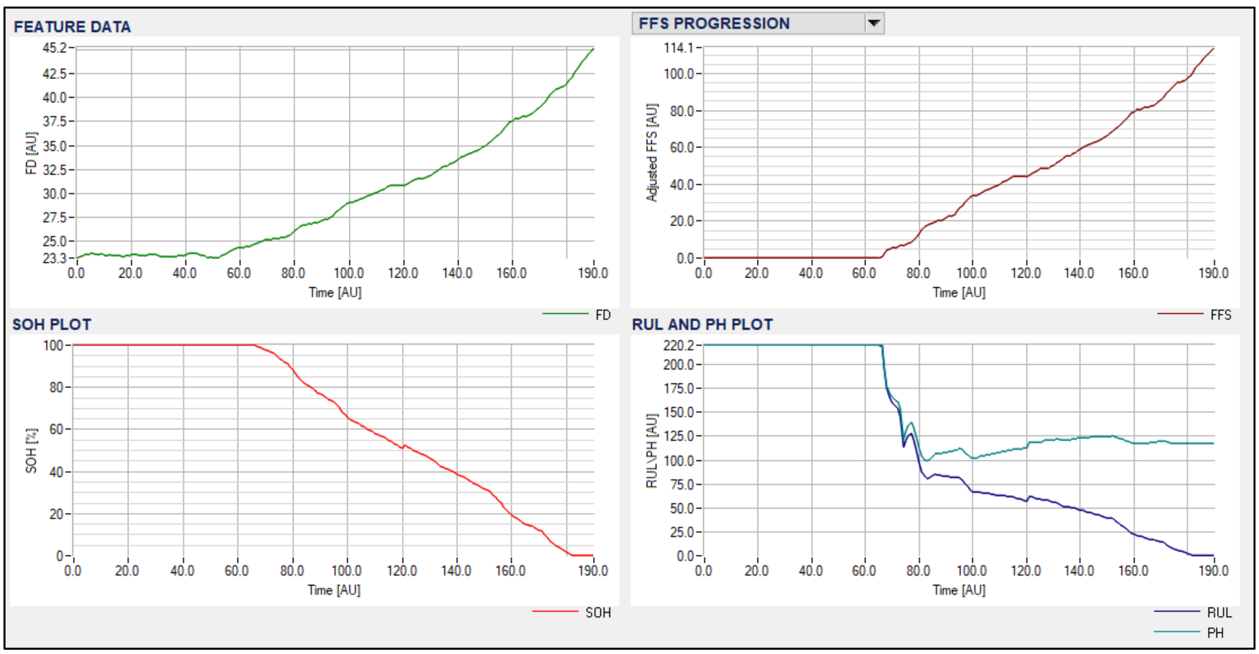


Figure 85 – Data Output for Second Node in DEMO2

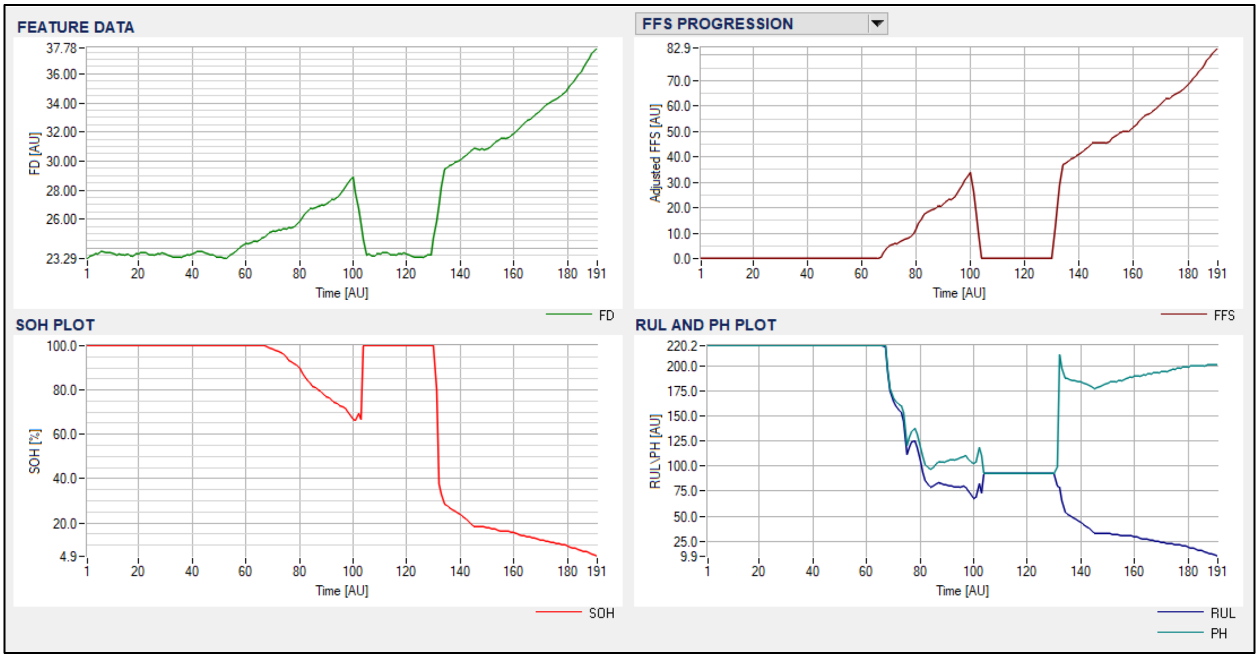


Figure 86 – Data Output for Third Node in DEMO2

6.2.4 DEMO2: Discussion

The first two data sets processed in this demonstration are equivalent to the DEMO1 data set, and therefore the same discussion notes can also be applied to this example. As previously noted, the primary purpose of this example was to demonstrate the capability of ARULE™ to process multiple NDEF files that point to two different types of input data sets. The first two input data sets were in .txt format, and the third input data set was in .csv format; and all output data sets are in .csv format.

A first glance of the plots of the output of all nodes might lead to a conclusion that the prognostic estimates are identical: they are not – because the two input files, SP4000–1.csv and SP4000–2.txt are not identical. Examples of non–identicality are shown in Figure 87.

SP4000-1.csv			SP4000-2.txt		
Position	Time	Amplitude	Position	Time	Amplitude
1	1	23.3	1	0	23.3
26	26	23.913	26	25	23.9125

Figure 87 – Examples of differences between the input data in SP4000–1.csv and SP4000–2.csv

6.3 XFD1: Transmission Power

The first real world example is demonstrated with the SDEF titled XFD1. This example is related to the transmission of radio frequency (RF) being prone to loss of transmitted power. One method of monitoring this problem is to place a receiver at a known distance from the transmitter, receive transmitted signals, measure the received power, and use that measurement to detect loss.

6.3.1 XFD1: Definitions

As shown in Figure 88, the SDEF file for this example points to two node definitions. The NDEF files are named **XFD1_TPWR** and **XFD1_TPWRB**. Figure 89 and Figure 90 show the NDEF parameters and keywords used for the two nodes corresponding to XFD1. There is only one difference between the two NDEF files: **XFD1_TPWR** has a PITFFF keyword parameter value of 200 as compared to a value of 60 for **XFD1_TPWRB**. By applying this small change to the initial estimate of RUL and PH, the convergence and accuracy of the ARULE™ outputs are drastically improved as detailed in the discussion section for this example.

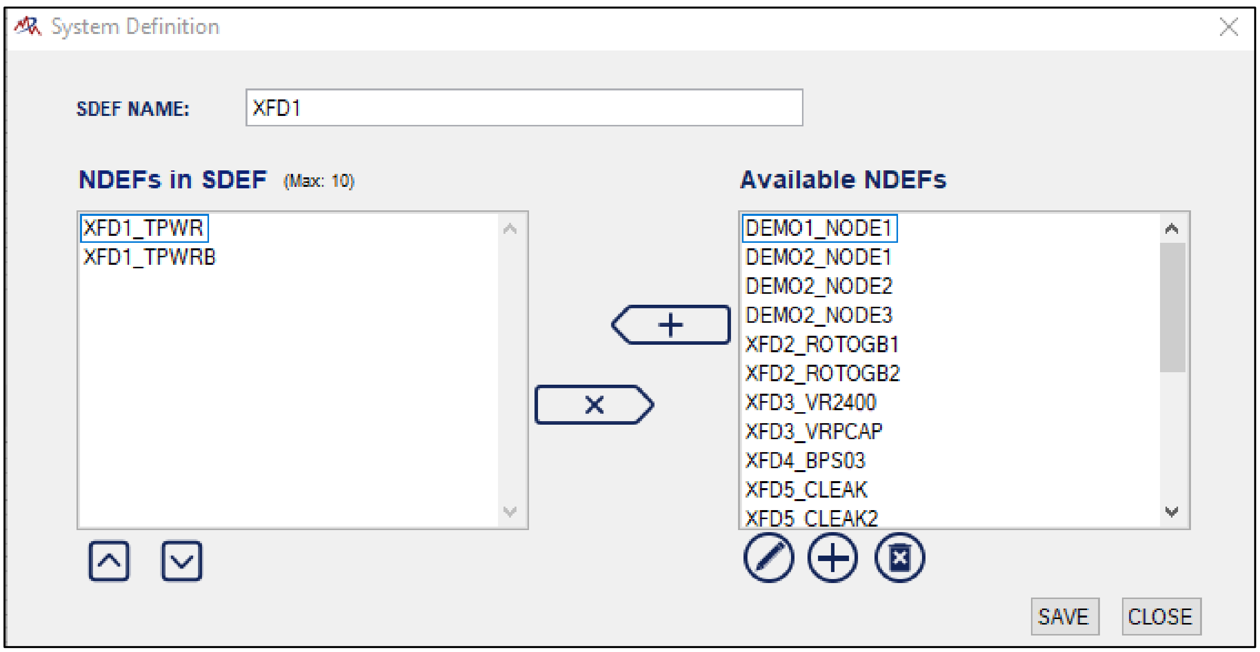


Figure 88 – XFD1 System Definition (SDEF) User Form

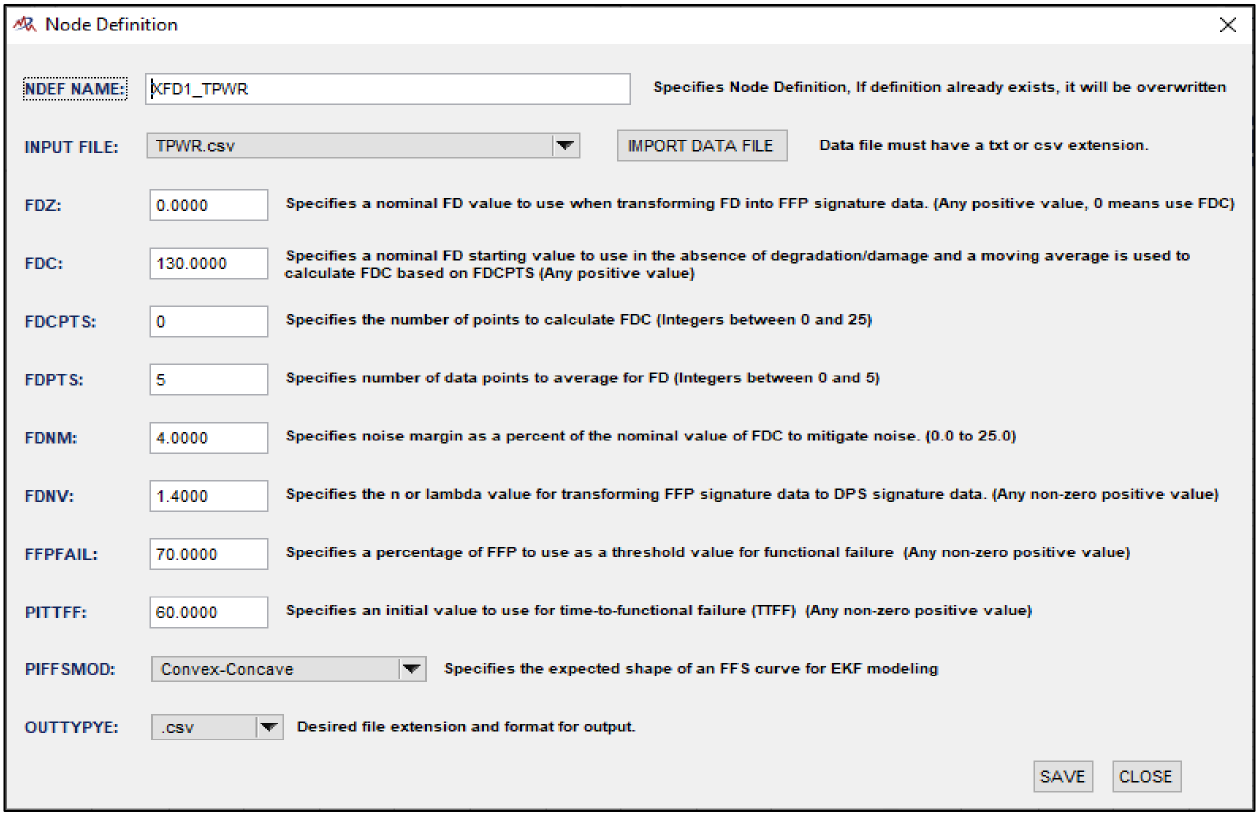


Figure 89 – Node Definition (NDEF) User Form for the First Node in XFD1

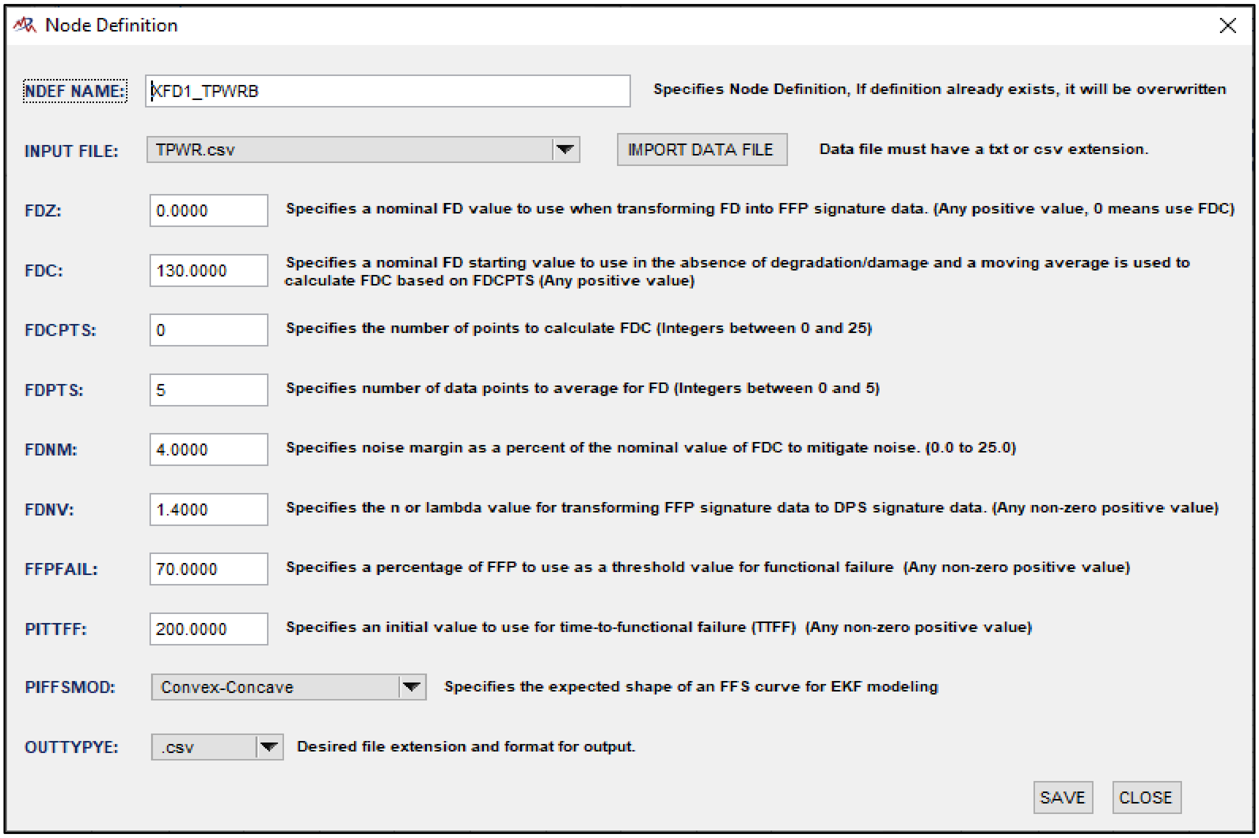


Figure 90 – Node Definition (NDEF) User Form for the Second Node in XFD1

6.3.2 XFD1: Input Feature Data

Figure 91 shows the Condition–Based Data input plot for both nodes used in this example. Note that the Condition–Based data input sets are identical for both nodes, as this example is demonstrating how changing the initial estimate for RUL and PH (PITTFF) impacts accuracy and convergence.

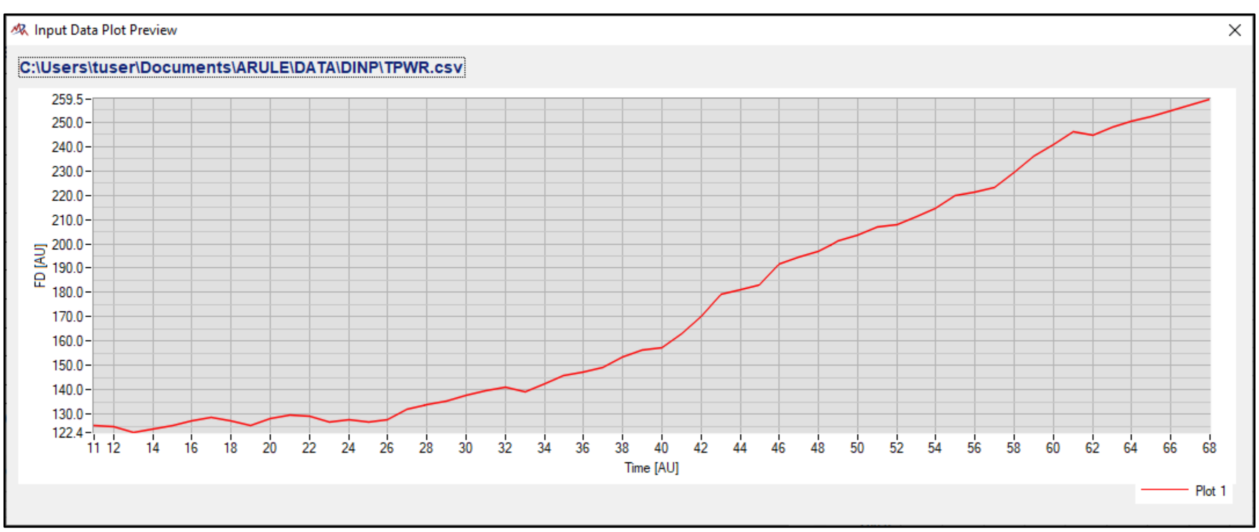


Figure 91 – XFD1 Input Feature Data for the First and Second Node.

6.3.3 XFD1: Data Outputs and Prognostic Information

Figure 92 and Figure 93 show the Data Output for the two nodes in XFD1: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

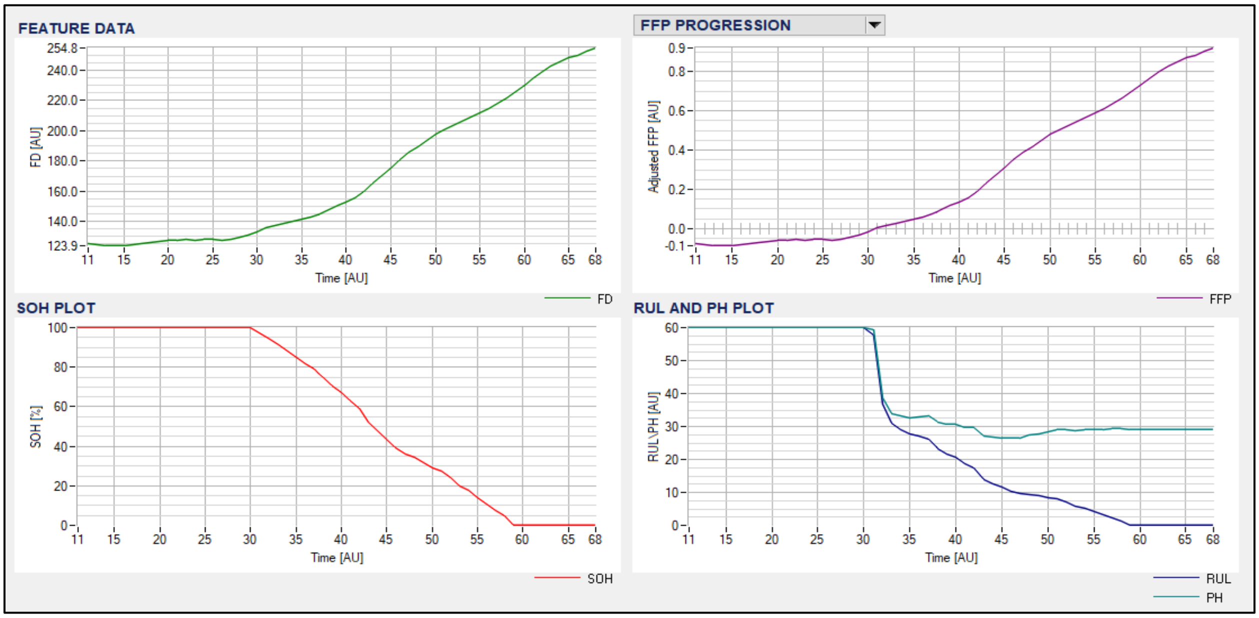


Figure 92 – Plots of the XFD1 Prognostic Information for the First Node

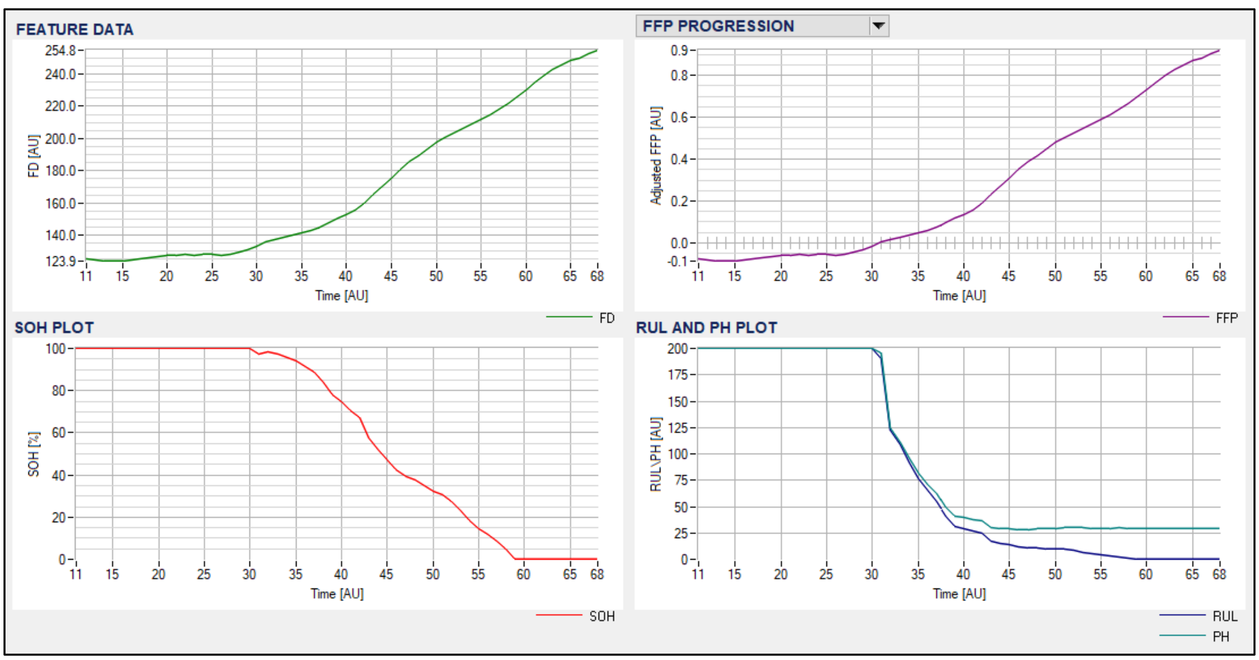


Figure 93 – XFD1 plots of the Prognostic Information for the Second Node

6.3.4 XFD1: Discussion

Transmission of radio frequency (RF) is prone to loss of transmitted power: one method of monitoring is to place a receiver at a known distance from the transmitter, receive transmitted signals, measure the received power, and use that measurement to detect loss. The results of processing the very irregular, noisy data are evaluated as excellent.

Figure 92 and Figure 93 illustrate the effect of the initial estimate of how long it takes for functional failure to occur after the onset of degradation: ARULE™ converged from an initial–estimate error of 800% to less than 10% in eight (8) samples with PITTFF = 60 (Figure 92); ARULE™ converged from an initial–estimate error of 100% to less than 10% in only two (2) samples with PITTFF = 200 (Figure 93).

6.4 XFD2: Helicopter Gear Box – Pinion Gear

Ridgetop Group has developed an innovative sensing solution under the Sentinel Motion product line for detecting degradation in rotating–shaft equipment such as helicopter gear box systems, transmissions, and other mission critical equipment that rely on shock, vibration, and temperature monitoring. Sentinel Motion is based on the Internet–of–Things (IoT) where the system comprises a wireless network of RotoSense™ smart sensors, the Sentinel Gateway communications device, and the Sentinel MotionView software package for data acquisition, analysis, and sensor–gateway management. An example for how this sensing system was used to collect and process Condition–based Data (CBD) while working with NASA is shown in Figure 94 and this White Paper. Additional details pertaining to the full experiment can be reviewed in this NASA Spinoff Article, or on the Ridgetop Group website. This User Guide example illustrates how Sentinel Motion, RotoSense™, and the prognostic analysis from ARULE™ can be used to monitor and diagnose cracked gear teeth.

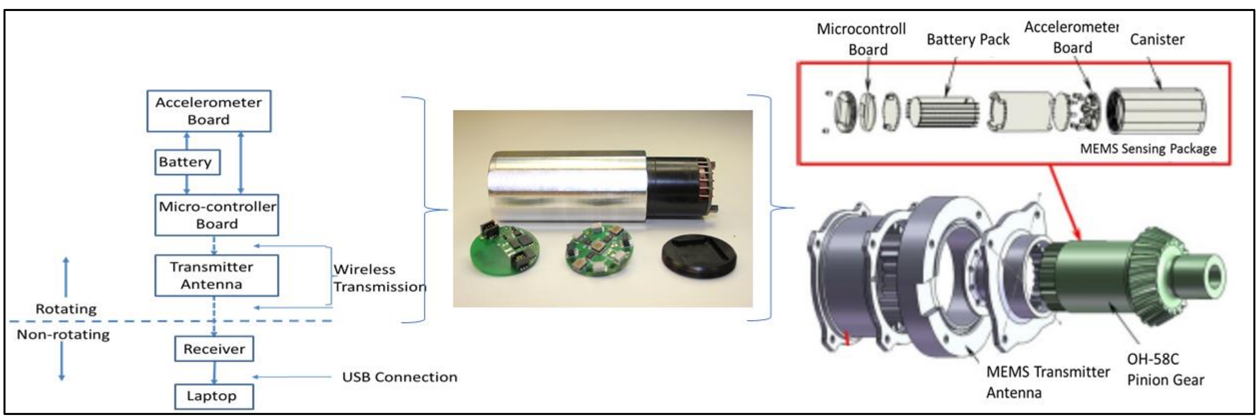


Figure 94 – RotoSense diagram, unit, and assembly in gearbox system.

6.4.1 XFD2: Definitions

Figure 95 shows there are two nodes in XFD2: **XFD2_ROTOGB1** and **XFD2_ROTOGB2**. The NDEF files are identical: the difference is the second input data set was simulated to remove the instances of where the transmission was stopped to further damage the cracked gear tooth for this experiment. Figure 96 and Figure 97 show the NDEF parameters and keywords used for the two nodes corresponding to XFD1.

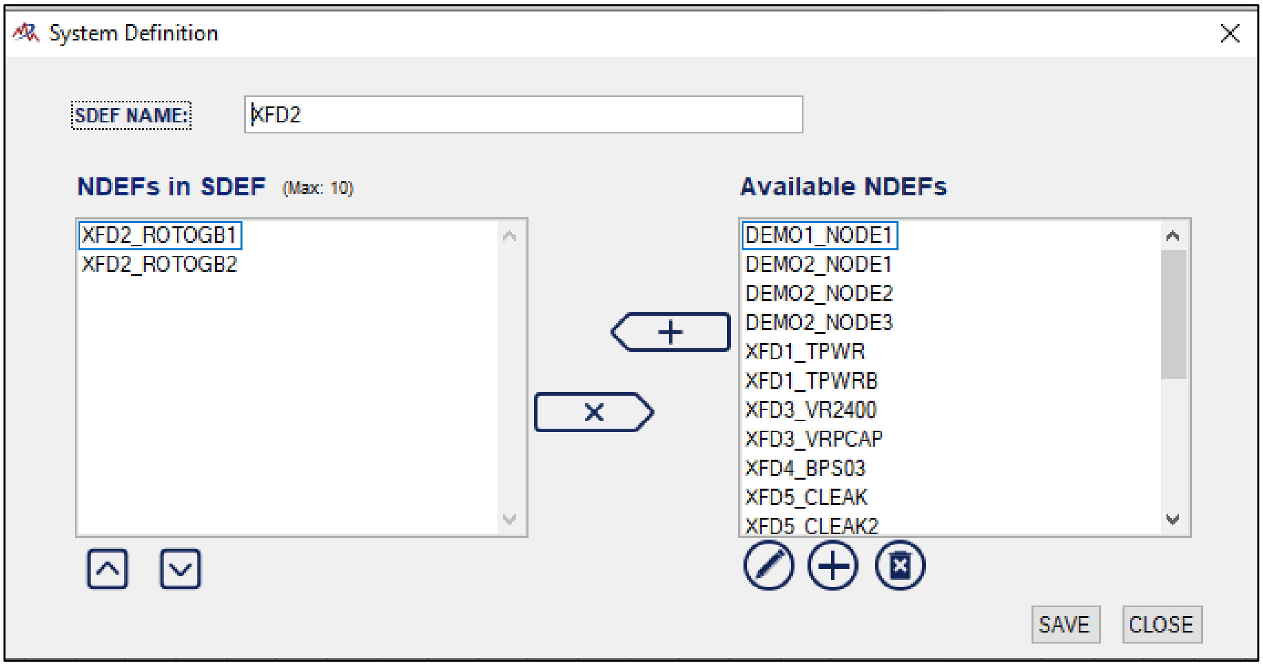


Figure 95 – XFD2 System Definition (SDEF) User Form

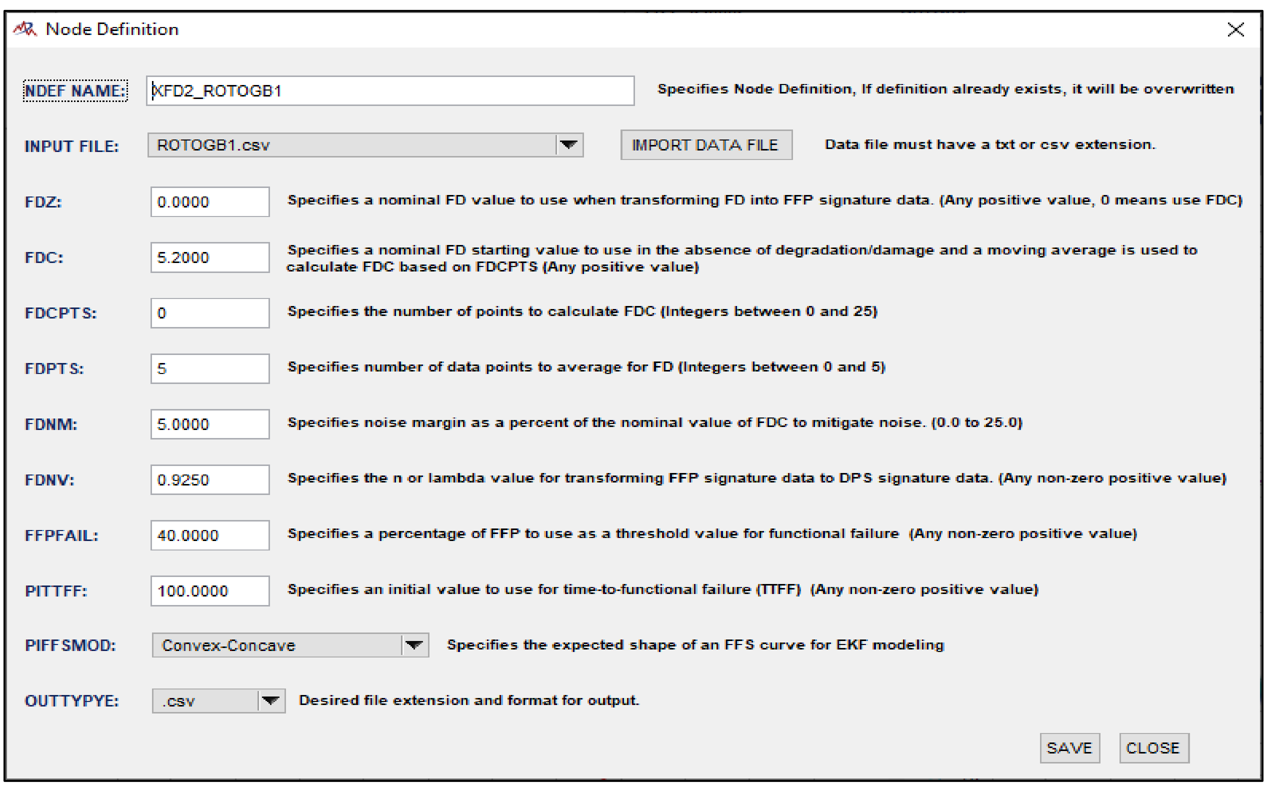


Figure 96 – Node Definition (NDEF) User Form for the First Node in XFD2

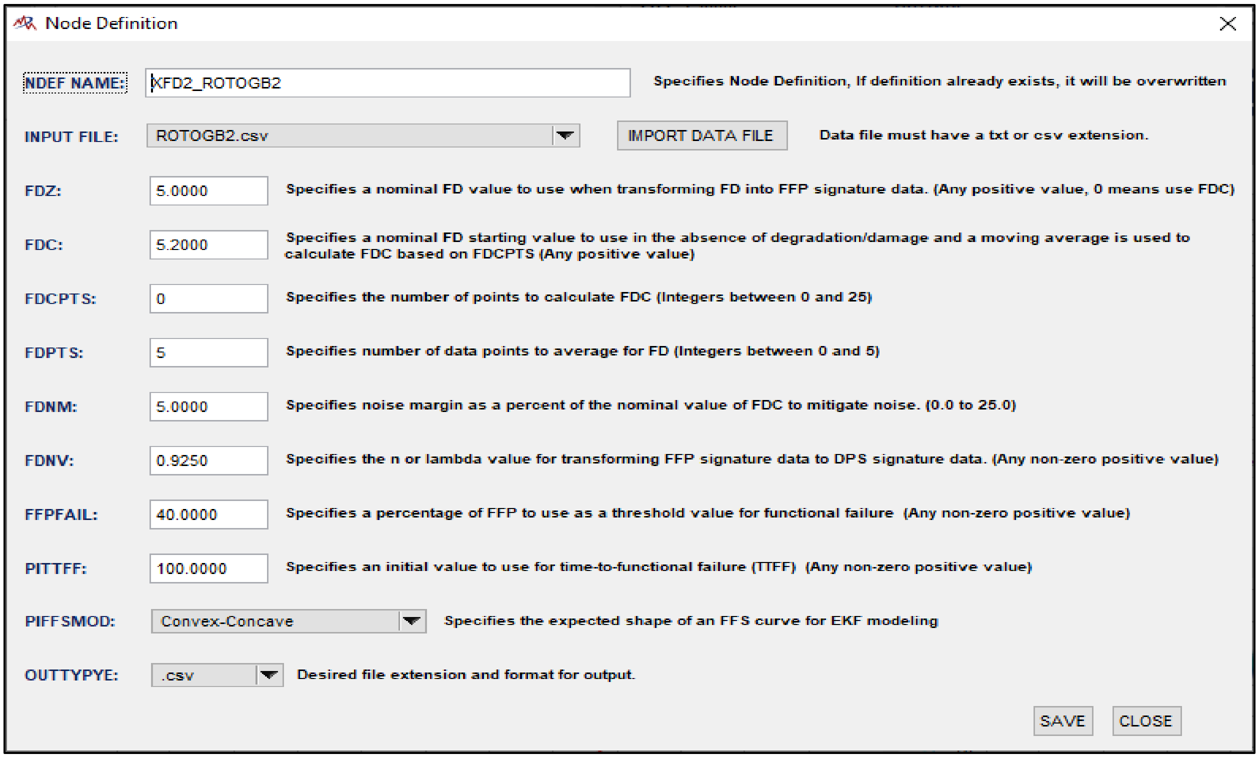


Figure 97 – Node Definition (NDEF) User Form for the Second Node in XFD2

6.4.2 XFD2: Input Feature Data

Figure 98 and Figure 99 show the Condition–Based Data input plots for **XFD2_ROTOGB1** and **XFD2_ROTOTGB2** respectively. The example data sets for both nodes represent the Time Synchronous Averaging (TSA) of the accelerometer data signals that were wirelessly collected with RotoSense™. Note that the example data set for XFD_ROTOGB2 starts at hour 135, as that is when the simulation data starts to progress from healthy, towards a state of degradation, and eventually towards functional failure.

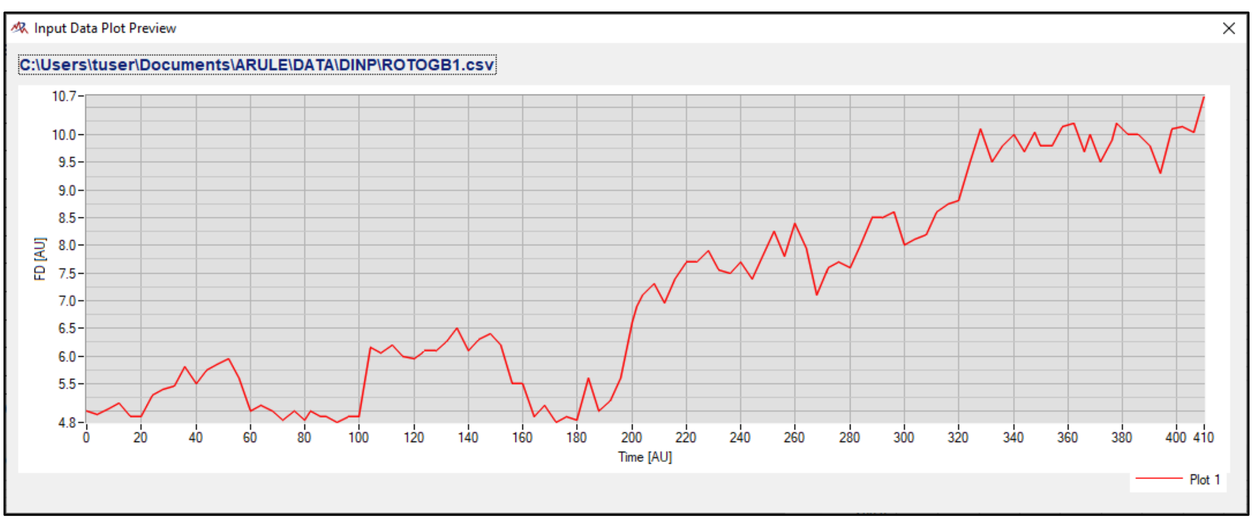


Figure 98 – XFD2 Input Data Plot Preview for Node 1

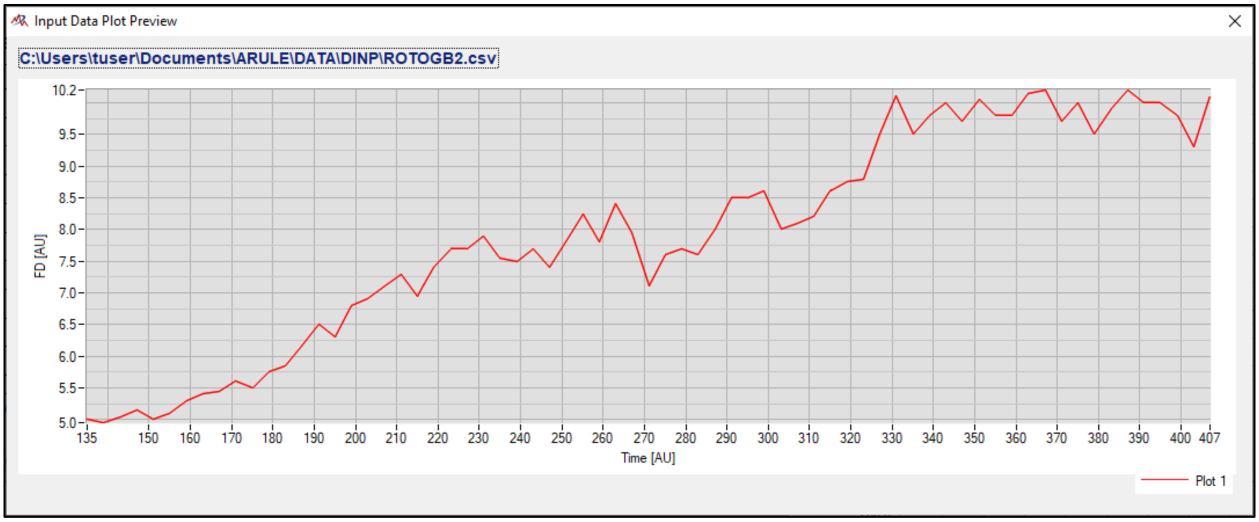


Figure 99 – XFD2 Input Data Plot Preview for Node 2

6.4.3 XFD2: Data Outputs and Prognostic Information

Figure 100 and Figure 101 show the Data Output for the two nodes in XFD2: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

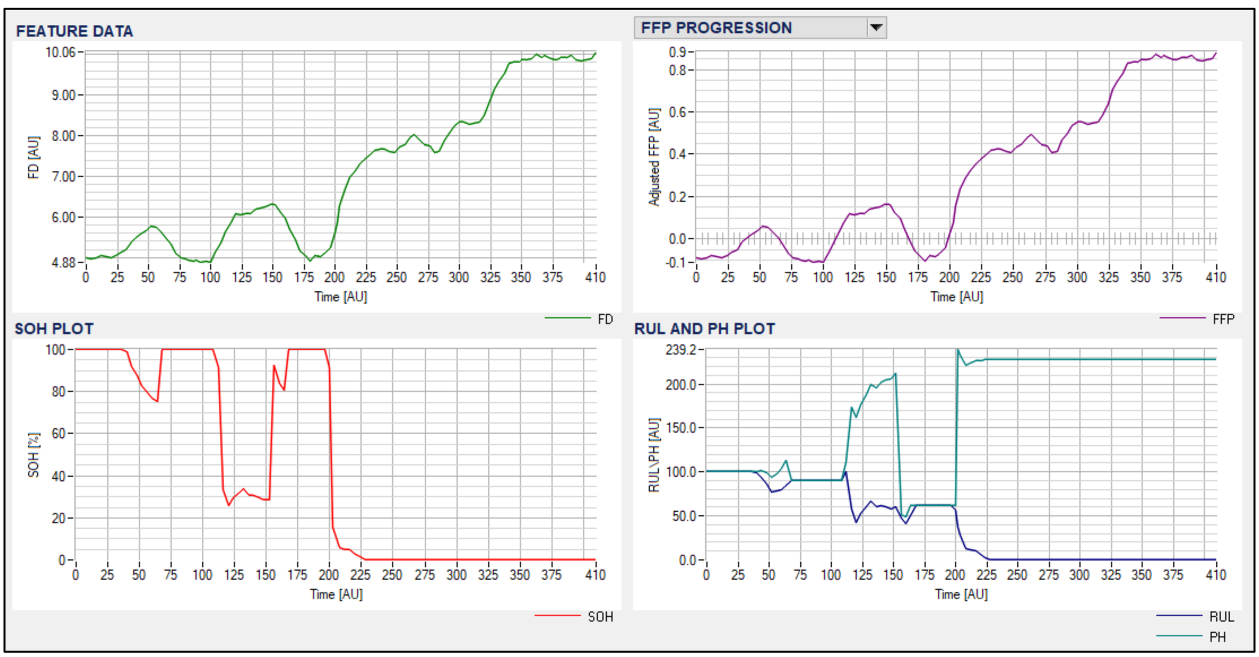


Figure 100 – Plots of the XFD2 Prognostic Information for the First Node

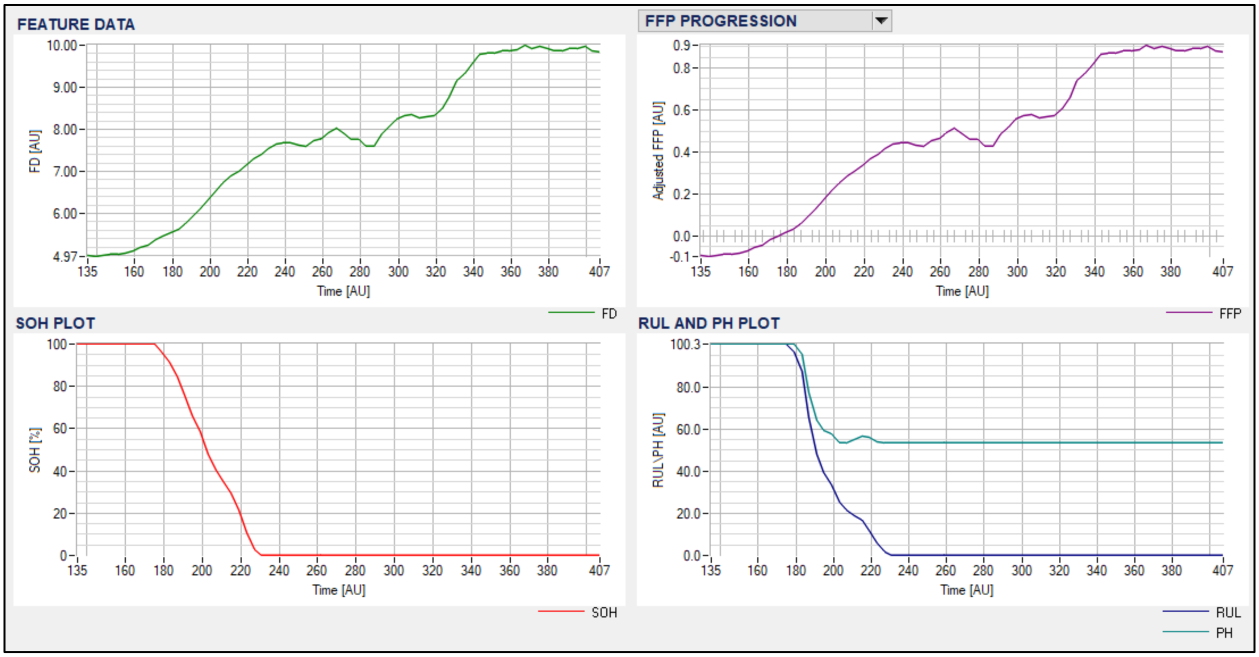


Figure 101 – Plots of the XFD1 Prognostic Information for the Second Node

6.4.4 XFD2: Discussion

Node 1 Discussion: Figure 98 shows input data in **XFD2_ROTOGB1**. There are two features in the dataset that are the result of starting an experiment with a pinion gear that was slightly damaged; then in the 5First and the 16First hour of the test, the experiment was stopped, the pinion gear was removed, the defected gear tooth was further damaged (by filing), and then re–installed. After re–installation, the gear experienced a temporary reduction in force before the again showing evidence of increased damage: as evidenced by the prognostic information shown in Figure 100.

Node 2 Discussion: Figure 99 shows input data in **XFD2_ROTOGB2** which is the same dataset shown in Figure 98 with the exception being that the input data set for Node 2 was derived from the first data set to simulate not stopping the transmission and removing and replacing the pinion gear. This was done to illustrate the significance of spauling. There is one evident feature in Figure 99 that is the result of the pinion gear spauling twice: fracture followed by separation which resulted in lessening of binding and roughness between gear teeth (see Figure 102). About 200 hours later (time ~ 520 hours), there was a sharp drop in measured force as seen in Figure 103.

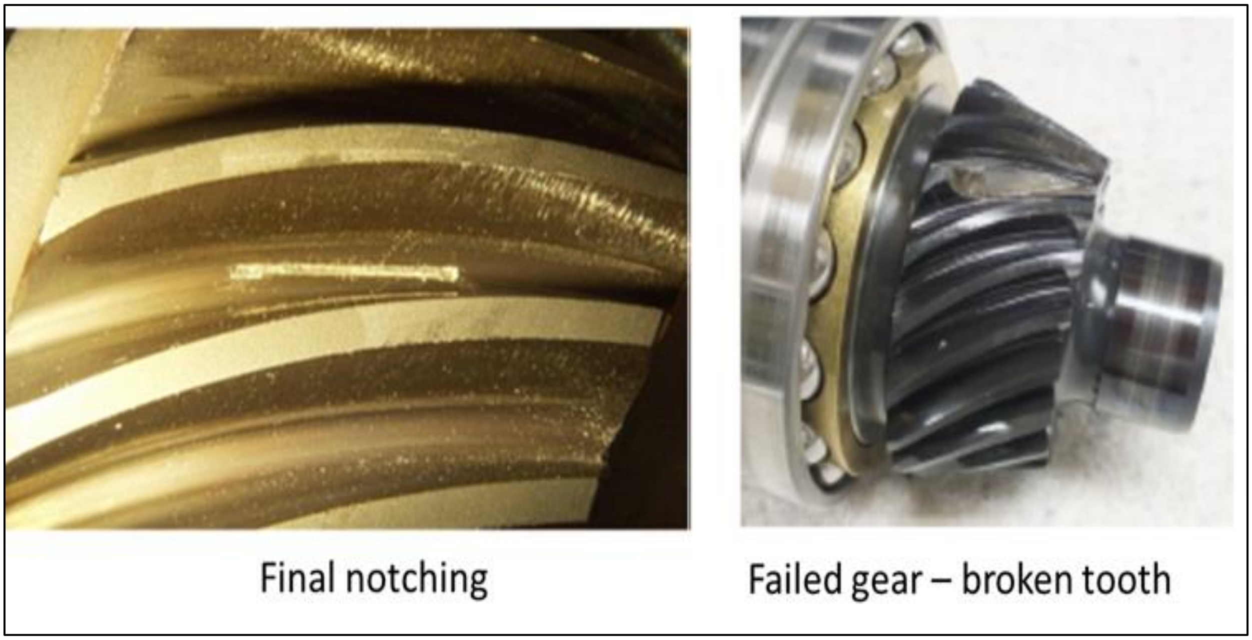


Figure 102 – Pinion gear after final notching @ 161 hours (left) and after the test was ended

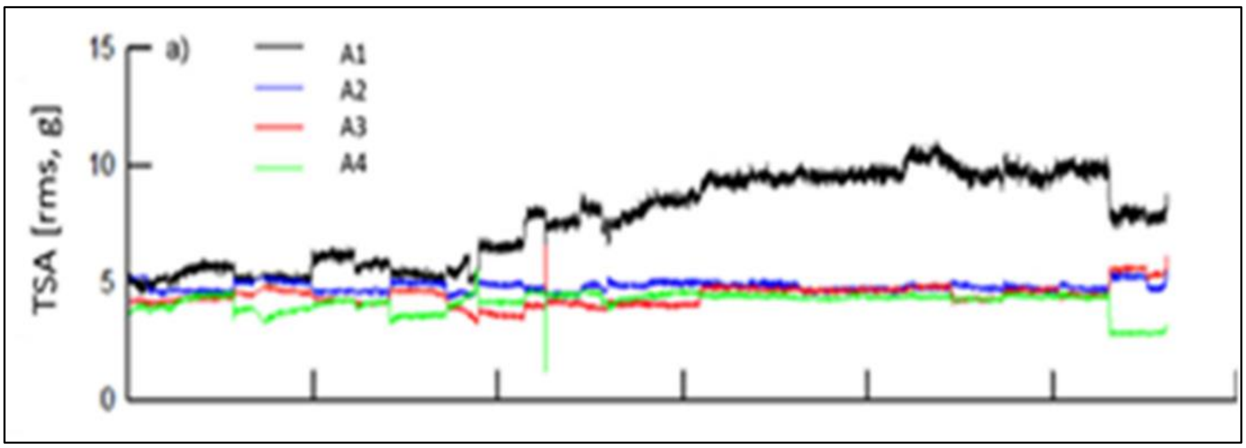


Figure 103 – Original data collection used to create input data for XFD2

6.5 XFD3: Ripple Voltage, PC Power Supply, Full–PI filter

The CBD from power supplies comprises all manner of FD that can be extracted and used to support diagnostic and/or prognostic applications: Figure 104 illustrates six examples of FD – ripple voltage, thermal noise, damped–ringing response, harmonic distortion, droop, and switching spikes and glitches. Figure 104 is a plot of ripple–voltage data collected in response to increasing loss of filtering capacitance of a power supply in a personal computer (PC). This ARULE™ example has been proven through Ridgetop's Sentinel Power product line, which is a family of prognostic solutions for power supply systems.

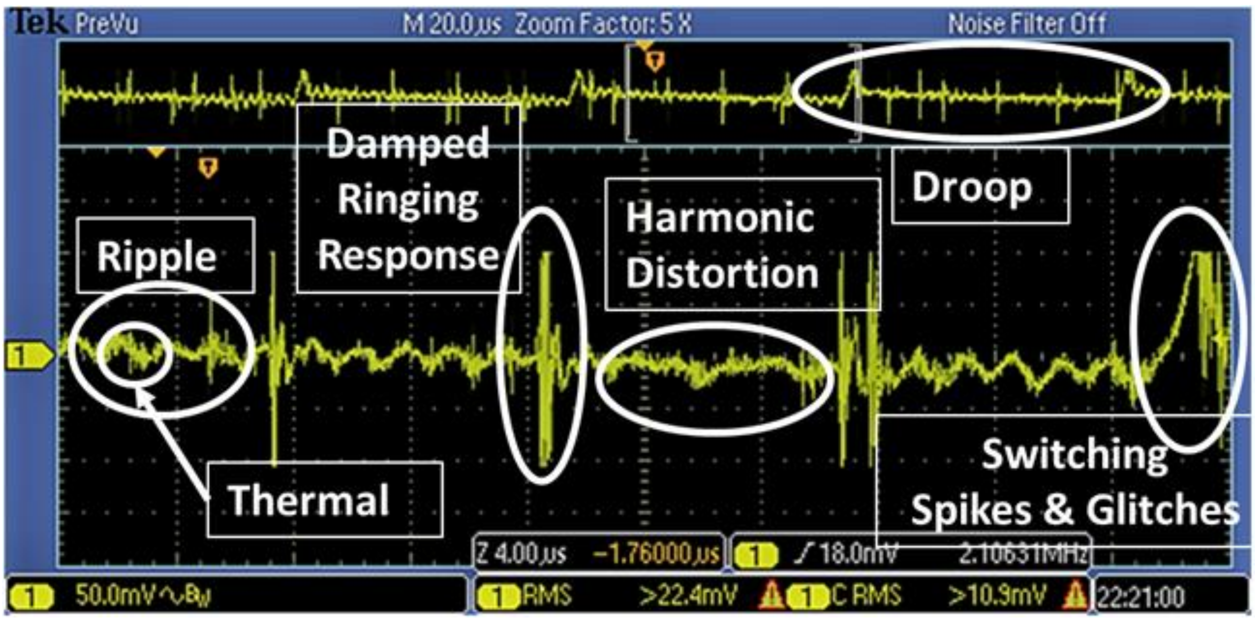


Figure 104 – Examples of Feature Data (FD) from power supply output

6.5.1 XFD3: Definitions

As shown in Figure 105, the SDEF file for this third example points to 2 node definitions, with NDEF files named **XFD3_VRPCAP** and **XFD3_VR2400**. Figure 106 and Figure 107 show the NDEF parameters and keywords used for XFD3 for the First and Second Node, respectively.

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

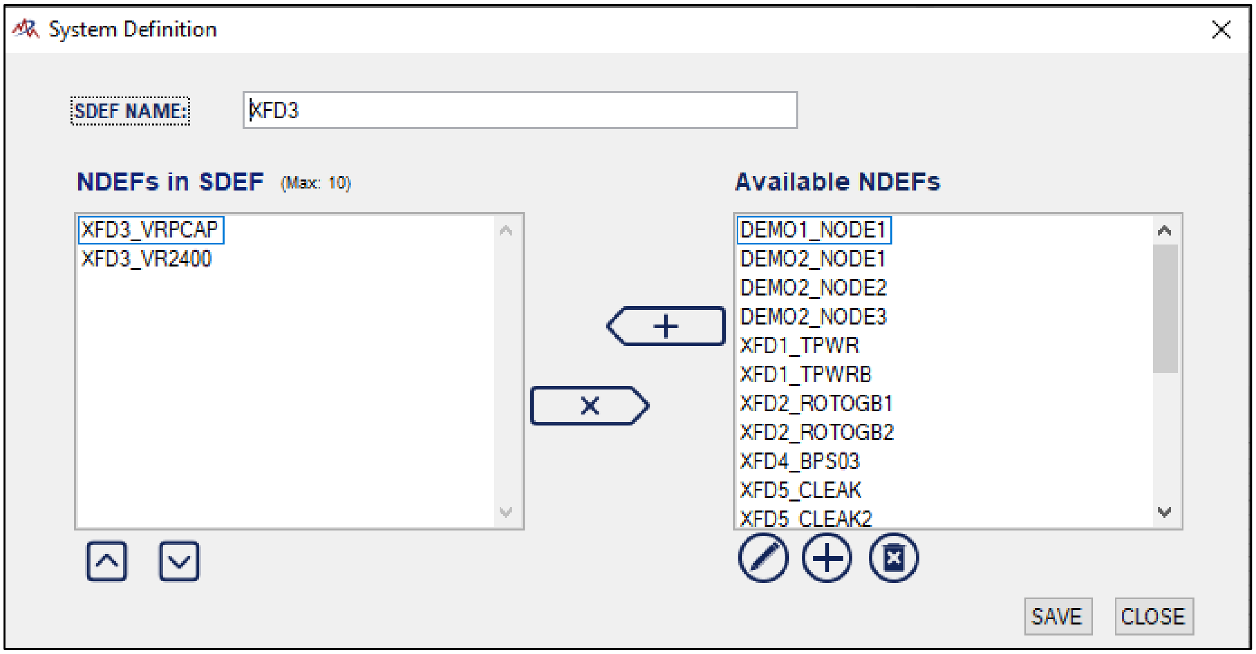


Figure 105 – XFD3 System Definition (SDEF) User Form

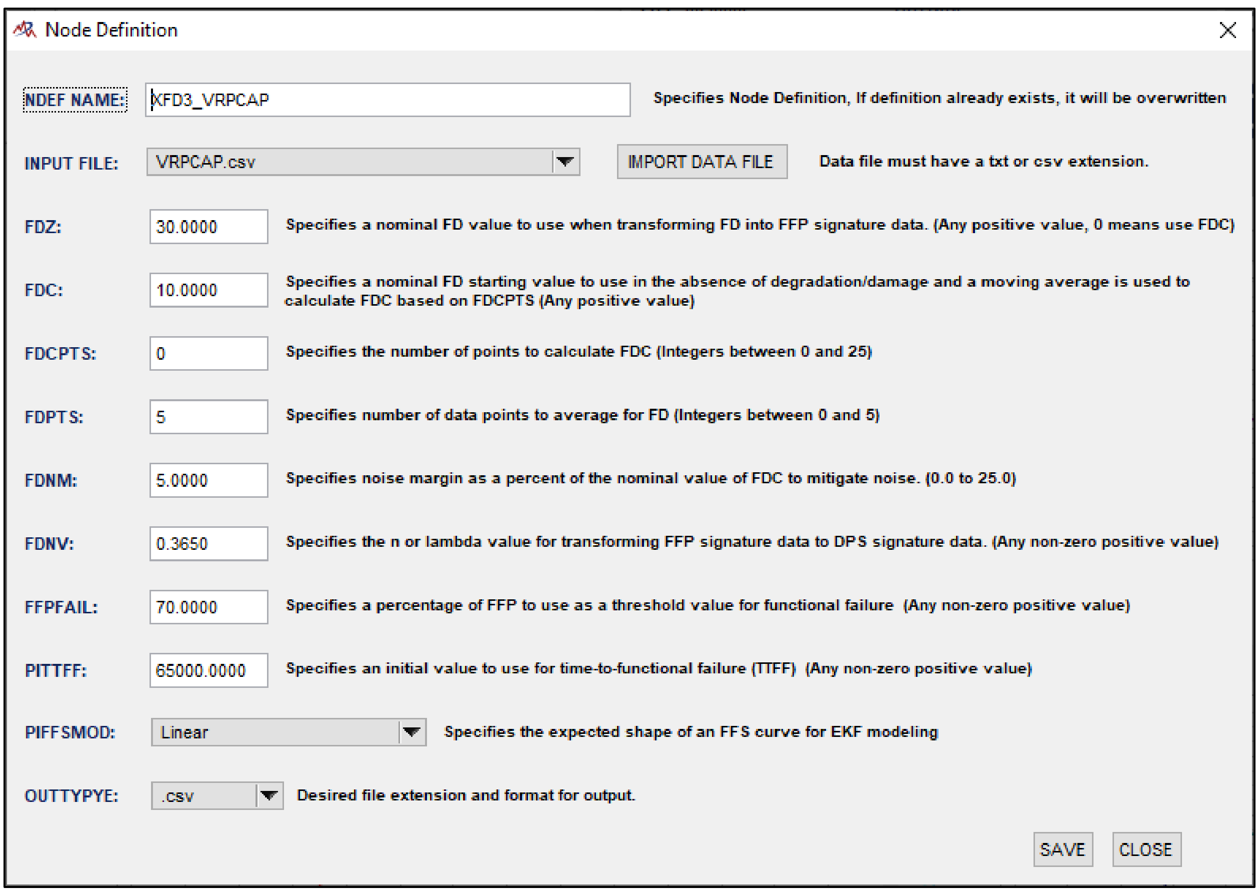


Figure 106 – Node Definition (NDEF) User Form for First Node in XFD3

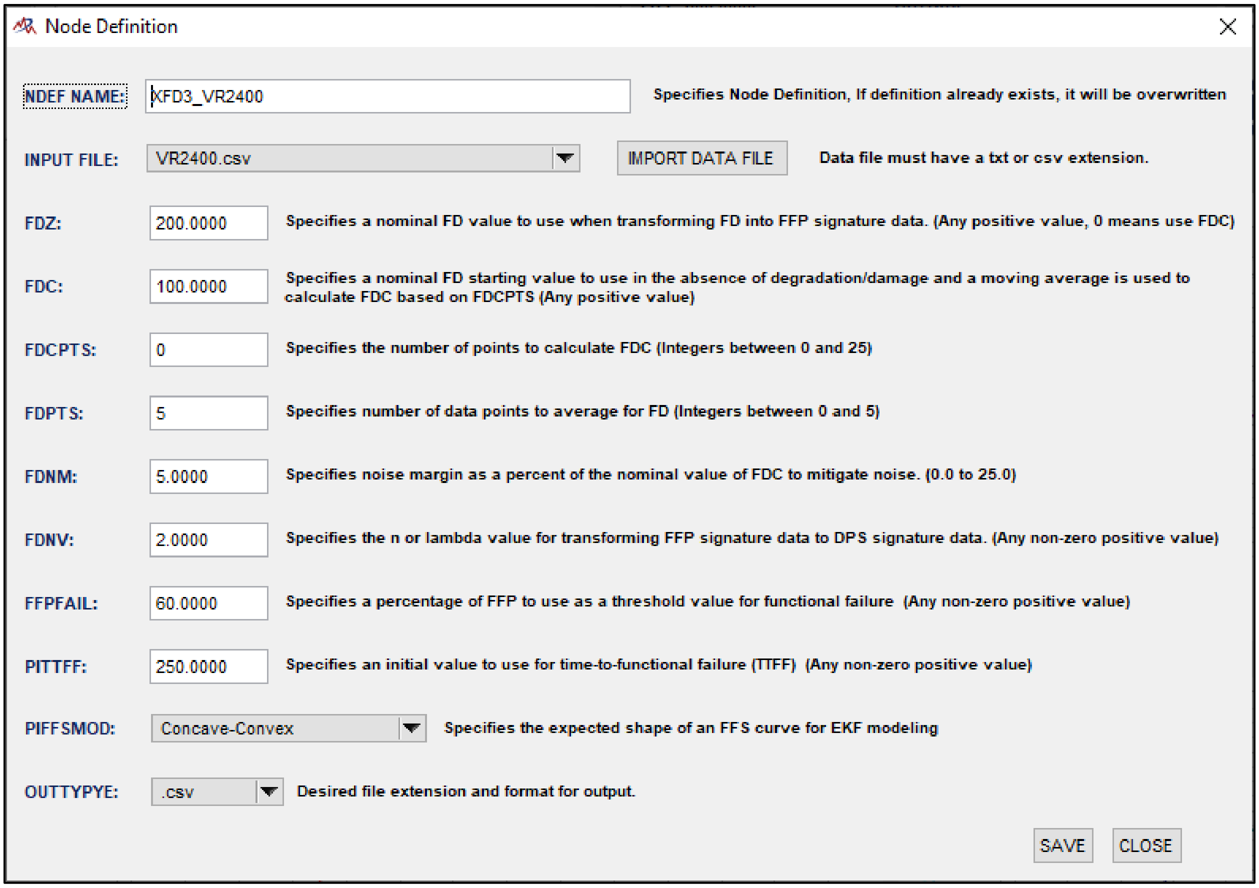


Figure 107 – Node Definition (NDEF) User Form for Second Node in XFD3

6.5.2 XFD3: Input Feature Data

Figure 108 and Figure 109 show the Condition–Based Data input plots for **XFD3_VRCAP** and **XFD3_VR2400** respectively.

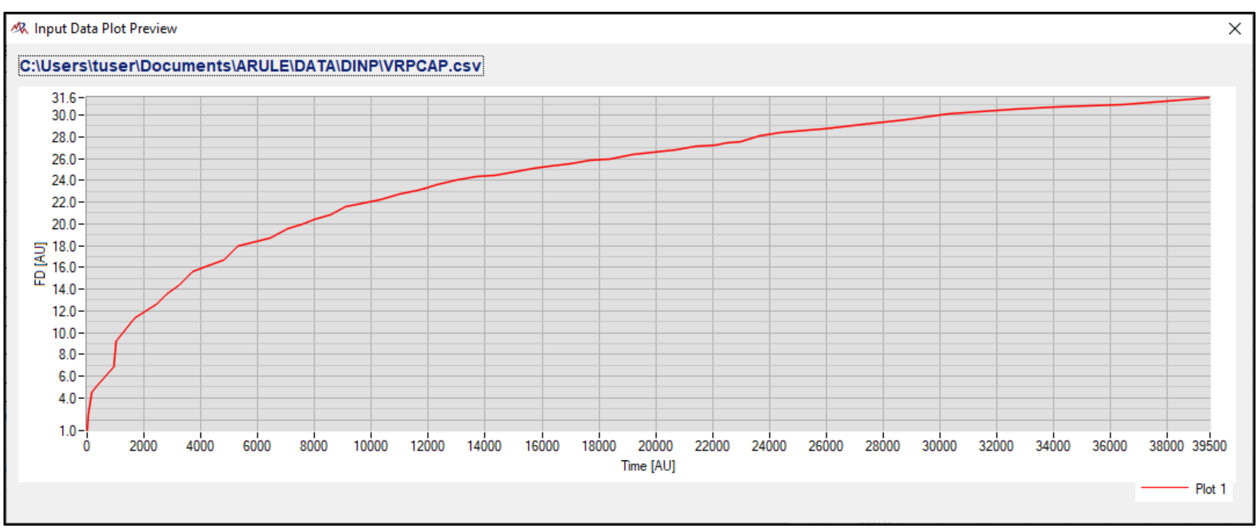


Figure 108 – XFD3 Input Data Plot Preview for Node 1

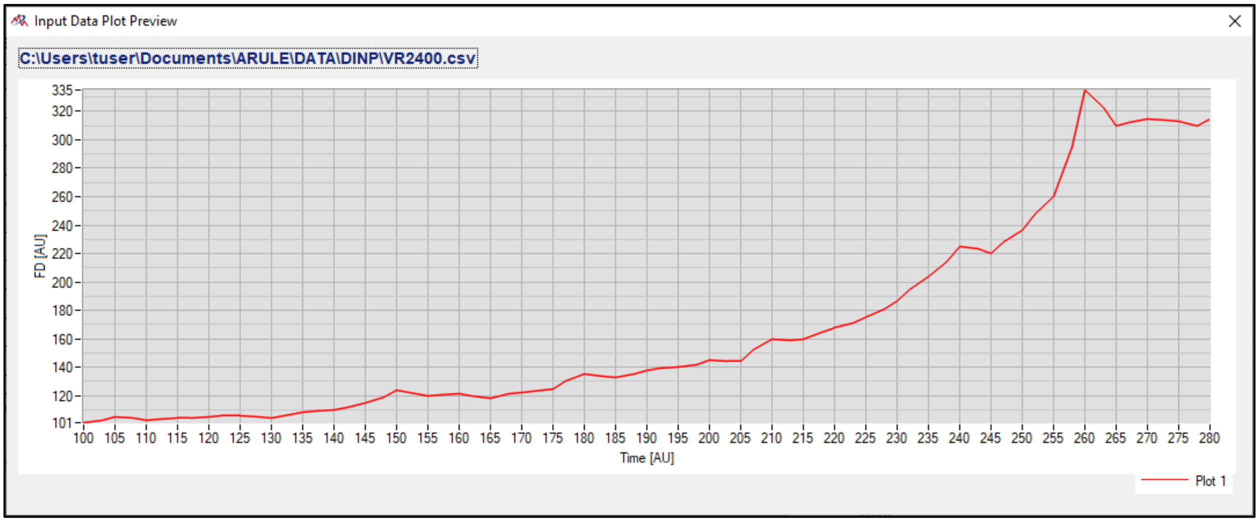


Figure 109 – XFD3 Input Data Plot Preview for Node 2

6.5.3 XFD3: Data Outputs and Prognostic Information

Figure 110 and Figure 111 show the Data Output for the two nodes in XFD3: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

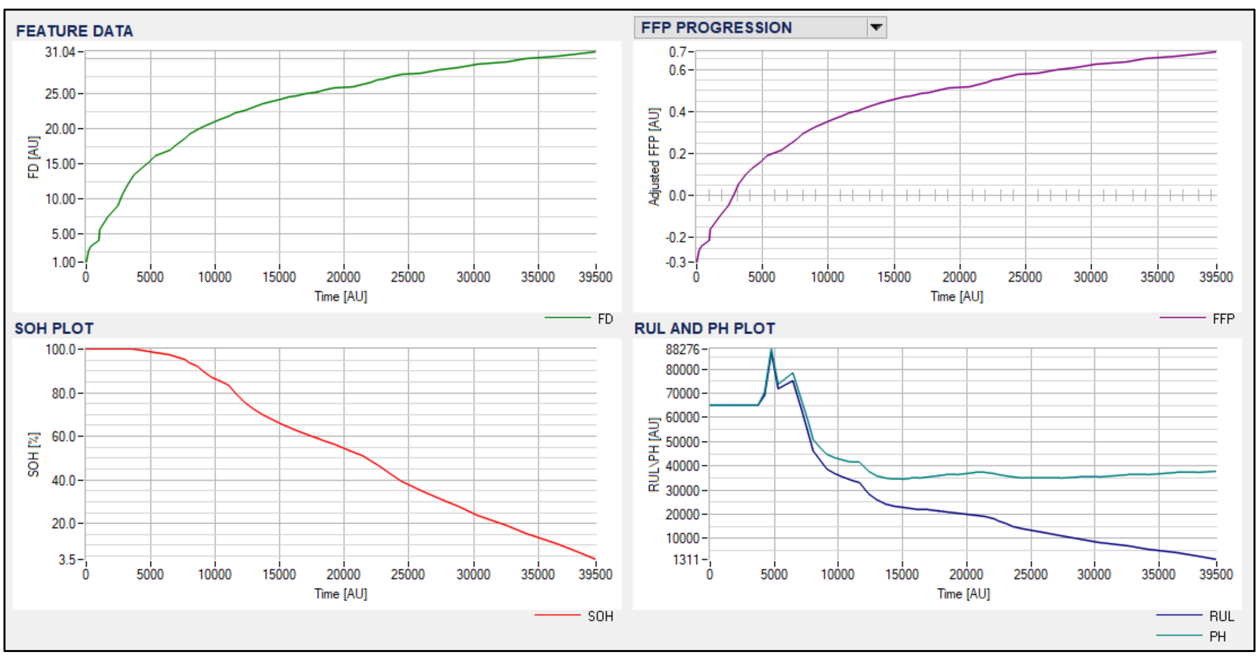


Figure 110 – Plots of the XFD3 Prognostic Information for the First Node

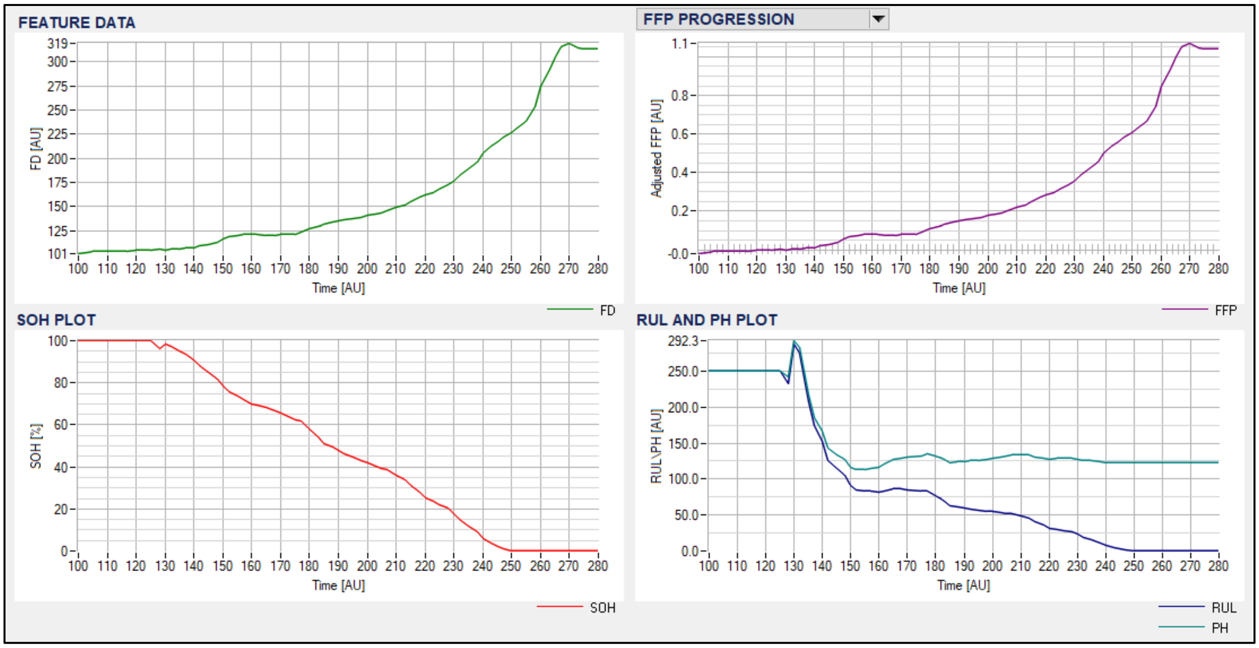


Figure 111 – Plots of the XFD3 Prognostic Information for the Second Node

6.5.4 XFD3: Discussion

This real–world example processes FD correlated to change in the amplitude of ripple voltage in the output of a power supply in response to loss of filtering capacitance. Analyses of this failure mode suggests a square root type of model, but experimentation suggests a higher model value yields more accurate prognostic information. That value offsets the significant step–like appearance in the ripple voltage during the initial onset of degradation, which is the result of stepped–fault injection to cause loss of filtering capacitance.

The two–stage ARULE™ APK is sufficiently robust to produce fairly accurate prognostic estimates without having to specify an exact model and parameter values as evidenced by comparing the output results shown in Figure 110 and Figure 111. Also note the comparison of the pseudo–ideal set of plots shown in Figure 29 on page 31.

6.6 XFD4: Ripple Voltage, Helicopter Power Supply, Half–PI filter

As mentioned in XFD3, power supply systems have 6 different types of feature data that can be analyzed for degradation. This example focuses on the ripple–voltage type of Feature Data (FD). This type of FD from power supplies has two distinct forms of characteristic signatures when filtering capacitance is reduced: when the output filter of the supply is a full–pi circuit (capacitance, inductance, capacitance) or when the output filter of the supply is a half–pi circuit (inductance, capacitance) – step appearances occur because the filtering capacitance was reduced by electronically disconnecting parallel–connected capacitors one at a time. This ARULE™ example has also been proven through Ridgetop's Sentinel Power product line.

6.6.1 XFD4: Definitions

As shown in Figure 112, the SDEF file for this fourth example points to a node definition number of 1, and an NDEF file named **XFD4_BPS03**. Figure 113 shows the NDEF parameters and keywords used for XFD4.

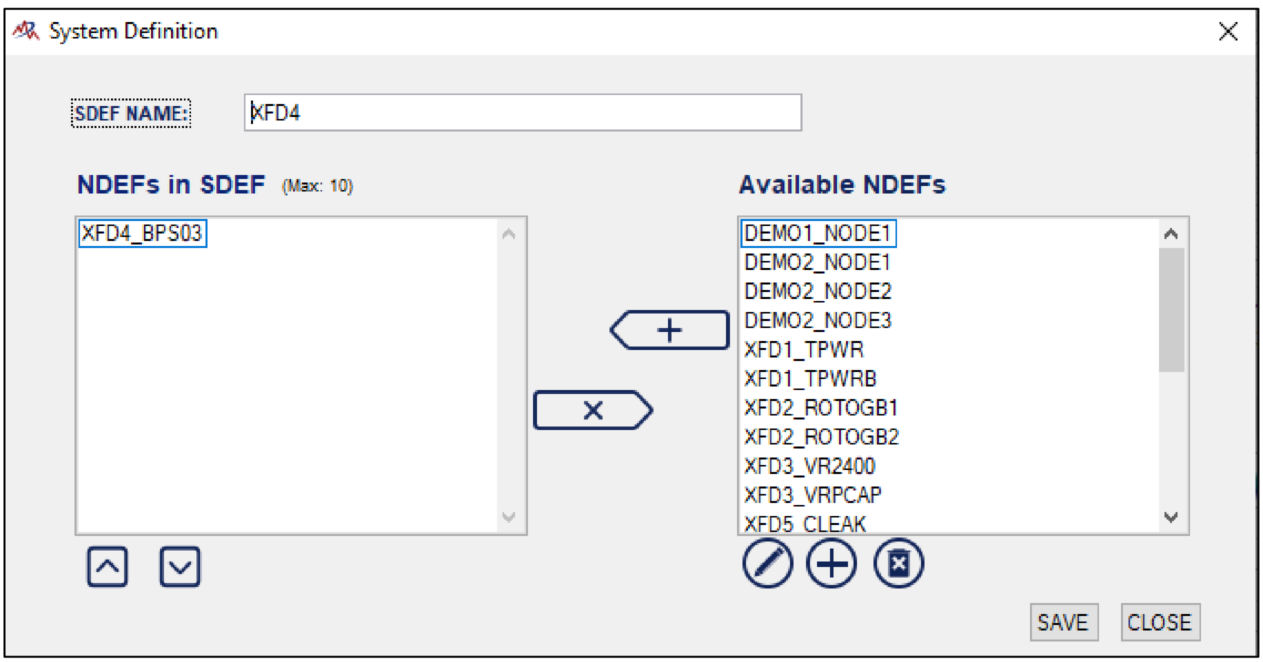


Figure 112 – XFD4 System Definition (SDEF) User Form

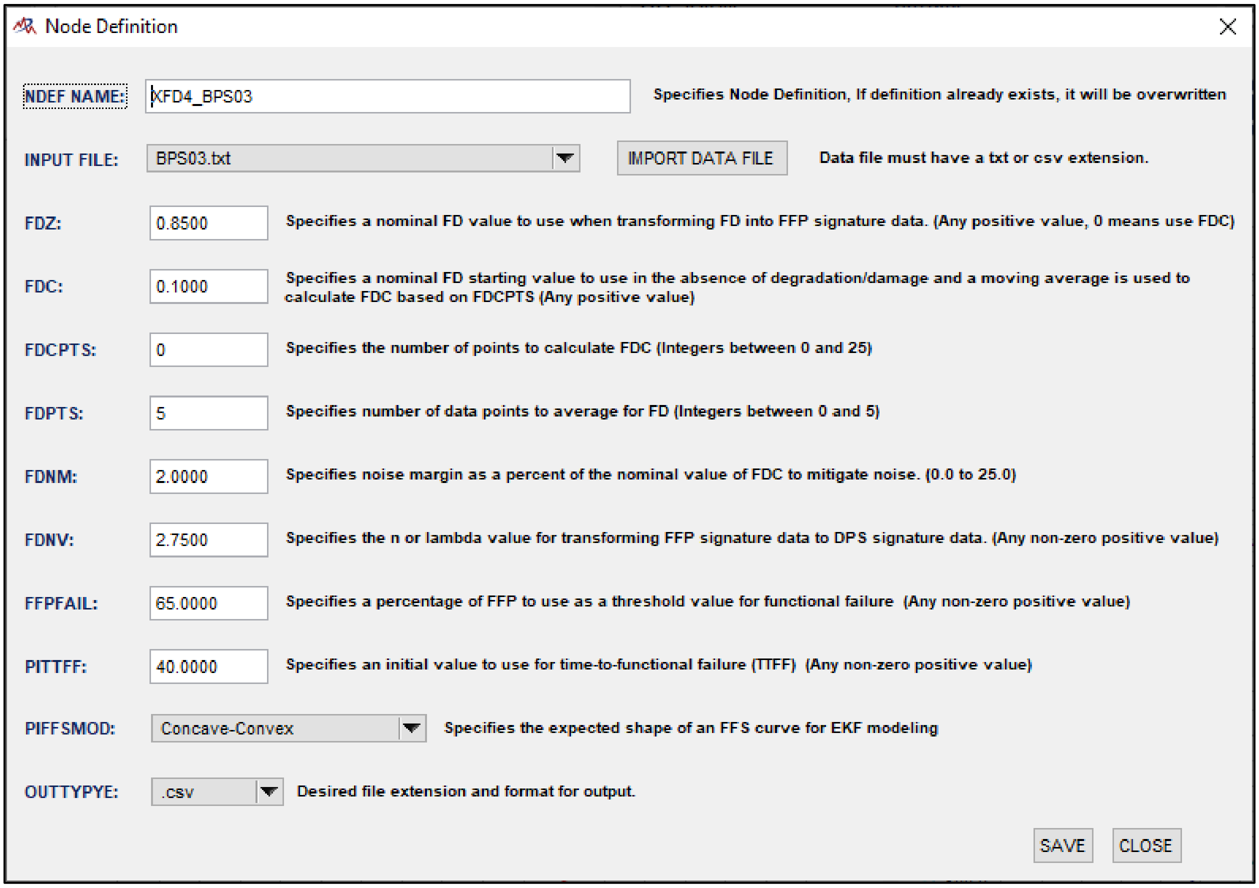


Figure 113 – XFD4 Node Definition (NDEF) User Form

6.6.2 XFD4: Input Feature Data

Figure 114 shows the Condition–Based Data input plot for **XFD4_BPS03**.

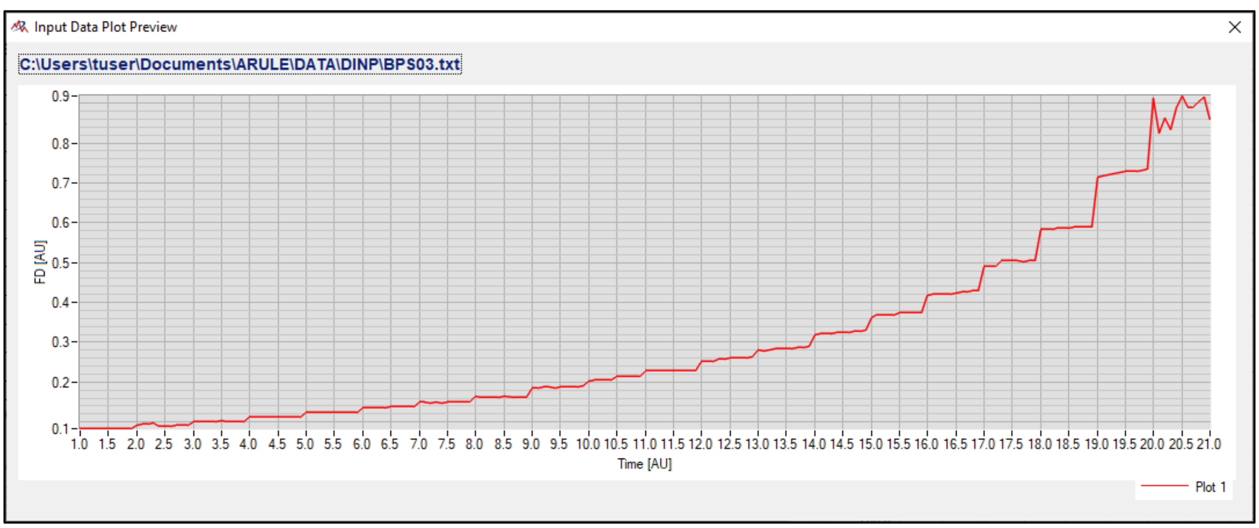


Figure 114 – XFD4 Input Data Plot Preview

6.6.3 XFD4: Data Outputs and Prognostic Information

Figure 115 shows the Data Output for the only node in XFD4: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

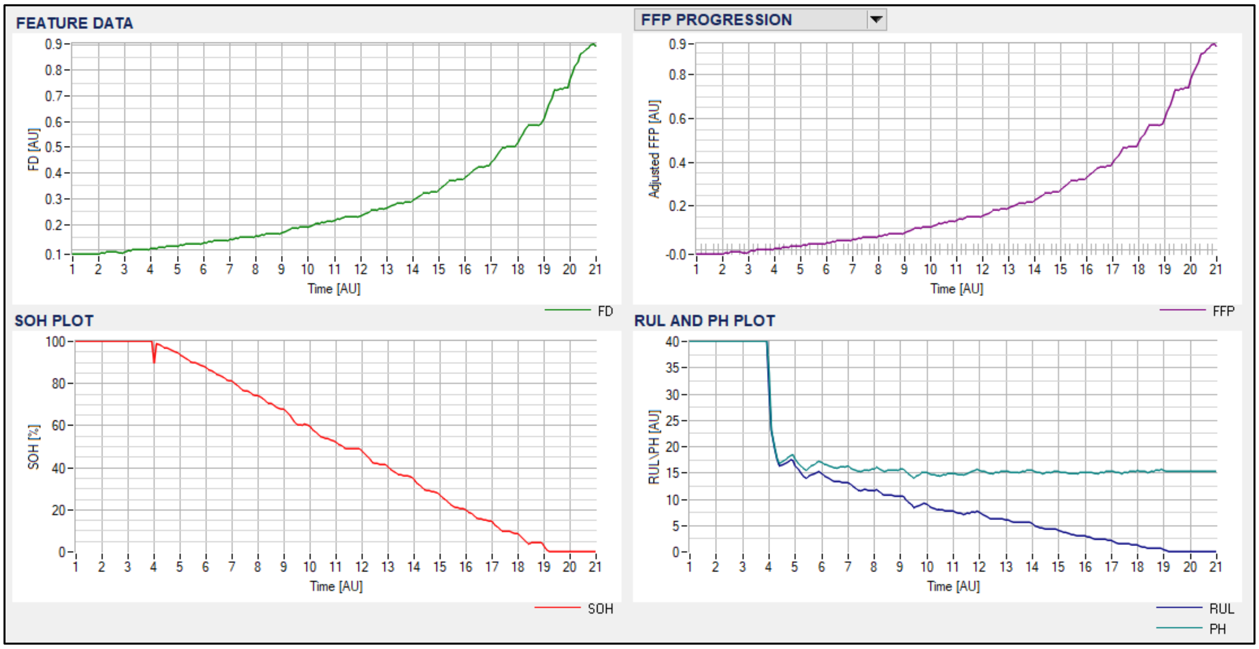


Figure 115 – Plots of the XFD4 Prognostic Information

6.6.4 XFD4: Discussion

The results of processing the input data are evaluated as excellent. Note the difference in the characteristic curve of the input data in this example (Figure 114) compared to the previous example (Figure 108): the curves are different (convex versus concave) because of the difference in the configuration of the output filter (full PI versus half PI) and this example exhibits a pronounced "step" appearance because the loss of filtering capacitance was caused by abruptly disconnecting and connecting capacitors in the filter (a fault–injection technique).

6.7 XFD5: Capacitor Leakage, Power Supply Filter

This ARULE™ example is another proven use case where Ridgetop has used the Sentinel Power product line to monitor and detect a common failure mode in power supply systems. Filtering capacitors, especially electrolytic type of capacitors, in power supplies are subject to a failure mode known as capacitor leakage. This failure mode is often the result of degradation of the dielectric material, and Ridgetop has developed the necessary hardware, firmware, and software solutions to prognostic enable several different types of power supply systems in the military, aerospace, and commercial market segments.

6.7.1 XFD5: Definitions

As shown in Figure 116, the SDEF file for this example points to 2 nodes with NDEF files namely **NODE_CLEAK** and **NODE_CLEAK2**. They are varied by changing the FDC parameter from 10 in **NODE_CLEAK** to 18 in **NODE_CLEAK2.** Figure 117 and Figure 118 show the NDEF parameters and keywords used for XFD5. The previous examples had file names associated with specific SDEF files: such association is unnecessary and might be detrimental were an NDEF be used by more than one SDEF.

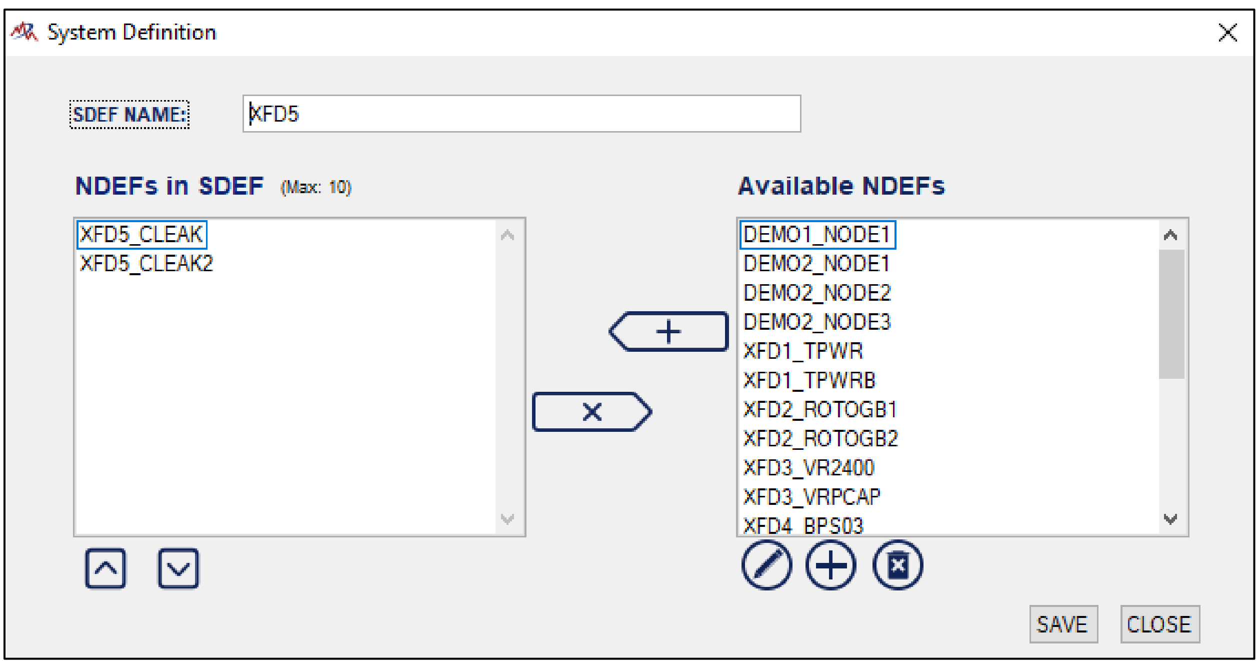


Figure 116 – XFD5 System Definition (SDEF) User Form

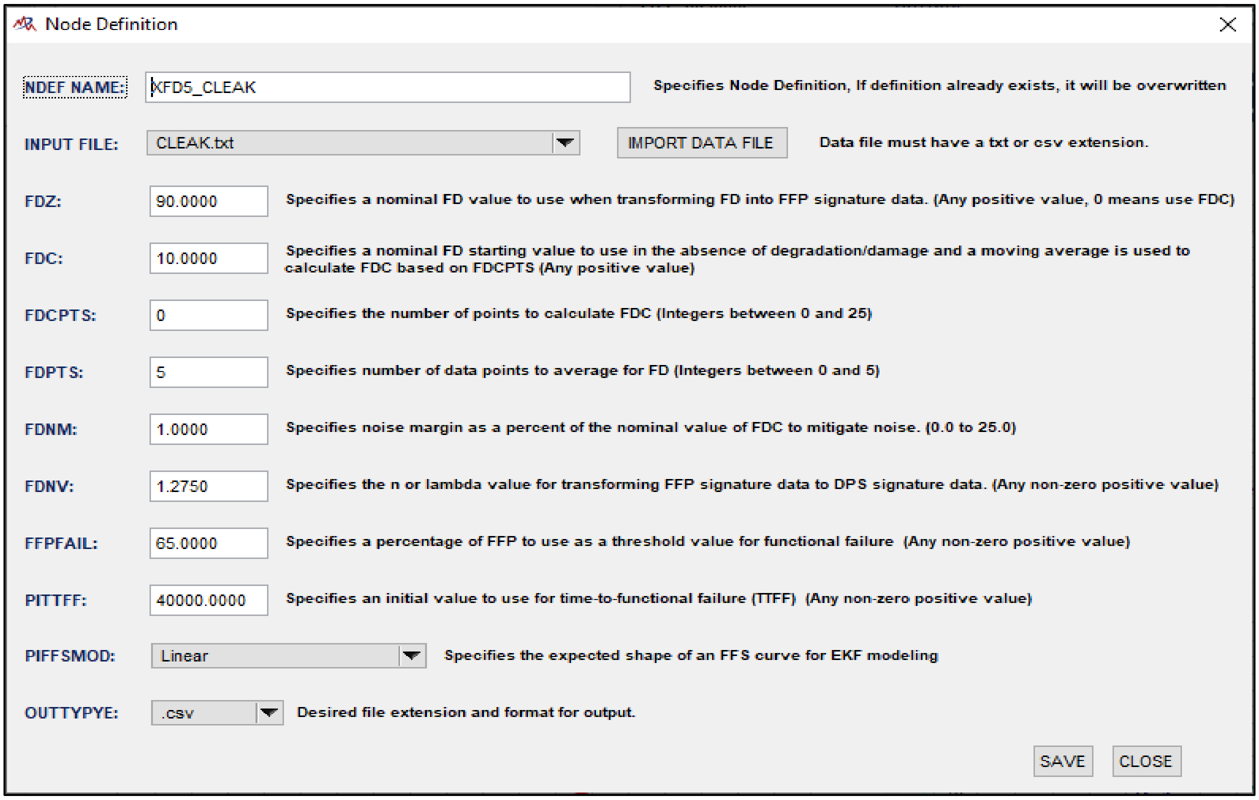


Figure 117 – Node Definition (NDEF) User Form for First Node in XFD5

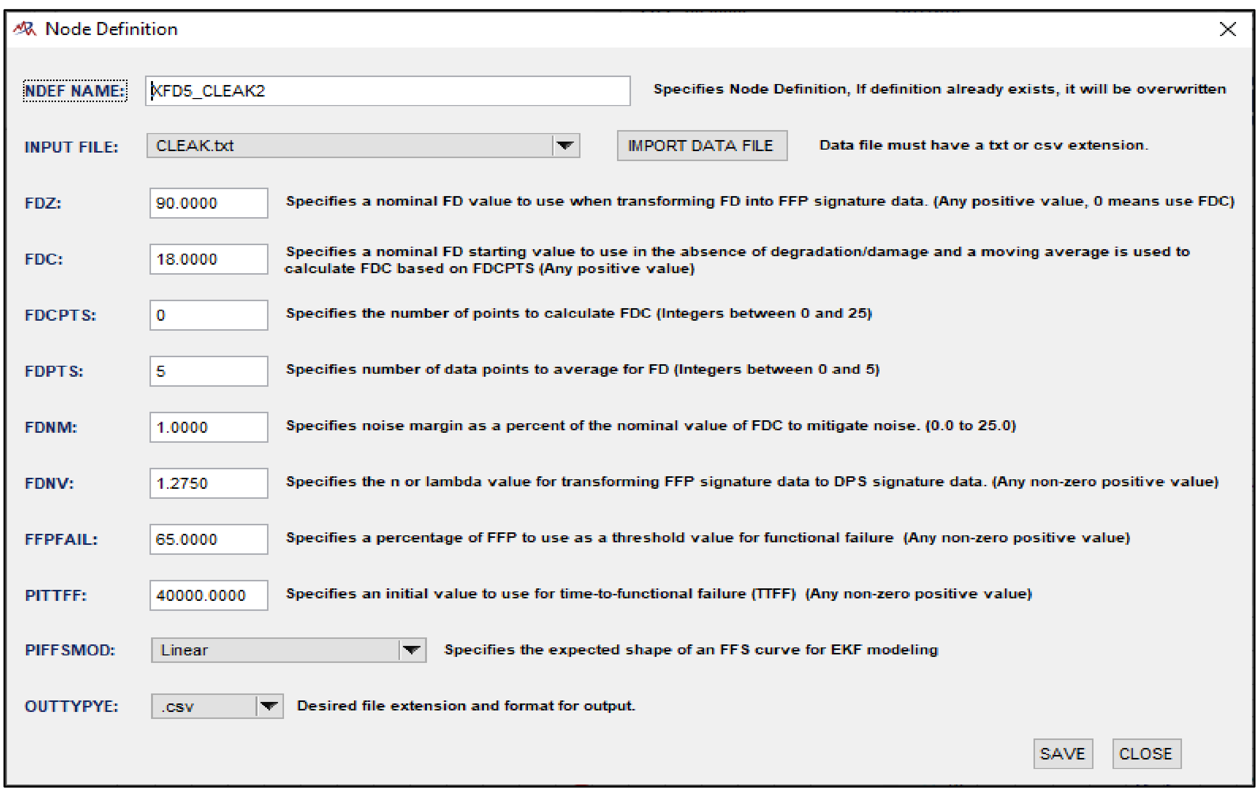


Figure 118 – Node Definition (NDEF) User Form for Second Node in XFD5

6.7.2 XFD5: Input Feature Data

Figure 119 shows the Condition–Based Data input plot for **NODE_CLEAK** and **NODE_CLEAK2** respectively.

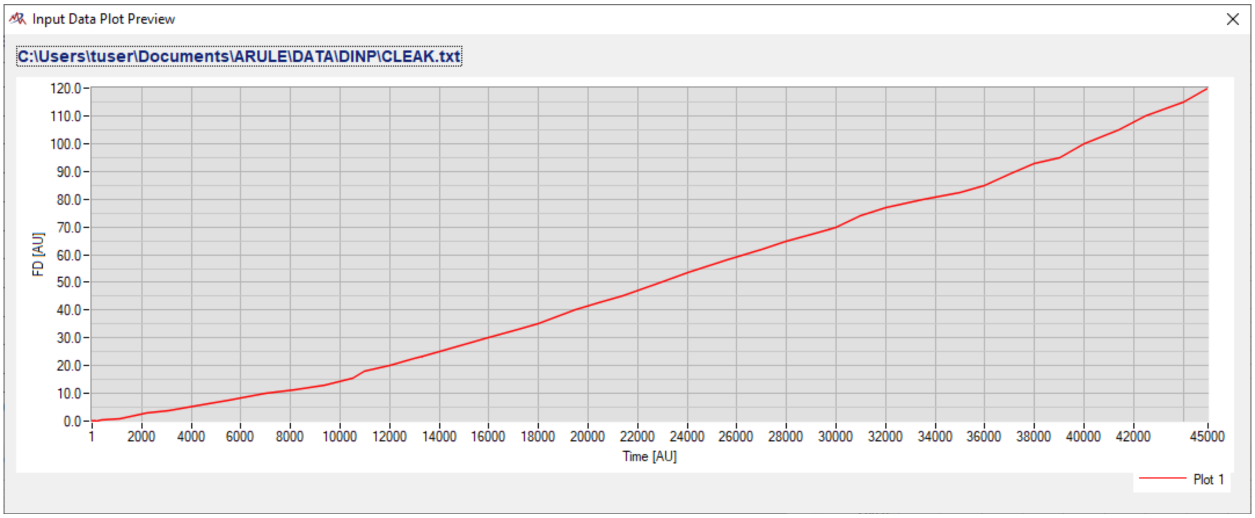


Figure 119 – XFD5 Input Data Plot Preview for Node 1 and Node 2

6.7.3 XFD5: Data Outputs and Prognostic Information

Figure 120 and Figure 121 show the Data Output for the two nodes in XFD5: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

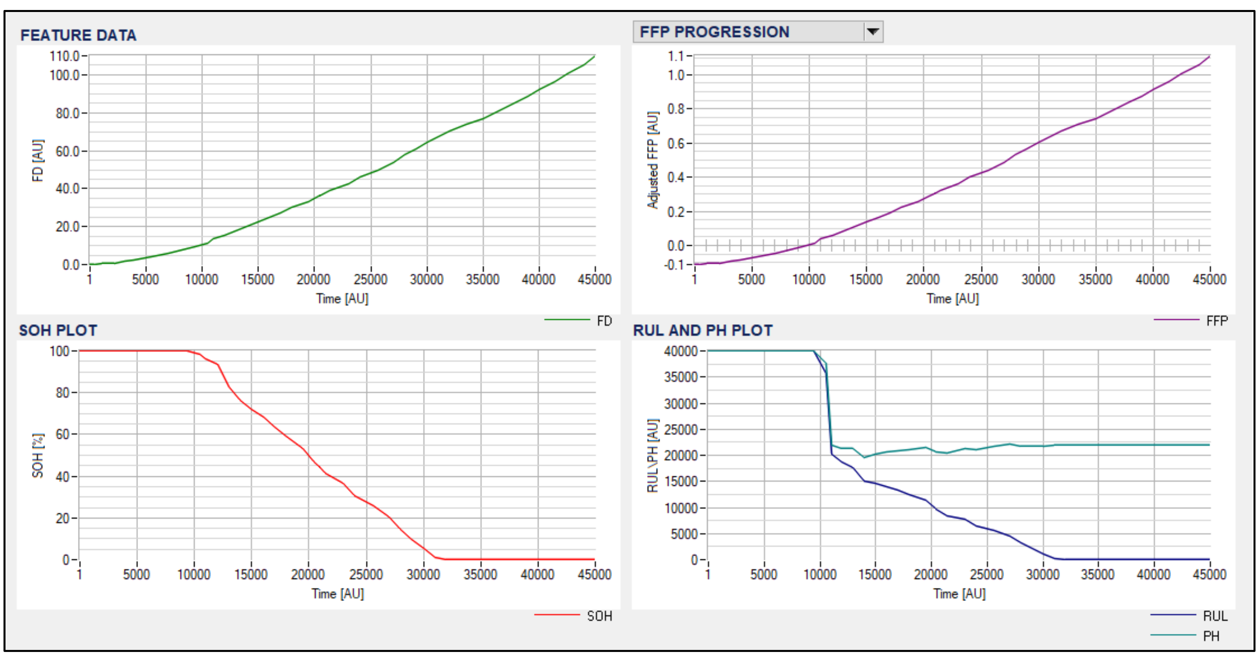


Figure 120 – Plots of the XFD5 Prognostic Information for the First Node

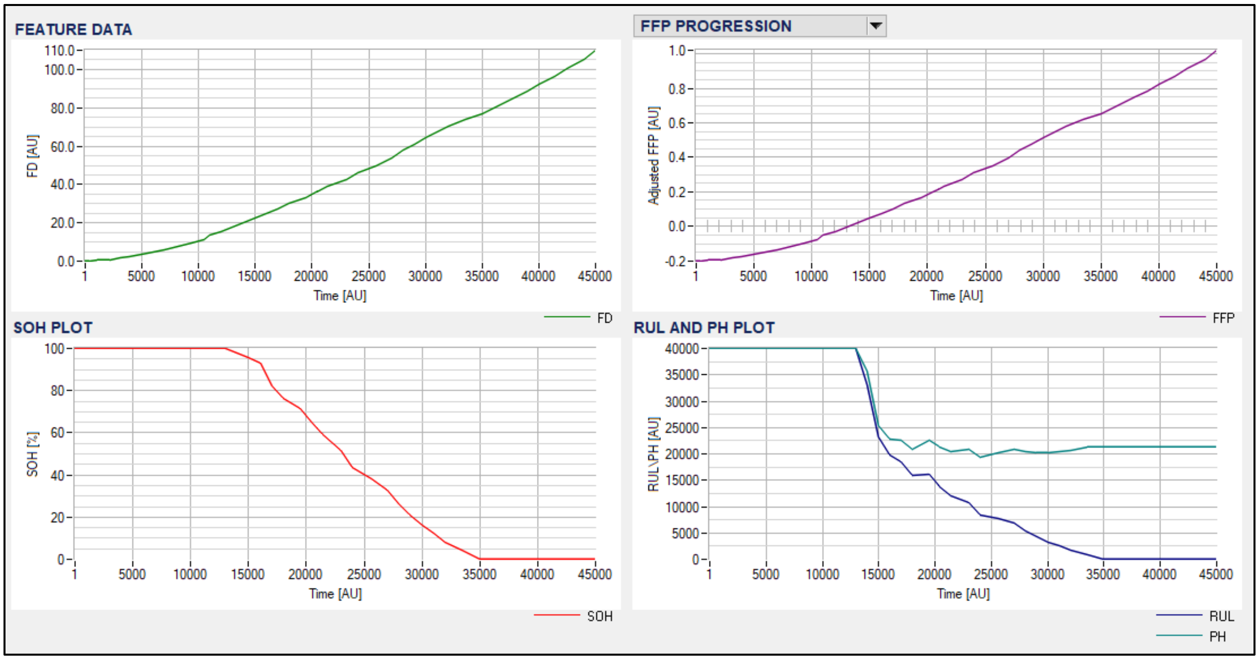


Figure 121 – Plots of the XFD5 Prognostic Information for the Second Node

6.7.4 XFD5: Discussion

Referring back to Figure 119, the signature of the input data is fairly linear, so we do not need a model to linearize the input data. The results shown in Figure 120 and Figure 121 are very good, which confirms that a transformation model was not needed. FMEA (Failure Mode Effects Analysis) indicates that leakage current is typically below 1 mA to 5mA, and can be as high as 100 mA before evidence of physical failure.

6.8 XFD6: Battery Internal Resistance

The next two ARULE™ examples, XFD6 and XFD7, are related to Ridgetop's prognostic solutions for batteries. Batteries are subject to a number of different failure modes including one called Internal Resistance/Internal Contact Resistance (IR/ICR). As the IR/ICR increases in batteries the output voltage of the battery is reduced as it supplies current to a load. This example demonstrates how the IR/ICR type of feature data (FD) can be analyzed with ARULE™, but this type of prognostic solution could also be implemented as an embedded process in a smart Battery Management System (BMS) design. This engineering service is offered through Ridgetop's CellSage product line, which also encompasses a suite of battery simulation and analysis software tools.

6.8.1 XFD6: Definitions

As shown in Figure 122, the SDEF file for this example points to one node and an NDEF file named **NODE_BATTIR**. Figure 123 shows the NDEF parameters and keywords used for XFD6.

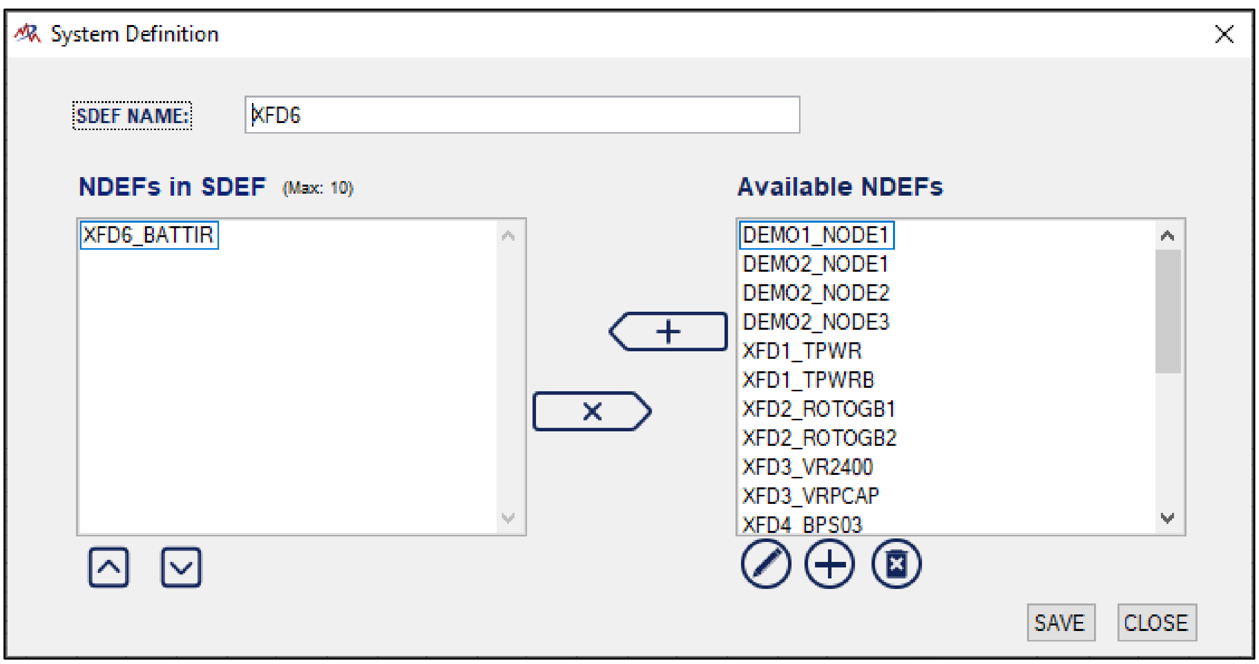


Figure 122 – XFD6 System Definition (SDEF) User Form

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

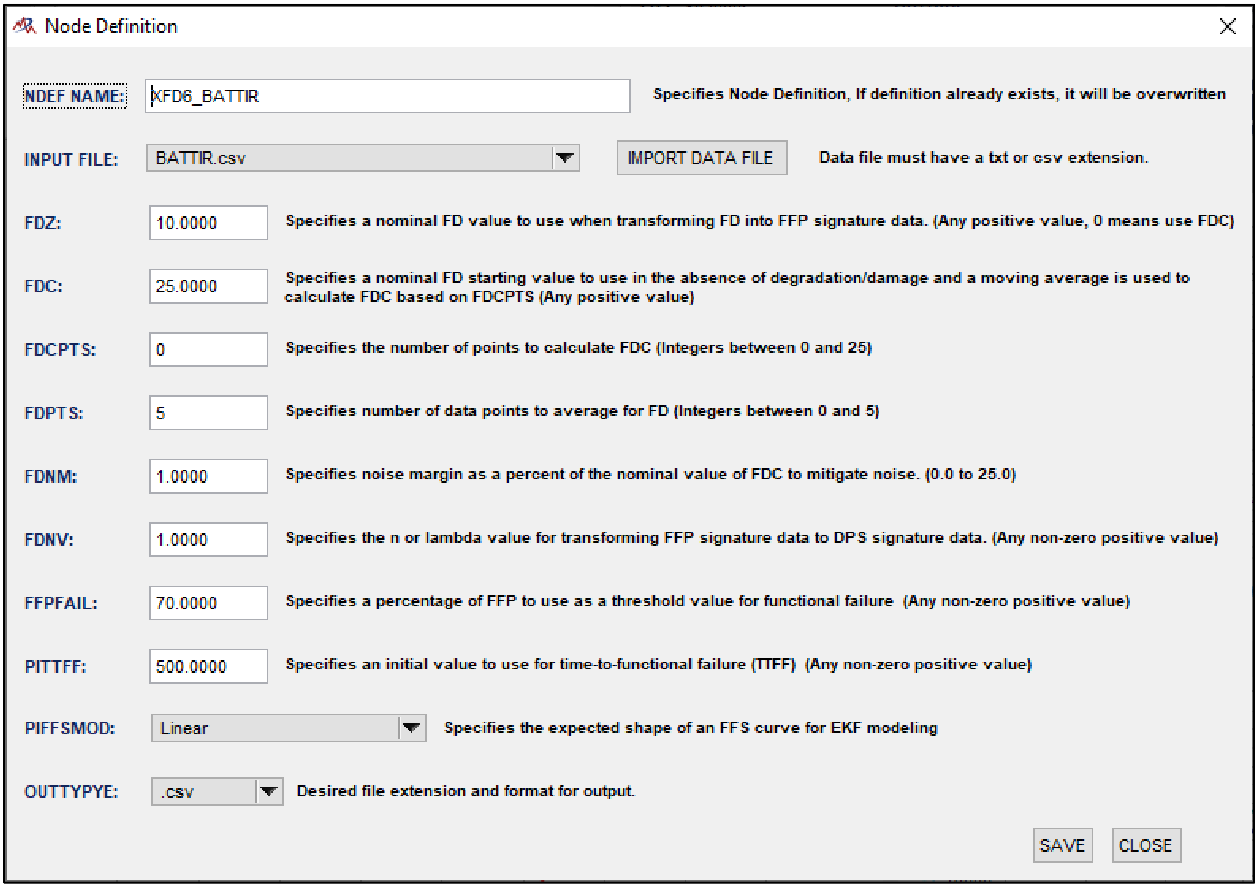


Figure 123 – XFD6 Node Definition (NDEF) User Form

6.8.2 XFD6: Input Feature Data

Figure 124 shows the Condition–Based Data input plot for **XFD6_BATTIR**.

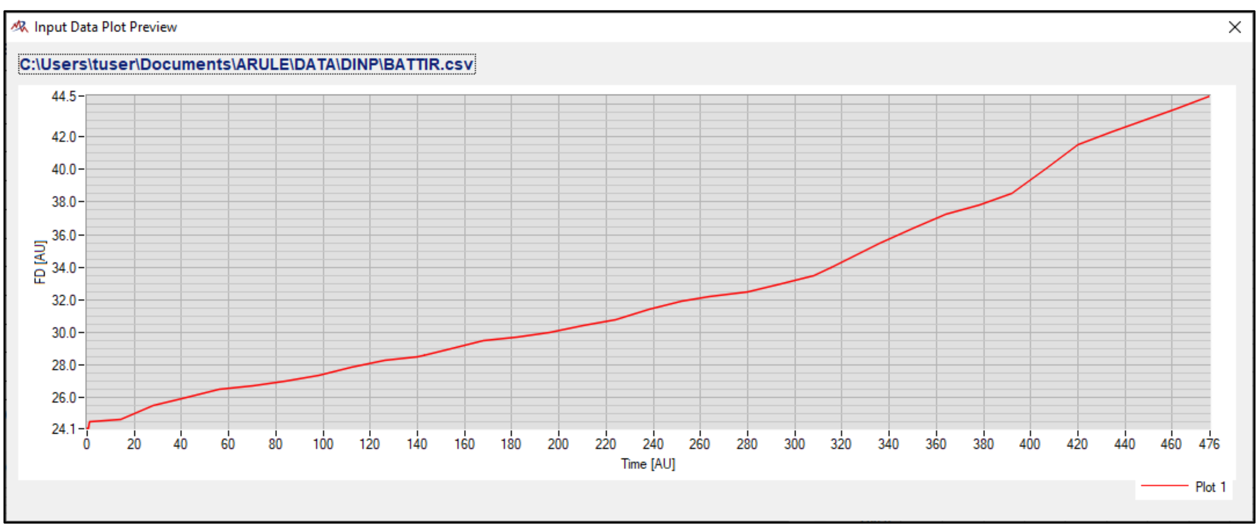


Figure 124 – XFD6 Input Data Plot Preview

6.8.3 XFD6: Data Outputs and Prognostic Information

Figure 125 shows the Data Output for the only node in XFD6: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

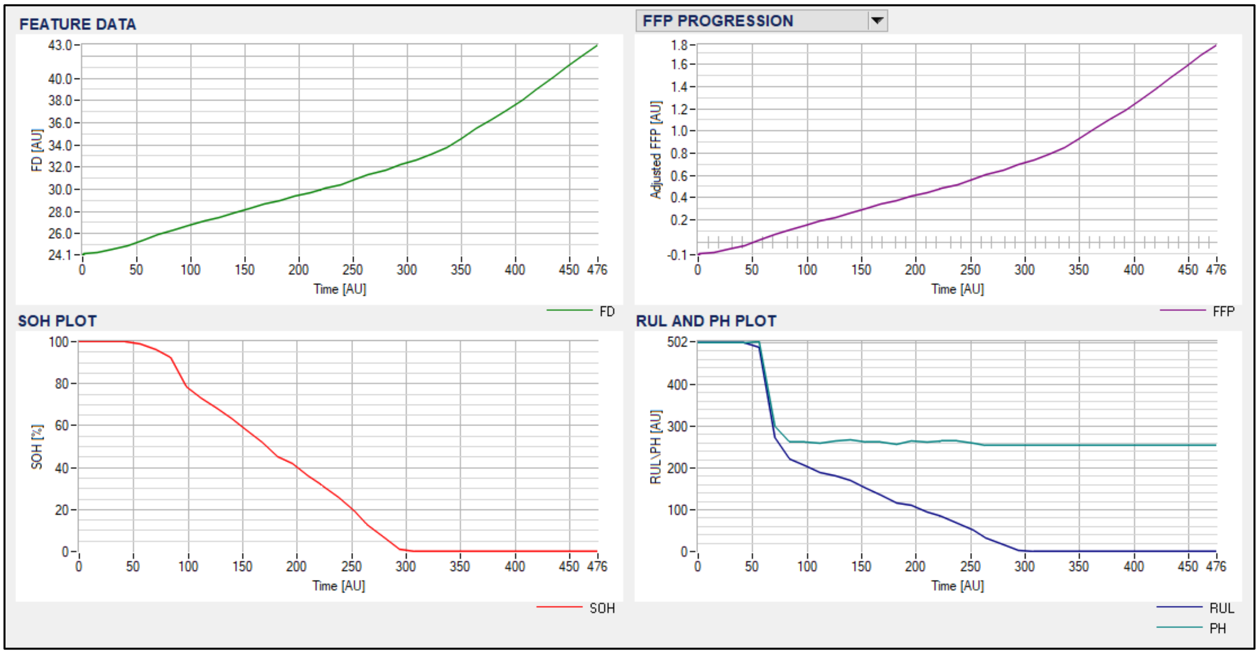


Figure 125 – Plots of the XFD6 Prognostic Information

6.8.4 XFD6: Discussion

Referring back to Figure 124, the signature of the input data is fairly linear. The input signature exhibits an inflection point around day 80, which is reflected in the SoH estimates plotted in Figure 125 – despite that inflection, the estimates of RUL and PH produced by the ARULE™ APK are very, very accurate.

6.9 XFD7: Battery Charging Capacity

As lithium–ion types of batteries degrade, in addition to increasing internal contact resistance, they also exhibit loss of charging capacity (see Figure 126). As covered in Ridgetop's CellSage product line, the loss of charging capacity is impacted by several different environmental and operational parameters. Such parameters include cell chemistry, temperature, state–of–charge, charging protocols, and application dependent duty cycles. Although each of these parameters have a known impact on battery ageing and degradation, the loss of charge capacity will typically follow an aging path that can be modeled by the single set of ARULE™ degradation signatures.

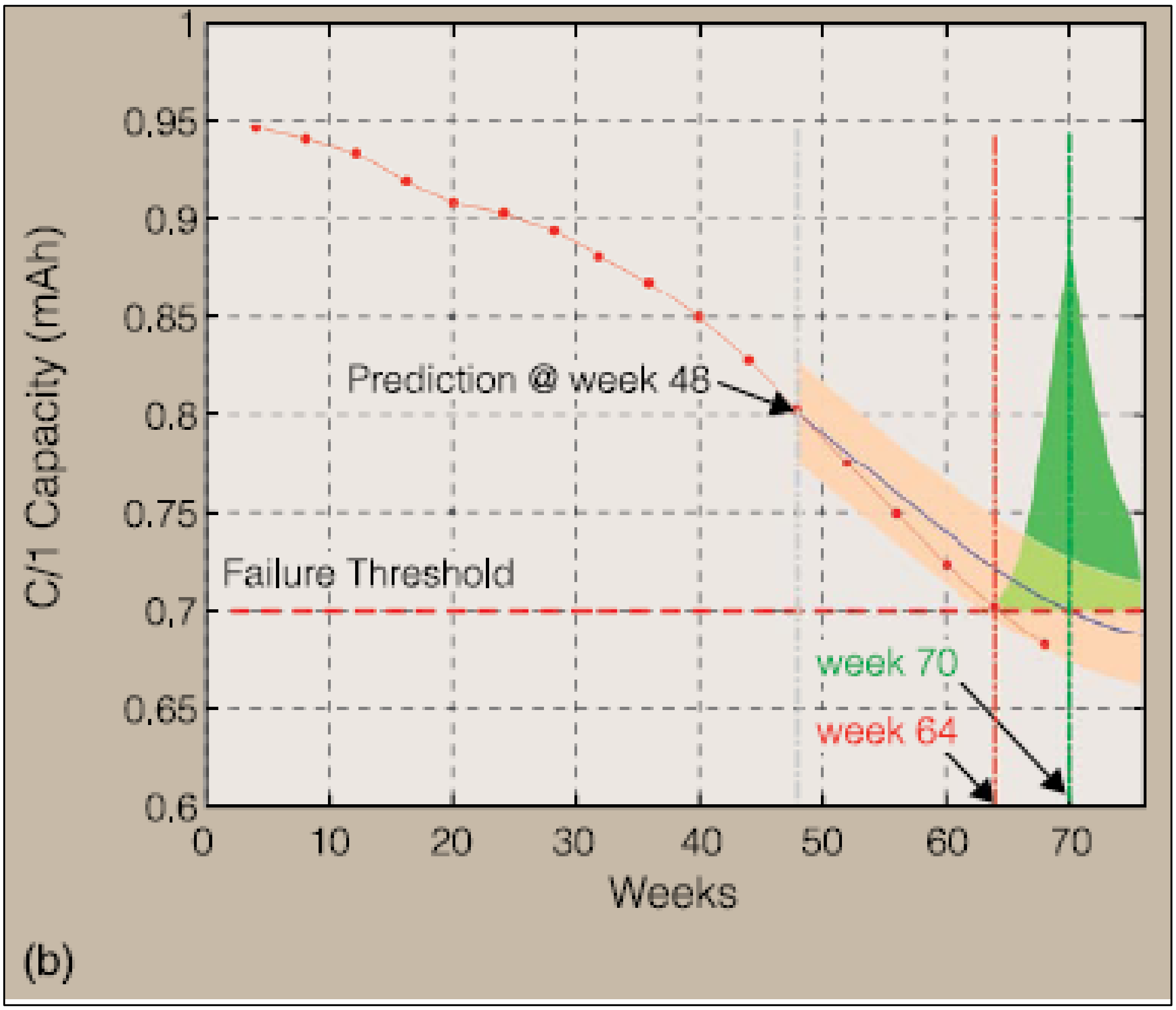


Figure 126 – Published data: battery capacity [Goebel, et.al, 2008, Prognostics in Battery Health Management, IEEE Instrumentation and Measurements Magazine]

This example will show how Ridgetop was able to successfully process capacity loss Feature Data (FD) from a design of experiments that were conducted through a NASA

SBIR program as well as simulated battery degradation data from CellSage. There are three nodes in this example: the first node uses input data derived from experimental data shown in Figure 126; the second and third nodes use worst-case and best-case sets of simulated data output from CellSage as shown in Figure 127. The FD values are in mAh values in Figure 126 versus percent values in Figure 127.

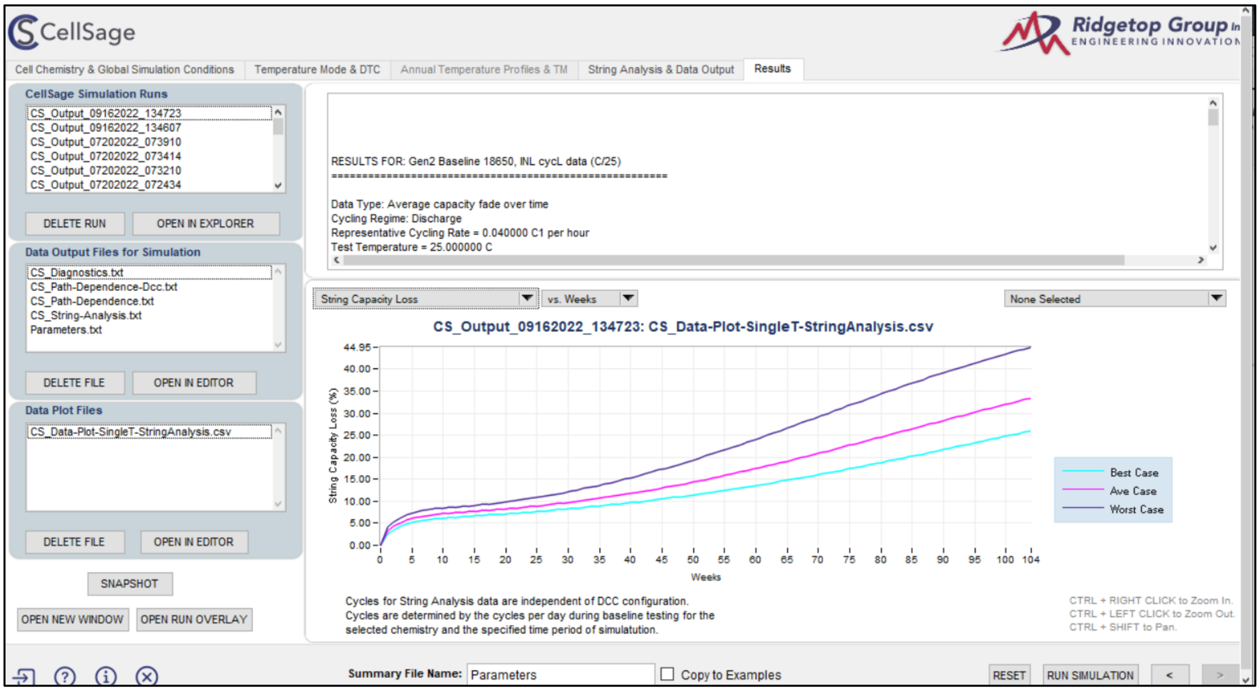


Figure 127 - CellSage screen shot showing loss of charge: annual temperature profile for Tucson, AZ for worst-case, average-case, and best-case simulation conditions

6.9.1 XFD7: Definitions

As shown in Figure 128, the SDEF file for this example points to 3 nodes with NDEF files named **XFD7_BATTCHG1**, **XFD7_BATTCHG2**, and **XFD7_BATTCHG3**. Figure 129 through Figure 131 show the NDEF parameters and keywords used for XFD7. The NDEF parameters are the same for all three cases except as follows:

1. The FDC value for Node 1 is specified as 0.07 (mAh) versus 10.0 (percent) for Node 2 and 3.
2. The FDZ value for Node 1 is specified as 0.20 (mAh) versus 23.0 (percent) for Node 2 and 3.

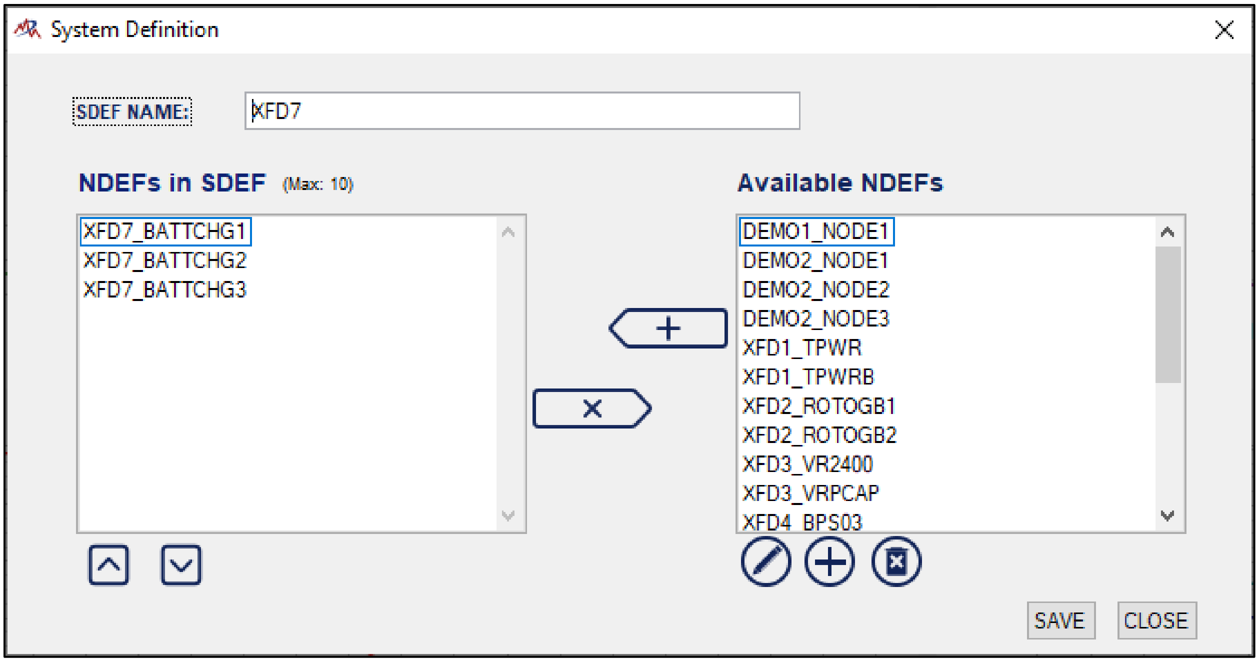


Figure 128 – XFD7 System Definition (SDEF) User Form

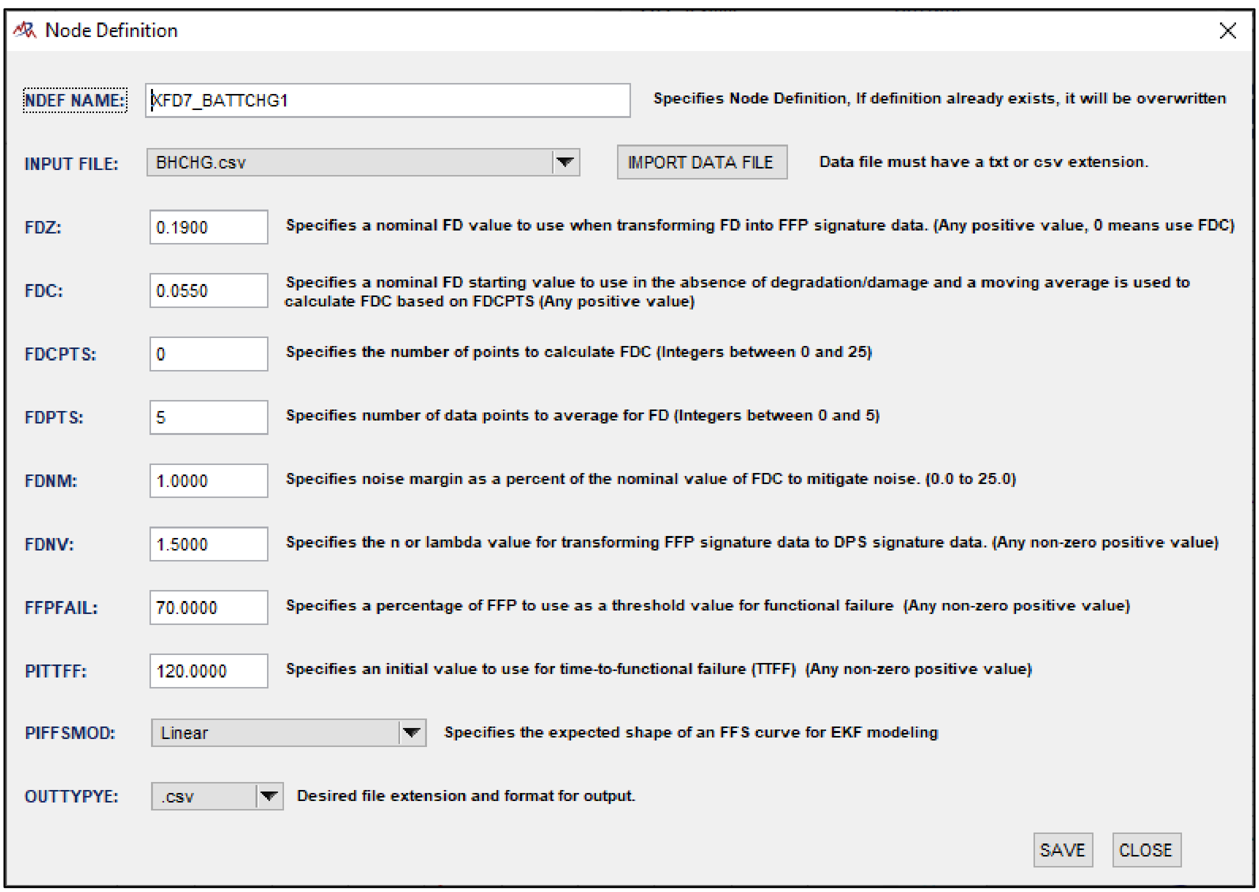


Figure 129 – Node Definition (NDEF) User Form for First Node in XFD7

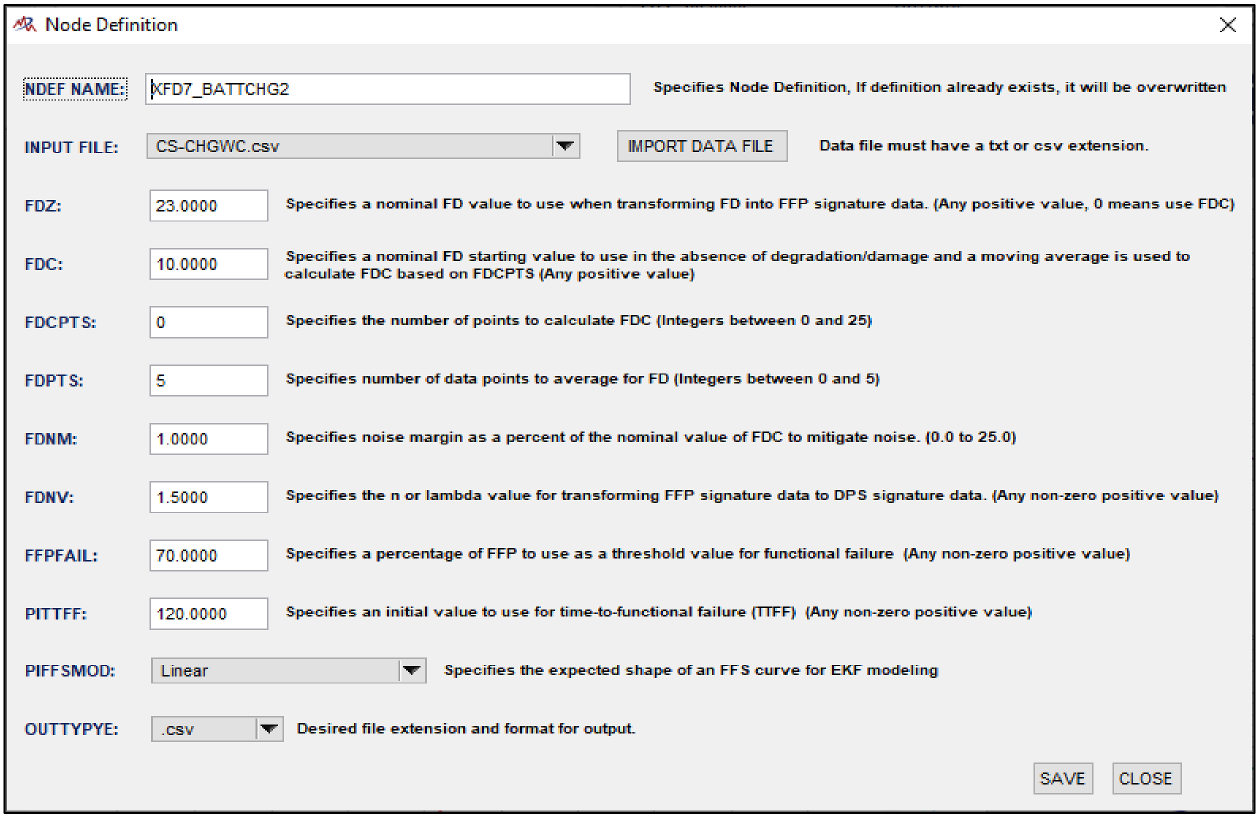


Figure 130 – Node Definition (NDEF) User Form for Second Node in XFD7

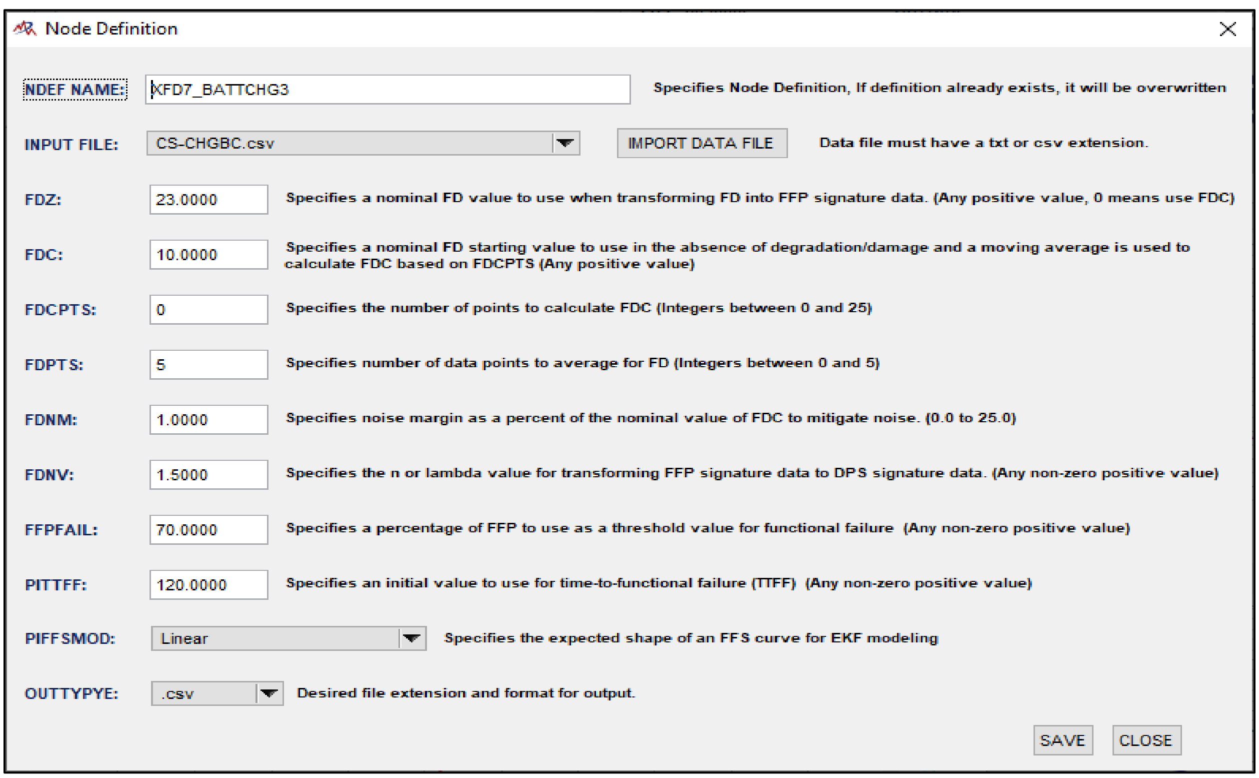


Figure 131 – Node Definition (NDEF) User Form for Third Node in XFD7

6.9.2 XFD7: Input Feature Data

Figure 132, Figure 133, and Figure 134 show the Condition–Based Data Input plots for **XFD7_BATTCHG1**, **XFD7_BATTCHG2**, and **XFD7_BATTCHG3** respectively.

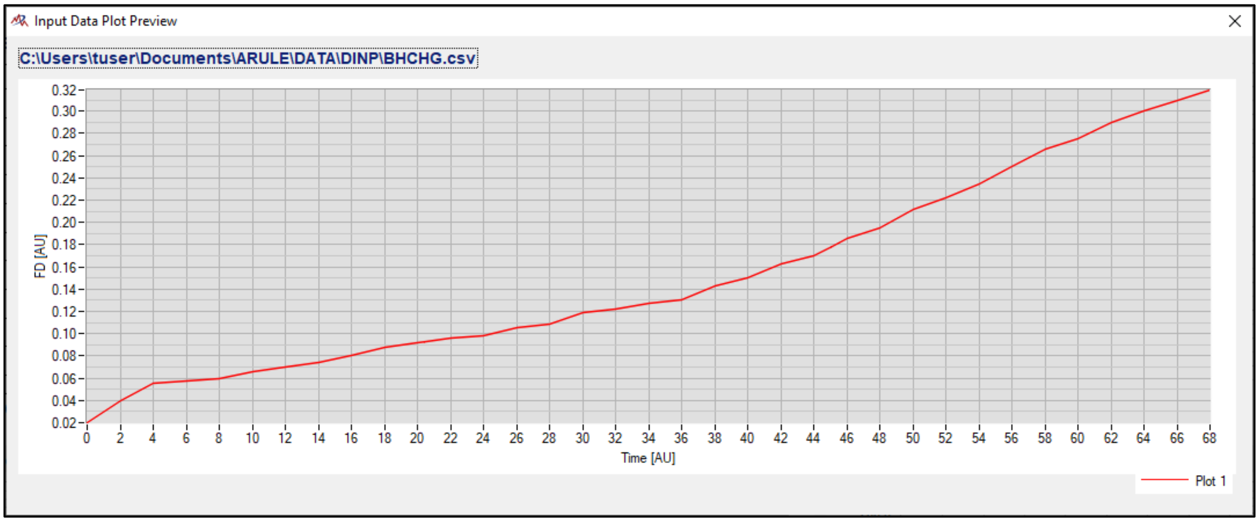


Figure 132 – XFD7 Input Data Plot Preview for Node 1

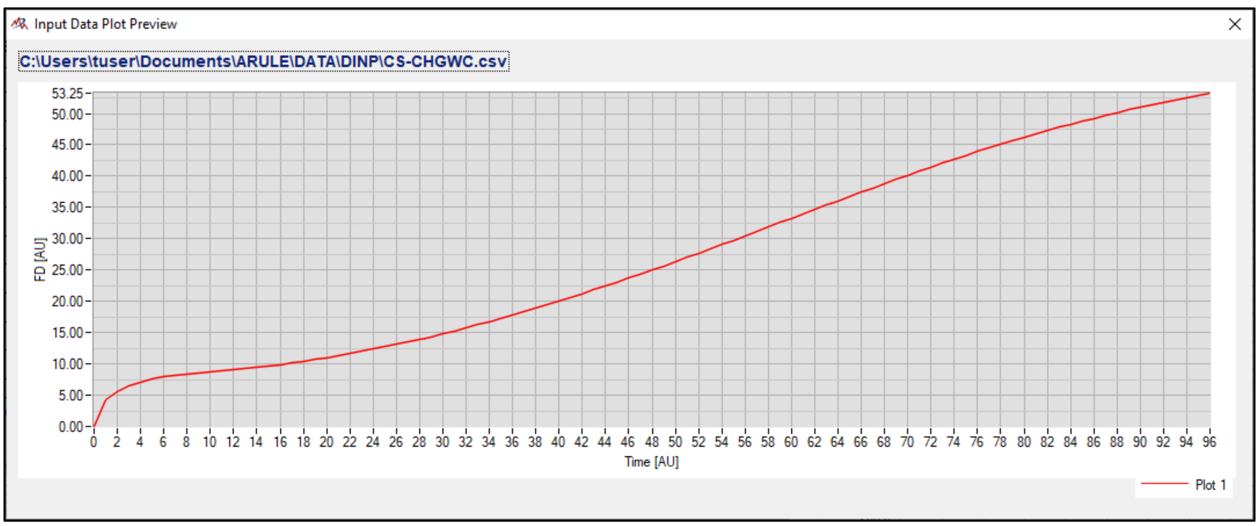


Figure 133 – XFD7 Input Data Plot Preview for Node 2

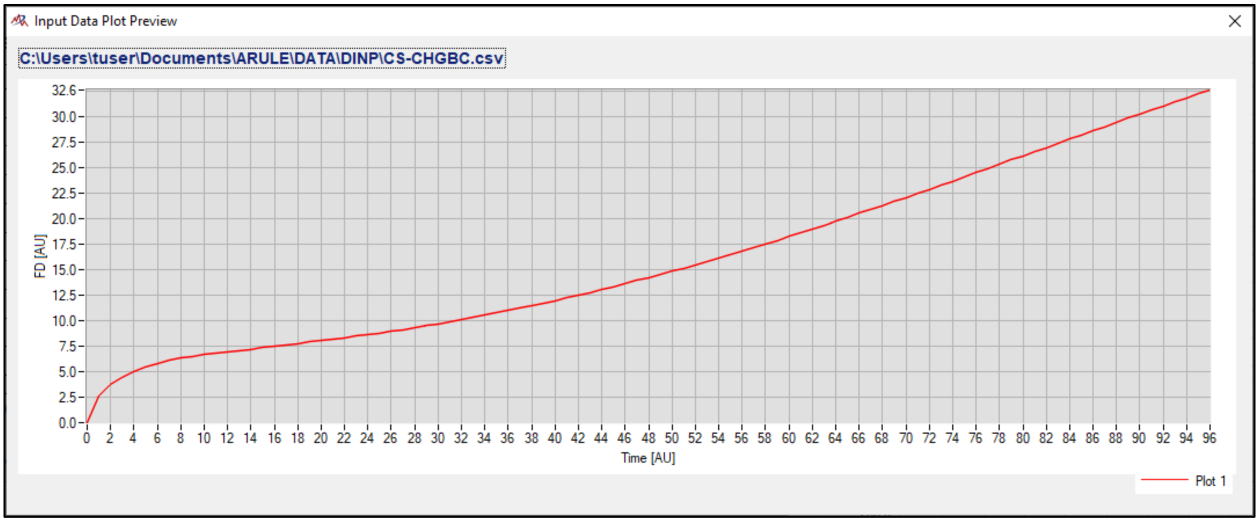


Figure 134 – XFD7 Input Data Plot Preview for Node 3

6.9.3 XFD7: Data Outputs and Prognostic Information

Figure 135, Figure 136, and Figure 137 show the Data Output for the two nodes in XFD3: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

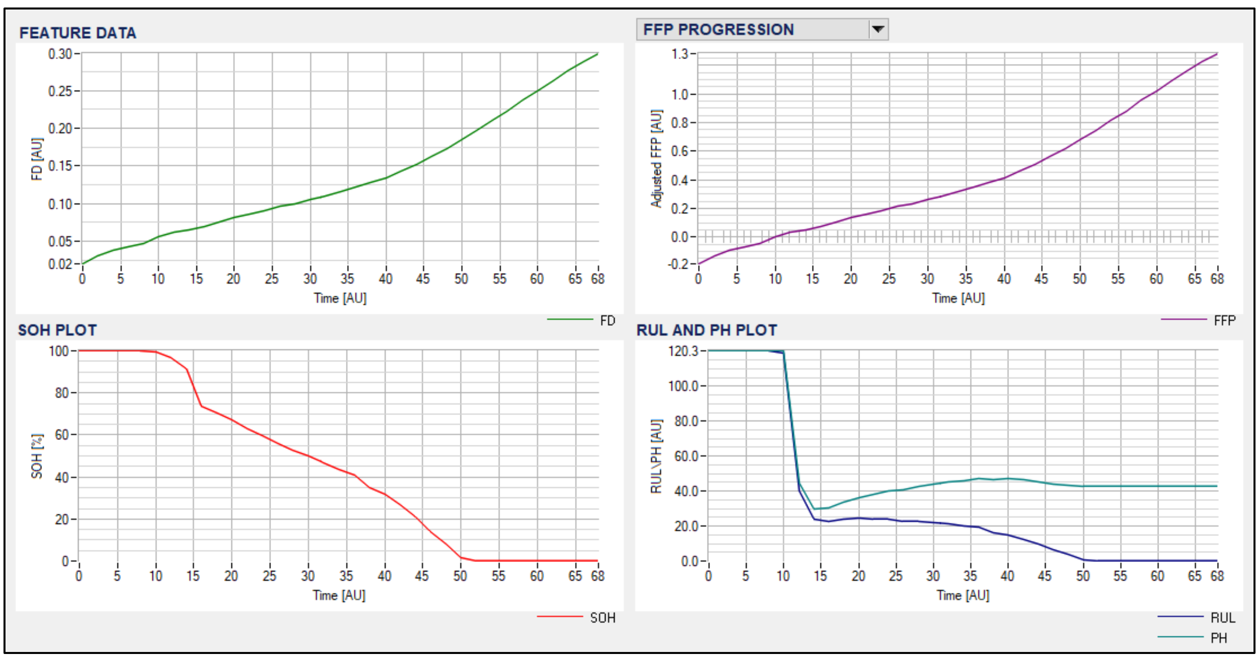


Figure 135 – Plots of the XFD7 Prognostic Information for the First Node

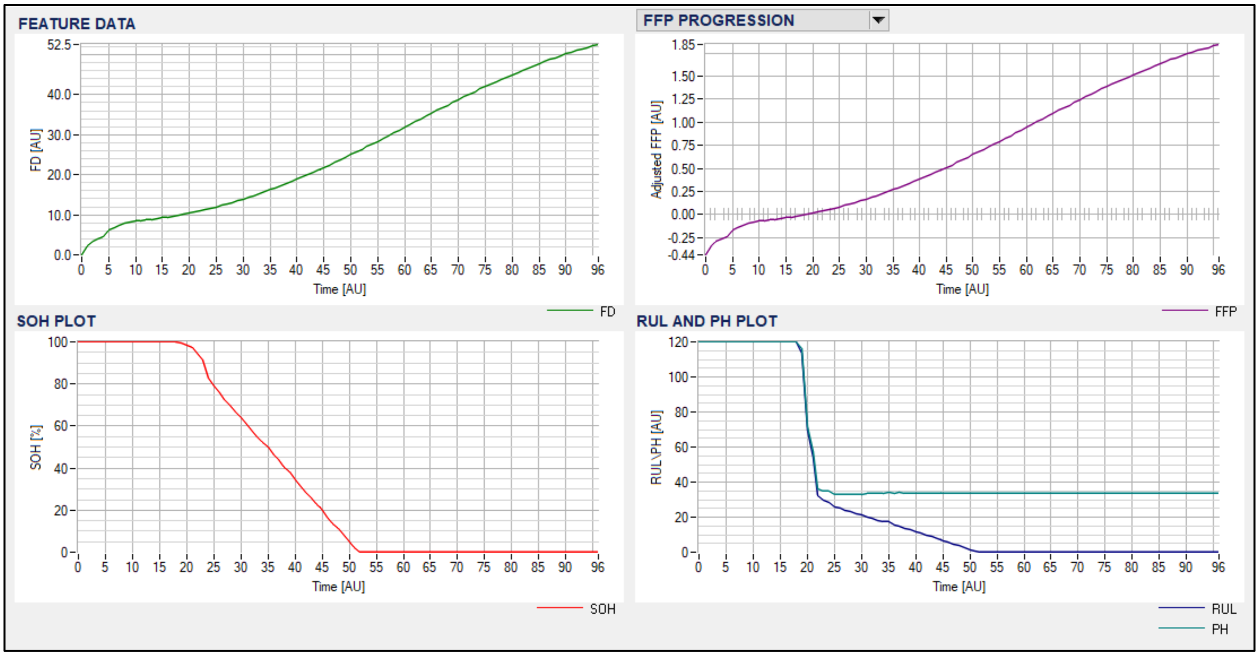


Figure 136 – Plots of the XFD7 Prognostic Information for the Second Node.

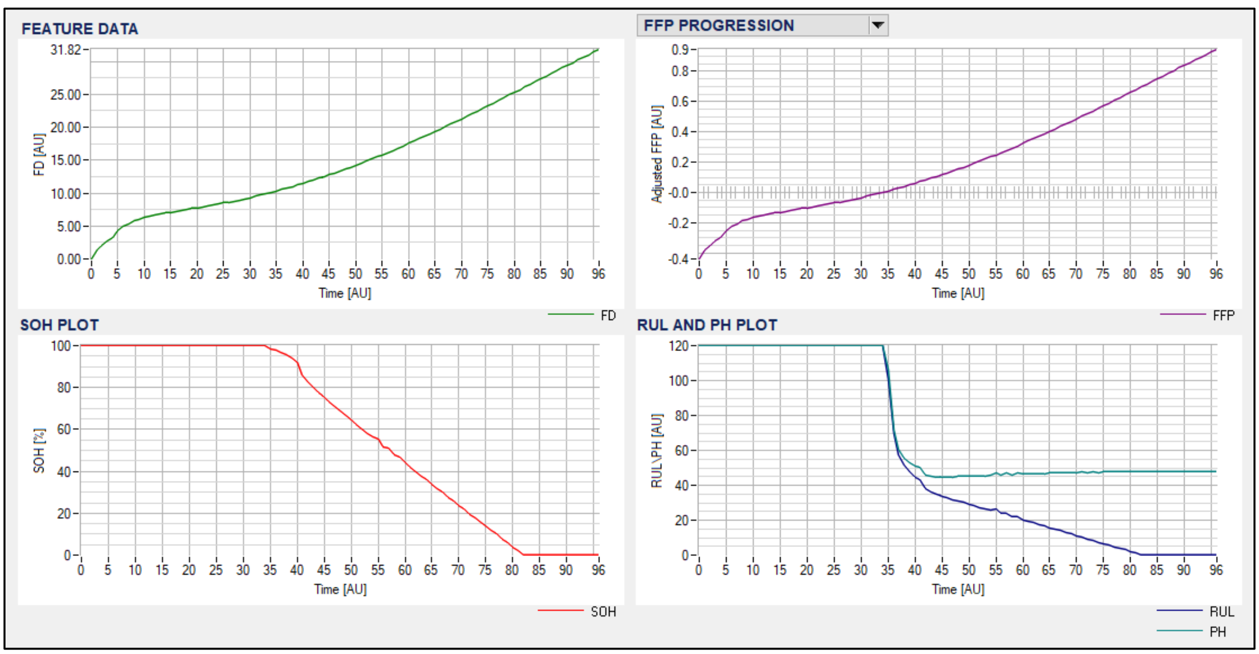


Figure 137 – Plots of the XFD7 Prognostic Information for the Third Node

6.9.4 XFD7: Discussion

Referring back to Figure 132, Figure 133, and Figure 134, the signature of the input data after the first inflection point resembles a square (power of n) function. Inspection of experimental data indicates the signature is somewhat flattened – indicating a power value closer to 1.5 rather than 2.0. Experimentation with parameters reveals that a wide range the square–root model values produce reasonable values of prognostic estimates: a final value of 1.25 is used to produce the very accurate estimates shown in Figure 135, Figure 136, and Figure 137. These types of prognostic solutions could also be implemented as an embedded process in a smart Battery Management System (BMS) design. This engineering service is offered through Ridgetop's CellSage product line, which also encompasses a suite of battery simulation and analysis software tools.

6.10 XFD8: High Data Rate Cables

High Data Rate (HDR) cables, such as those with antennas on submarines, are attached to an antenna mast assembly at one end while at the other end they are attached to connectors. The connectors are cabled to transceivers inside the submarine (Figure 138 and Figure 139).

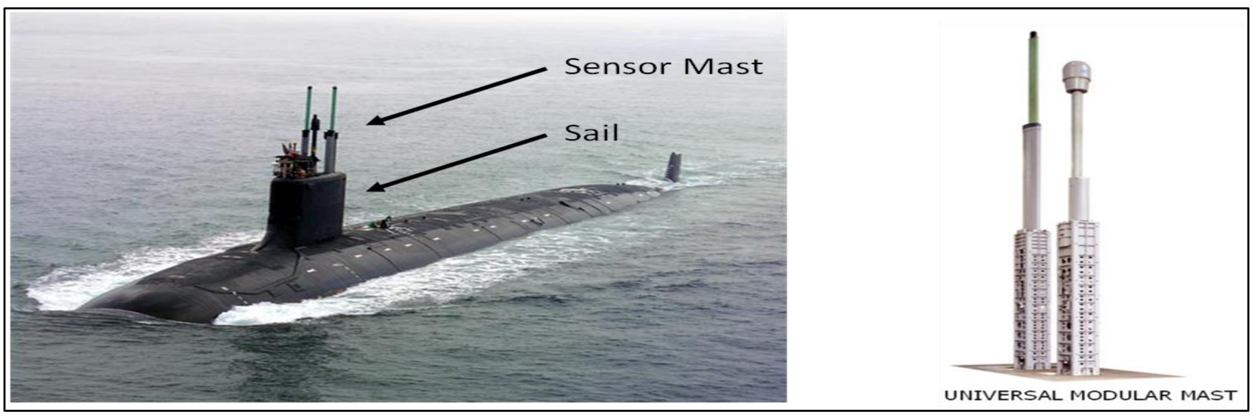


Figure 138 – Virginia class submarine with sensor mast attached to the sail

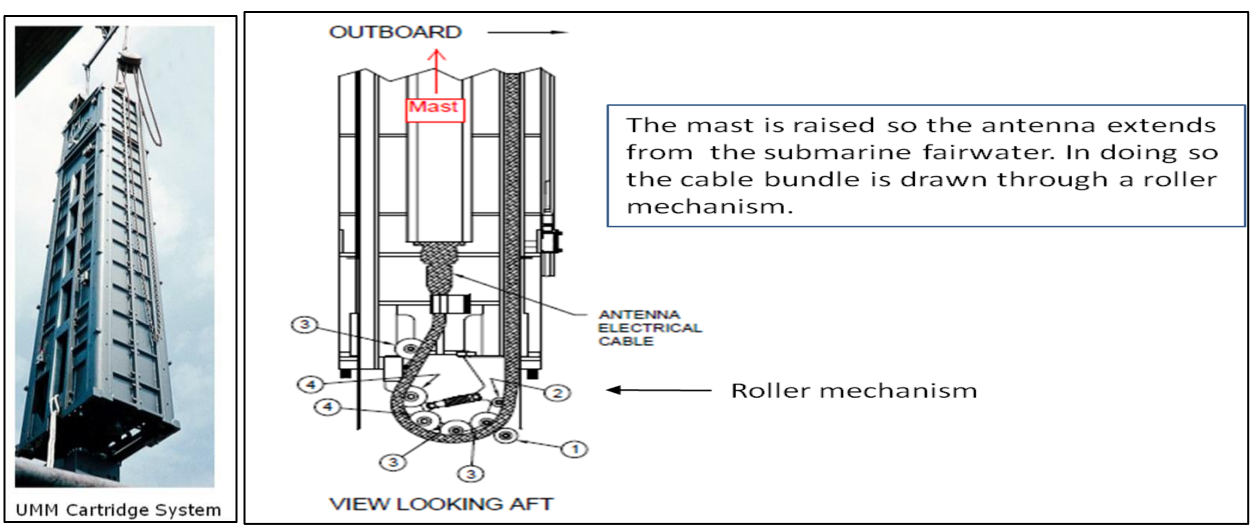


Figure 139 – Universal modular mast (UMM) cartridge and cabling system

Problem Description

As the antennas are raised and lowered, the HDR cables are reeled out and reeled in. During this process they can be subjected to crushing and crimping type of damage leading to loss of transmission and receiving power (Figure 140). Physical and traditional means of inspection HDR cables are problematic and impractical because the mast assembly is mounted inside the sail of a submarine.

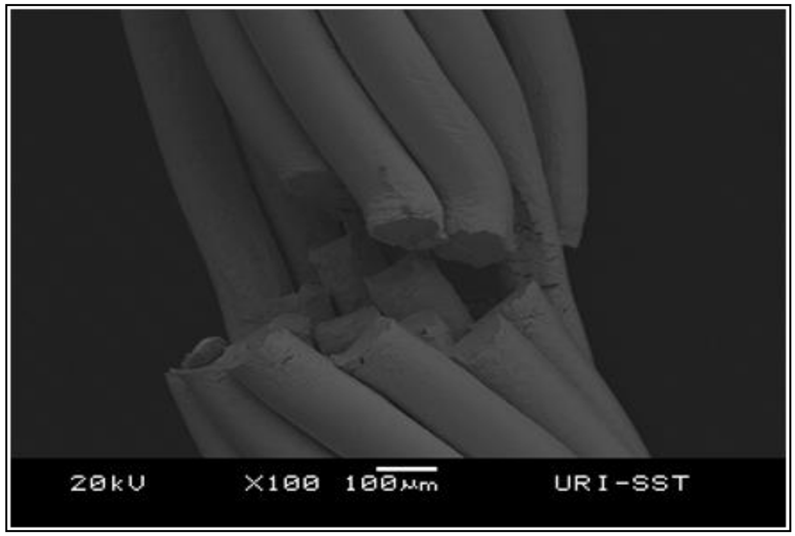


Figure 140 – Twisted triplet 24–AWG segment showing a kink; microscopic view of conductor failure

Solution Summary

Because high–frequency signals are transmitted and received by HDR cables, changes in distributed impedance can all also occur at different damage locations due to crimps, crushed damage dielectric material, distances, shielding, and wire conductors. Those impedance changes in transmission lines cause signal energy to be reflected and/or absorbed. A solution method to detect crushes and crimps is to do the following:

- Use a one–port network analysis method: Figure 141.
- Apply an RF signal and, capture its reflection, and process the signal with a smart processing unit (SPU) that produces CBD: Figure 142 and Figure 143.
- Send CBD output of the SPU to an ARULE™–based front–end application program as done in this example: Figure 144.

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

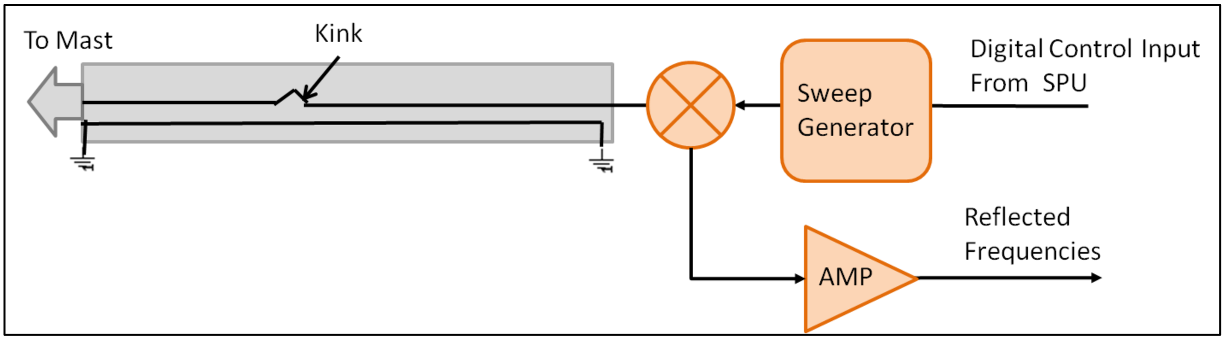


Figure 141 – SMRT Sweeping Probe block diagram

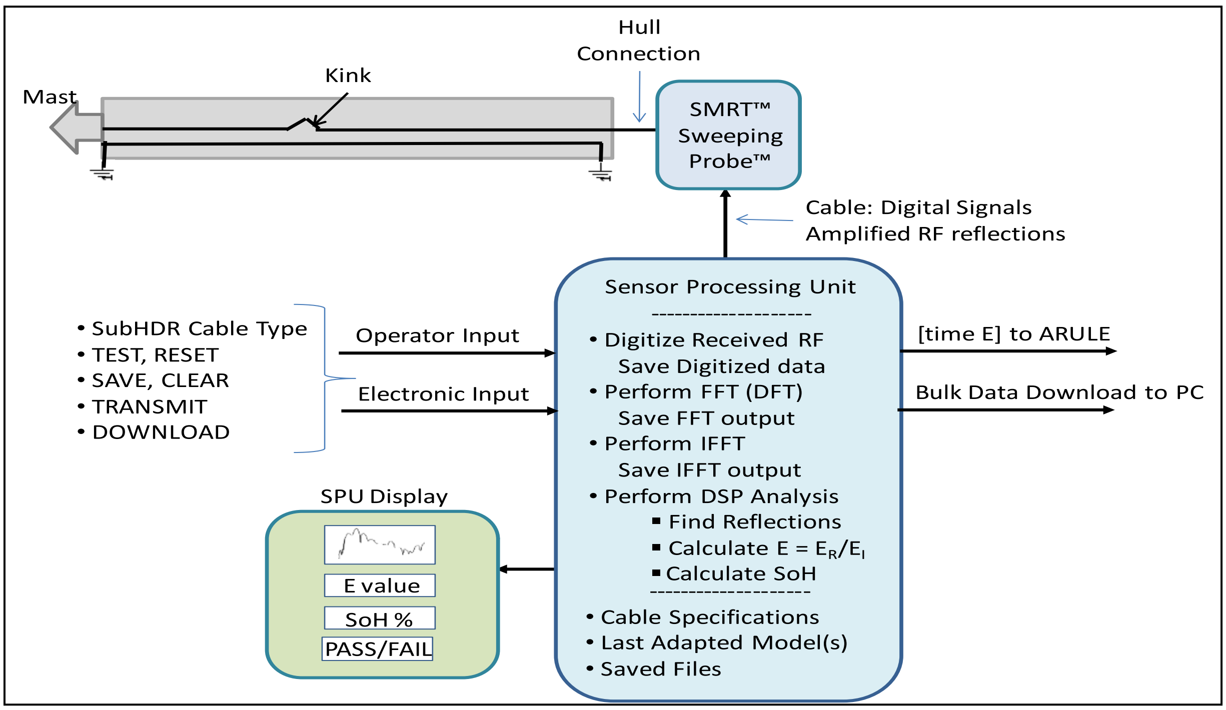


Figure 142 – SMRT Sweeping Probe and SPU.

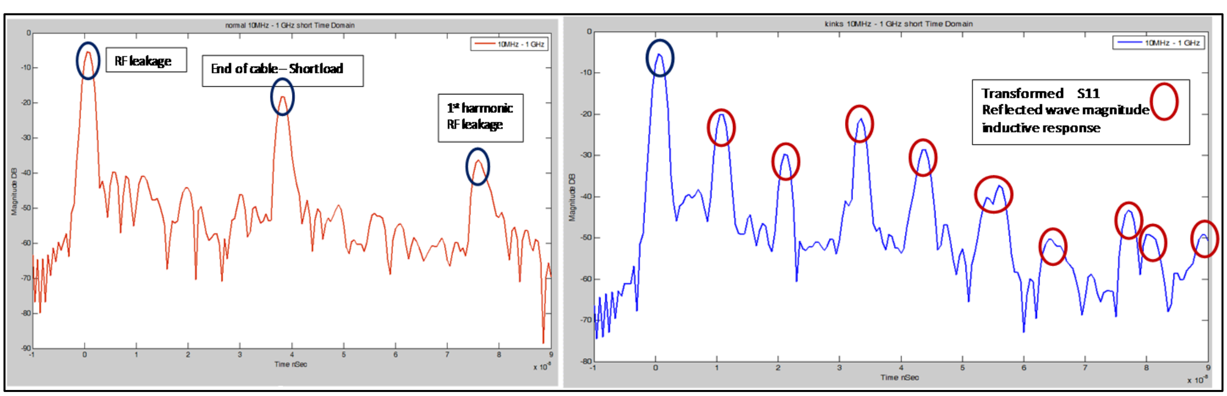


Figure 143 – CBD: Plots of the IFF transform – undamaged cable (left) and damaged cable (right)

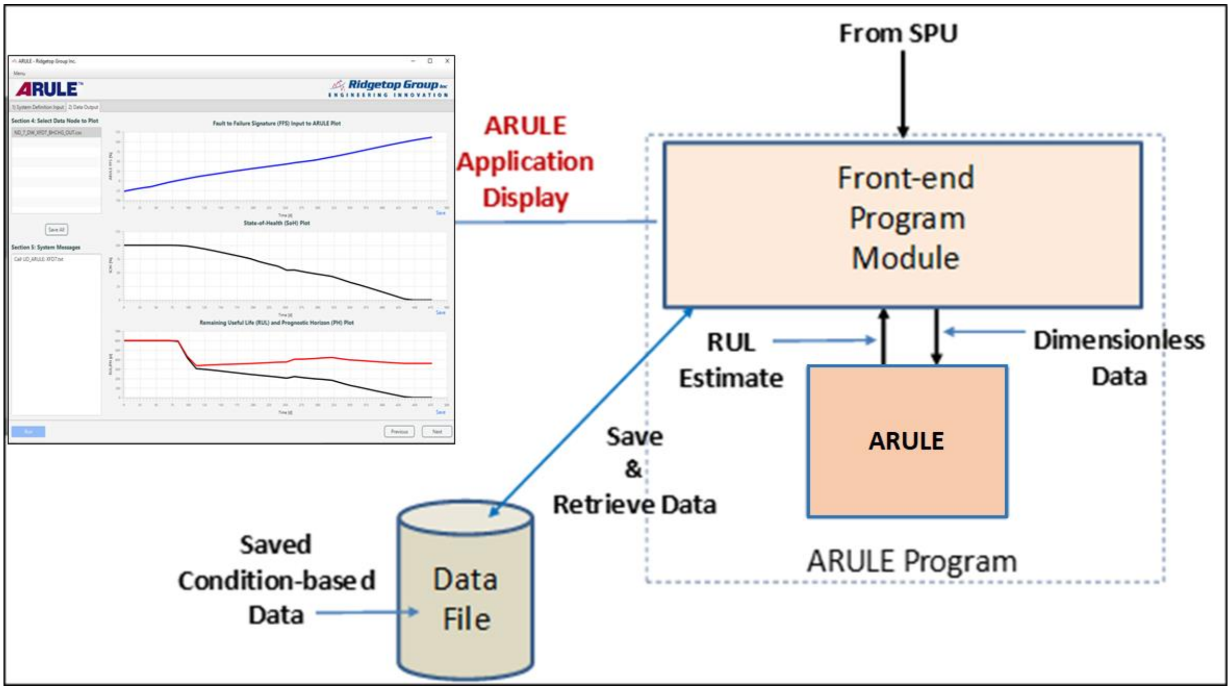


Figure 144 – ARULE™ Application

6.10.1 XFD8: Definitions

As shown in Figure 145, the SDEF file for this example points to 2 nodes with NDEF files named **XFD8_HDRA** and **XFD8_HDRA2**. Figure 146 and Figure 147 show the NDEF parameters and keywords used for XFD8.

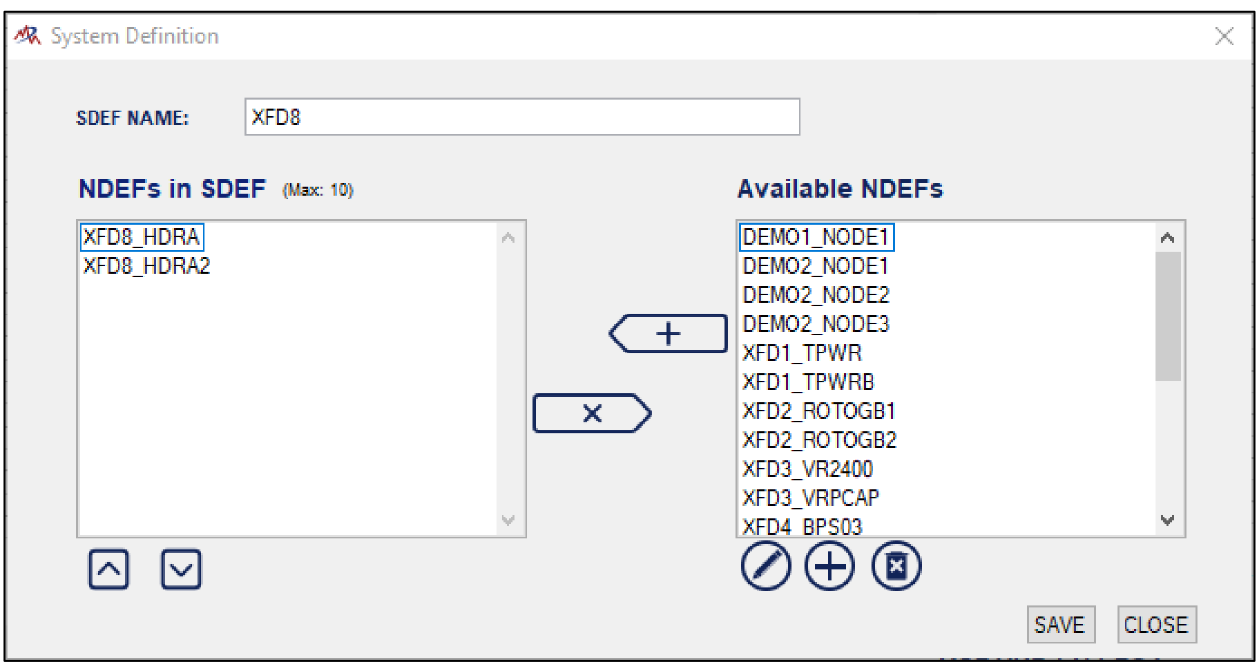


Figure 145 – XFD8 System Definition (SDEF) User Form

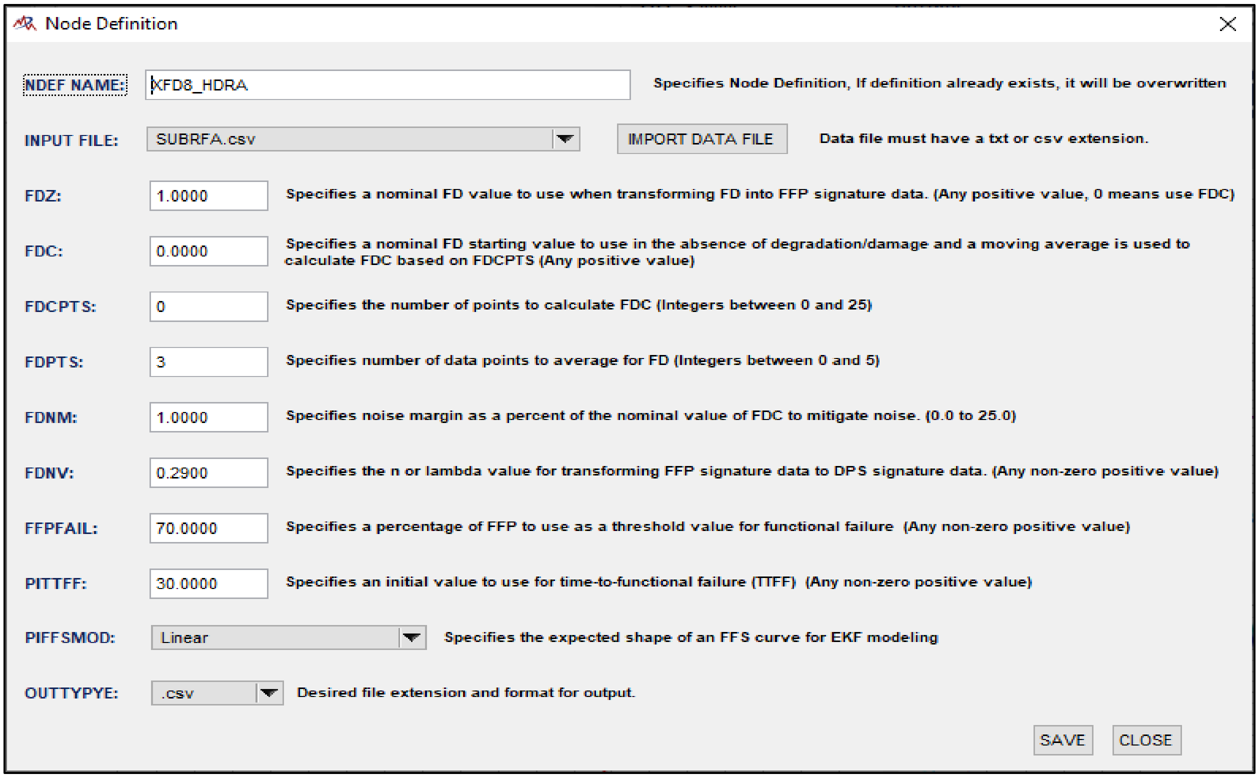


Figure 146 – Node Definition (NDEF) User Form for First Node in XFD8

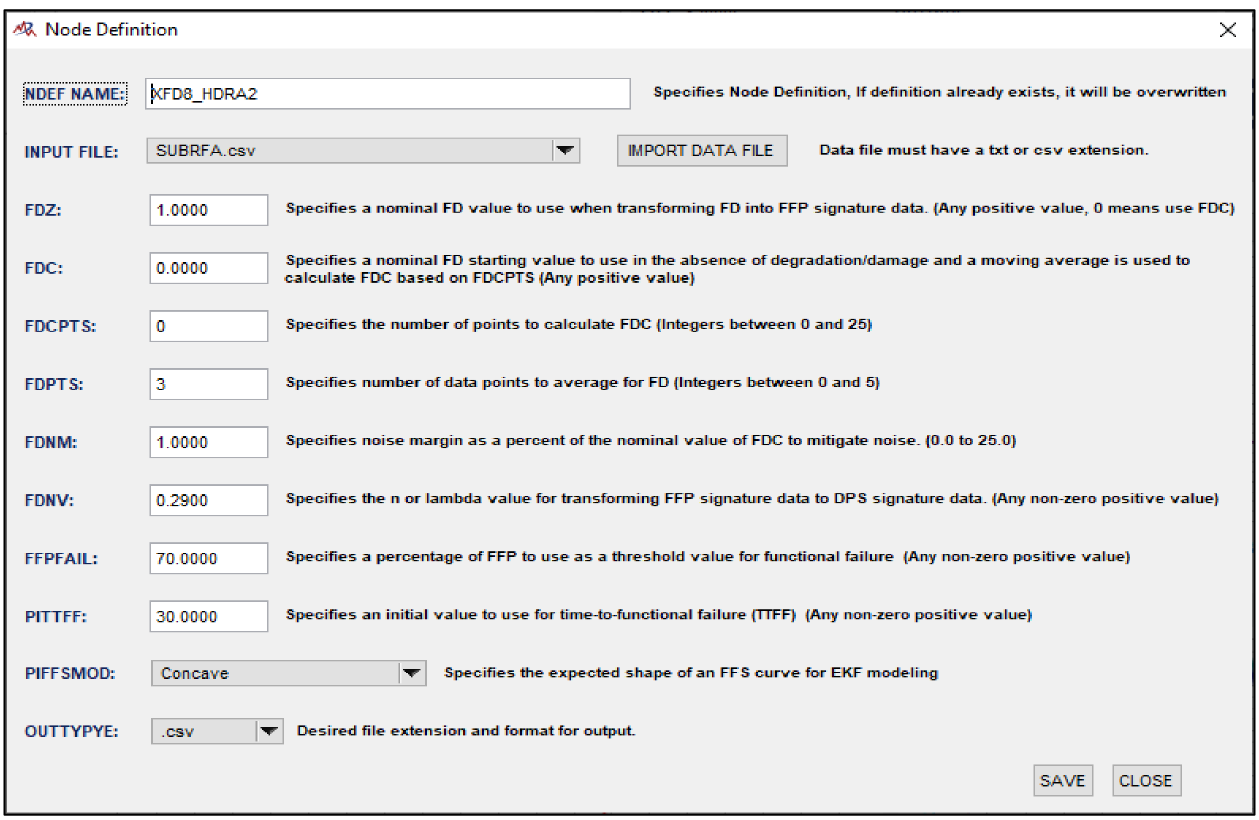


Figure 147 – Node Definition (NDEF) User Form for Second Node in XFD8

6.10.2 XFD8: Input Feature Data

Figure 148 shows the Condition–Based Data Input plot for **XFD8_HDRA** and **XFD8_HDRA2**.

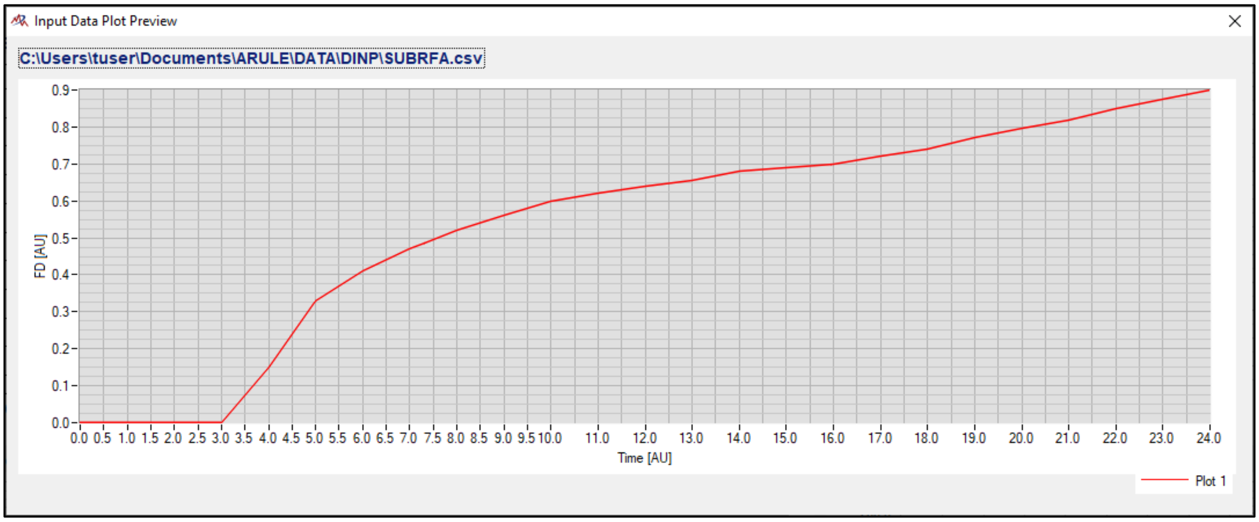


Figure 148 – XFD8 Input Data Plot Preview for Node 1 and Node 2

6.10.3 XFD8: Data Outputs and Prognostic Information

Figure 149 and Figure 150 show the Data Output for the two nodes in XFD3: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

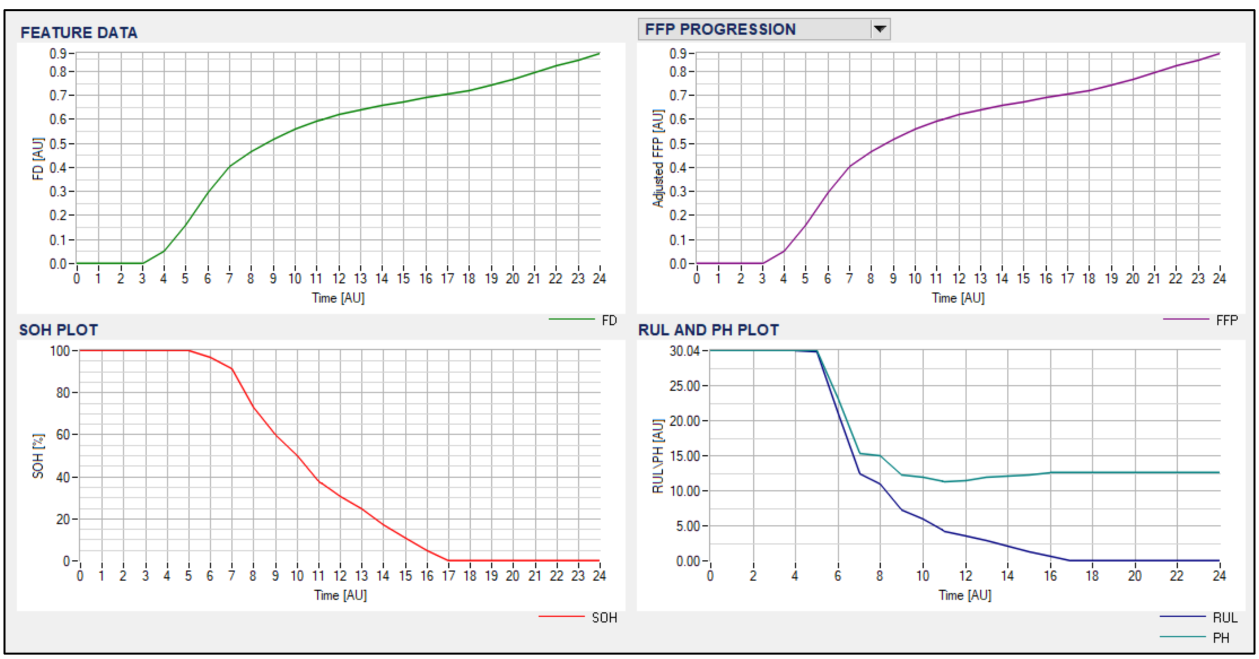


Figure 149 – Plots of the XFD8 Prognostic Information for the First Node

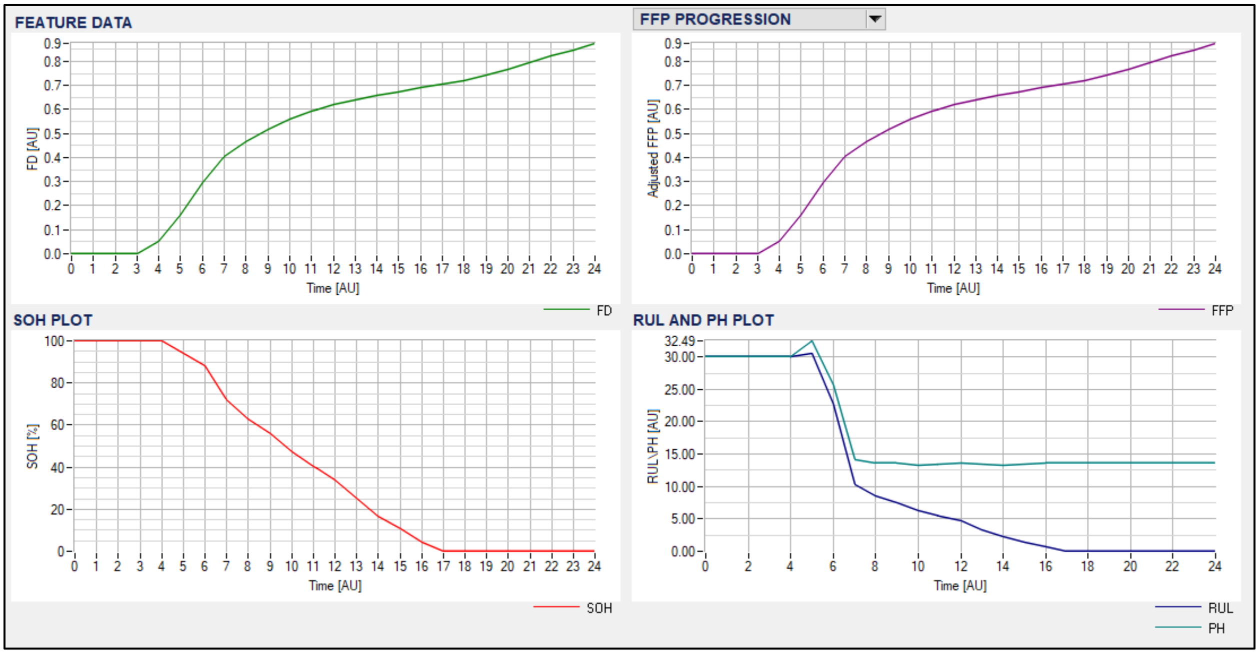


Figure 150 – Plots of the XFD8 Prognostic Information for the Second Node

6.10.4 XFD8: Discussion

Referring back to Figure 143, analysis of the reflected energy (left–hand plot) reveals the following:

1. In the absence of degradation in the form of damage to the wire (grounding shield, insulation material, or any of the twisted–wire conductors), three instances of high reflection occur: when the high–frequency energy is injected into the antenna, when the injected energy reaches the end of the wire; and when the energy is reflected back to the beginning of the antenna.
2. At every location of damage to the antenna, there will be change in the effective impedance because the distributed capacitance, inductance, and resistivity is changed,
3. When the magnitude of the reflected energy exceeds a predetermined level above the level for an undamaged state plus noise, the antenna is evaluated as being degraded.
4. As the level of damage increases, the magnitude of the reflected energy increases.
5. The change in magnitude at that location is a FD having a signature as shown in Figure 149 and Figure 150.
6. A "weakest–link evaluation" can be used:
    - The SoH of the antenna is the lowest–valued SoH of all locations of damage.
    - The RUL of the antenna is the lowest–valued RUL of all locations of damage.

6.11 XFD9: Electro–mechanical Actuators (EMAs)

This example provides a detailed overview for how Ridgetop has used ARULE™ to prognostic enable an electro–mechanical actuator system with multiple nodes. For a NASA–funded program, Physical Modeling and Diagnostic and Prognostic Reasoning (MAPR) program, Ridgetop Group designed and developed a number of solutions related to EMA devices that are widely used in different types of equipment, vehicles, and structures, to position and rotate surfaces. An EMA comprises devices, assemblies, and units: such as one or more power sources, positioning controllers, power converters, motors, and shafts. The Hardware Test Bed in Figure 151 comprises two line–replaceable units (LRUs): a switch–mode power supply (SMPS) with a sensor and an EMA motor with a phase–current sensor: the EMA motor comprises a brushless DC motor (BLDC) with a shaft to raise and lower a surface (Figure 151 and Figure 152). Likewise, the software in a host computer comprises two blocks (Figure 153 and Figure 154).

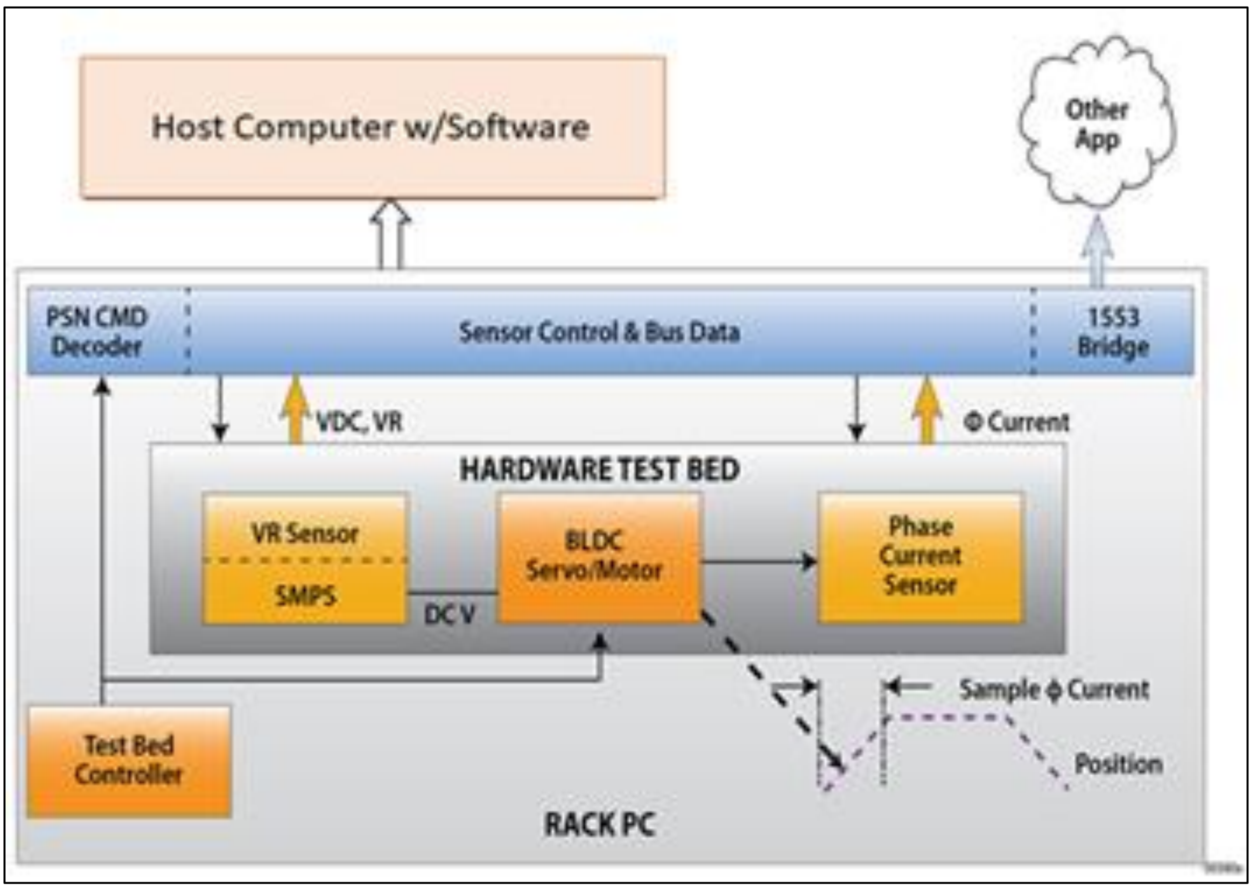


Figure 151 – EMA (controller and motor) test bed

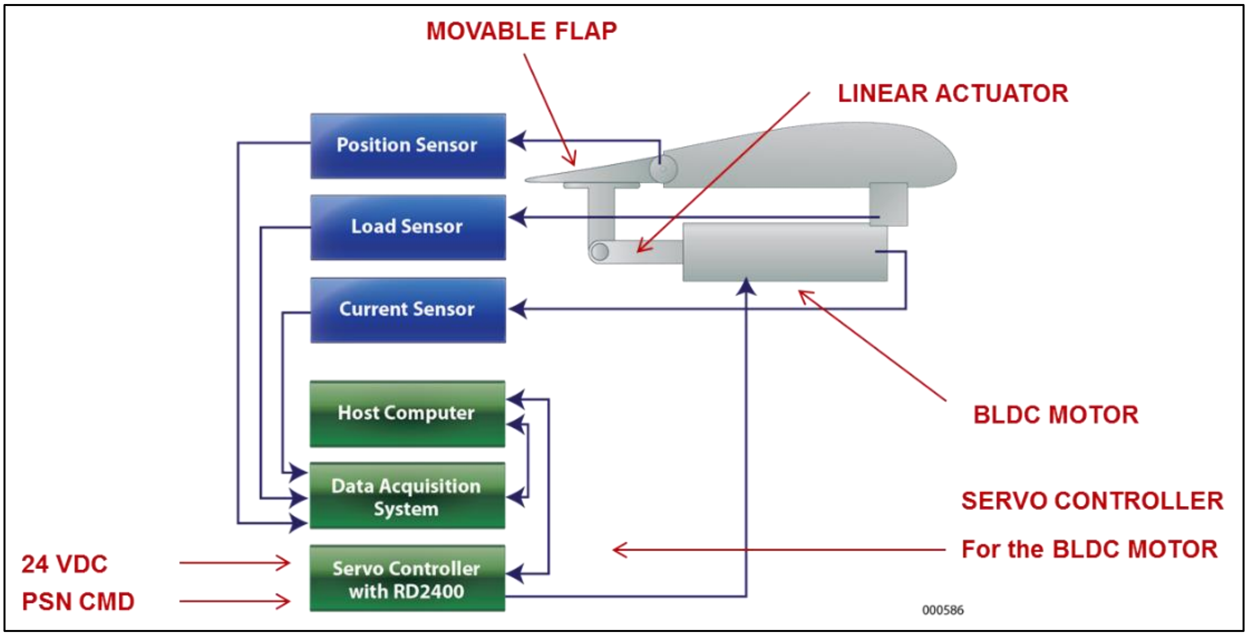


Figure 152 – Test Bed diagram

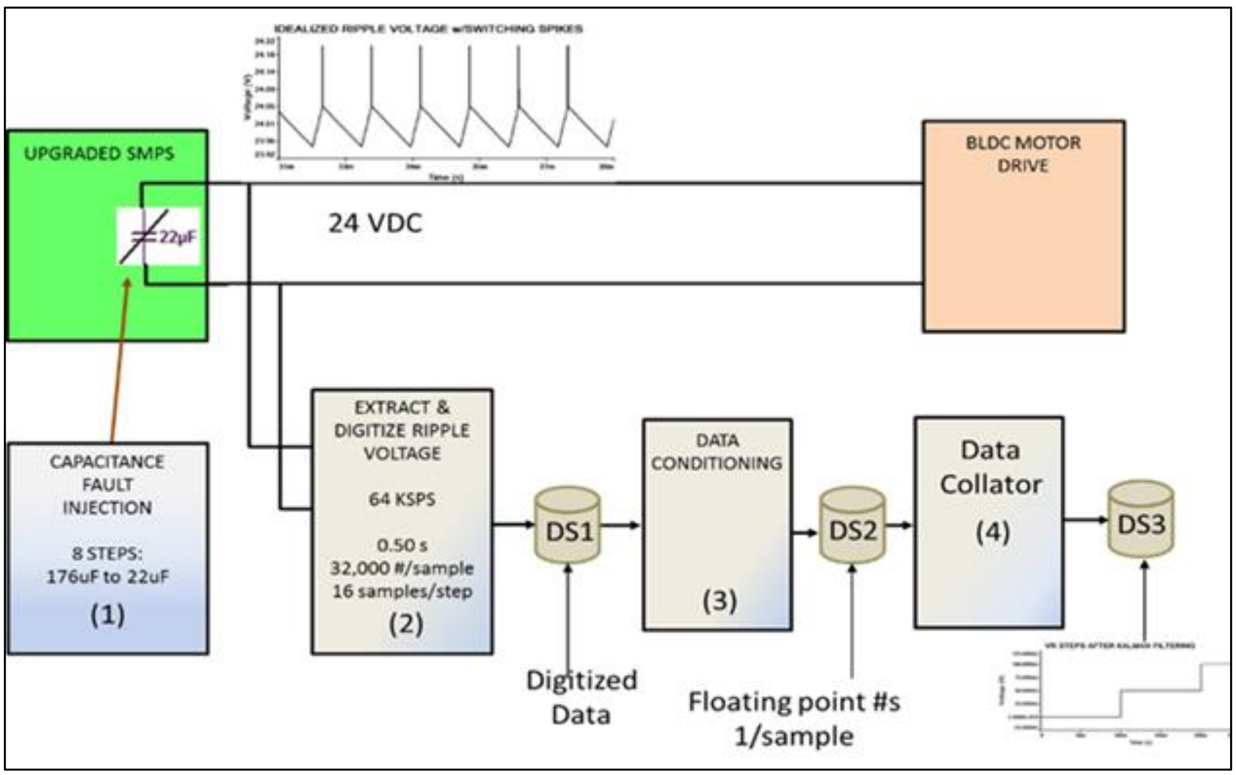


Figure 153 – Collect and process data from a switch–mode power supply: RD2400 in Figure 154

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

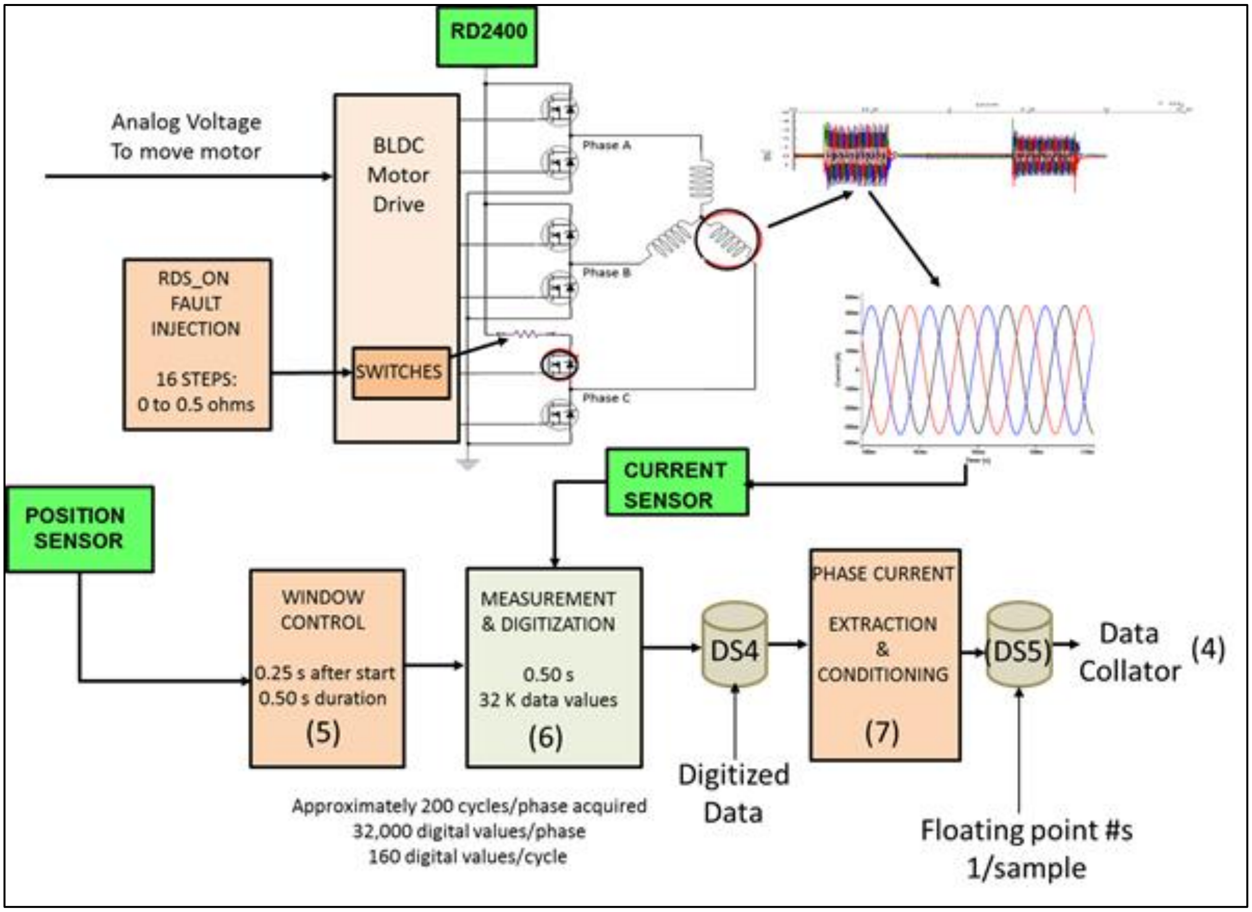


Figure 154 – Collect and process data from the EMA

The first LRU, a Ridgetop–designed and developed 5–V SMPS, had a smart sensor and a fault–injection controller that produced FD signature. Ridgetop–developed sampling, data management, and IEEE 1553 bus management collected that FD signature in a data file (DS3 in Figure 155). Figure 156 is a plot of the collected FD.

The second LRU, Ridgetop–designed and developed an EMA test bed that had four sensors (a positioning sensor and three AC phase–current sensors). The EMA test bed also had fault–injection hardware, firmware, and a gated–window software application to control data sampling. The positioning sensor is used with gating logic to create windows during which phase currents are sampled (see Figure 157).

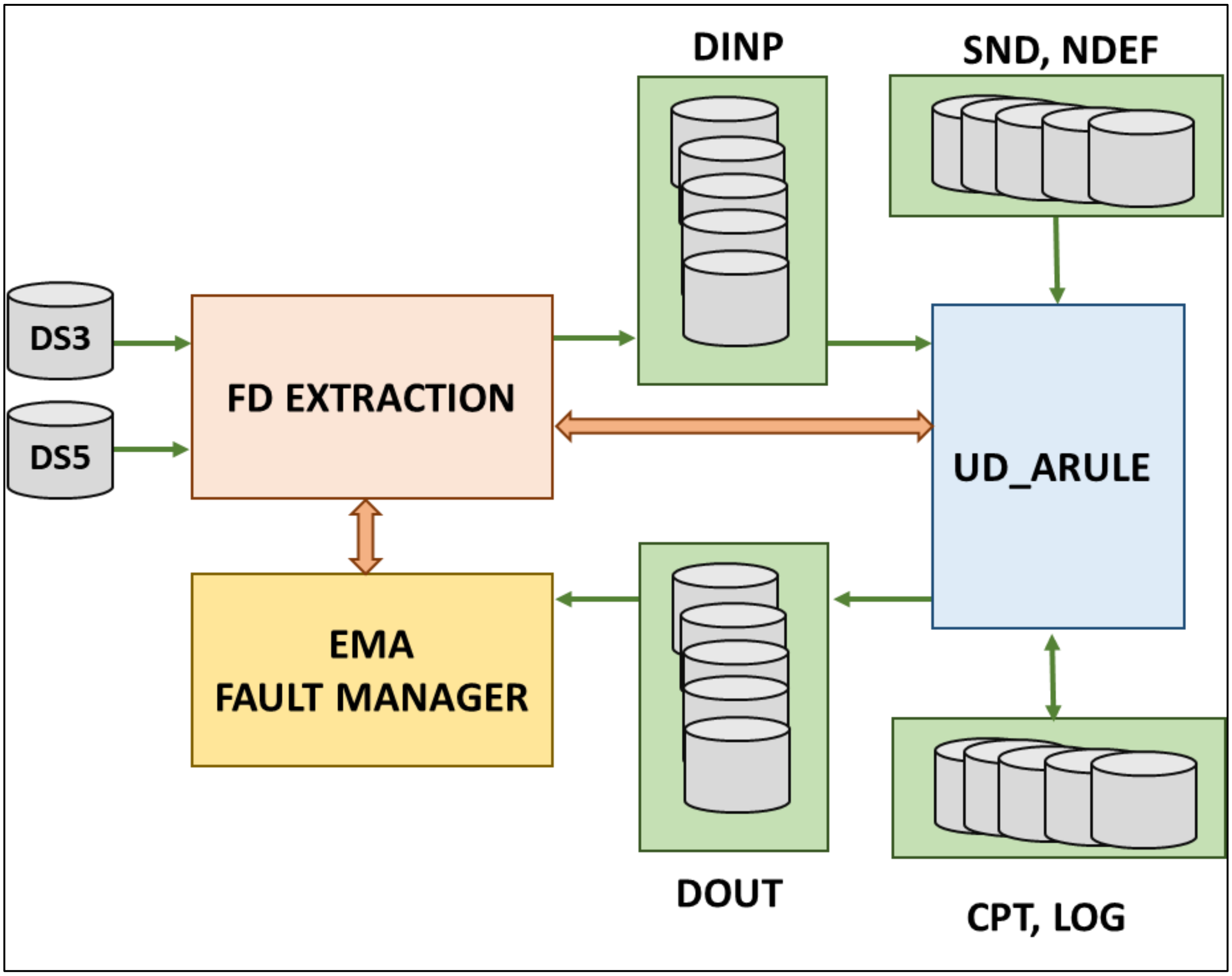


Figure 155 – Diagram: experimental solution

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

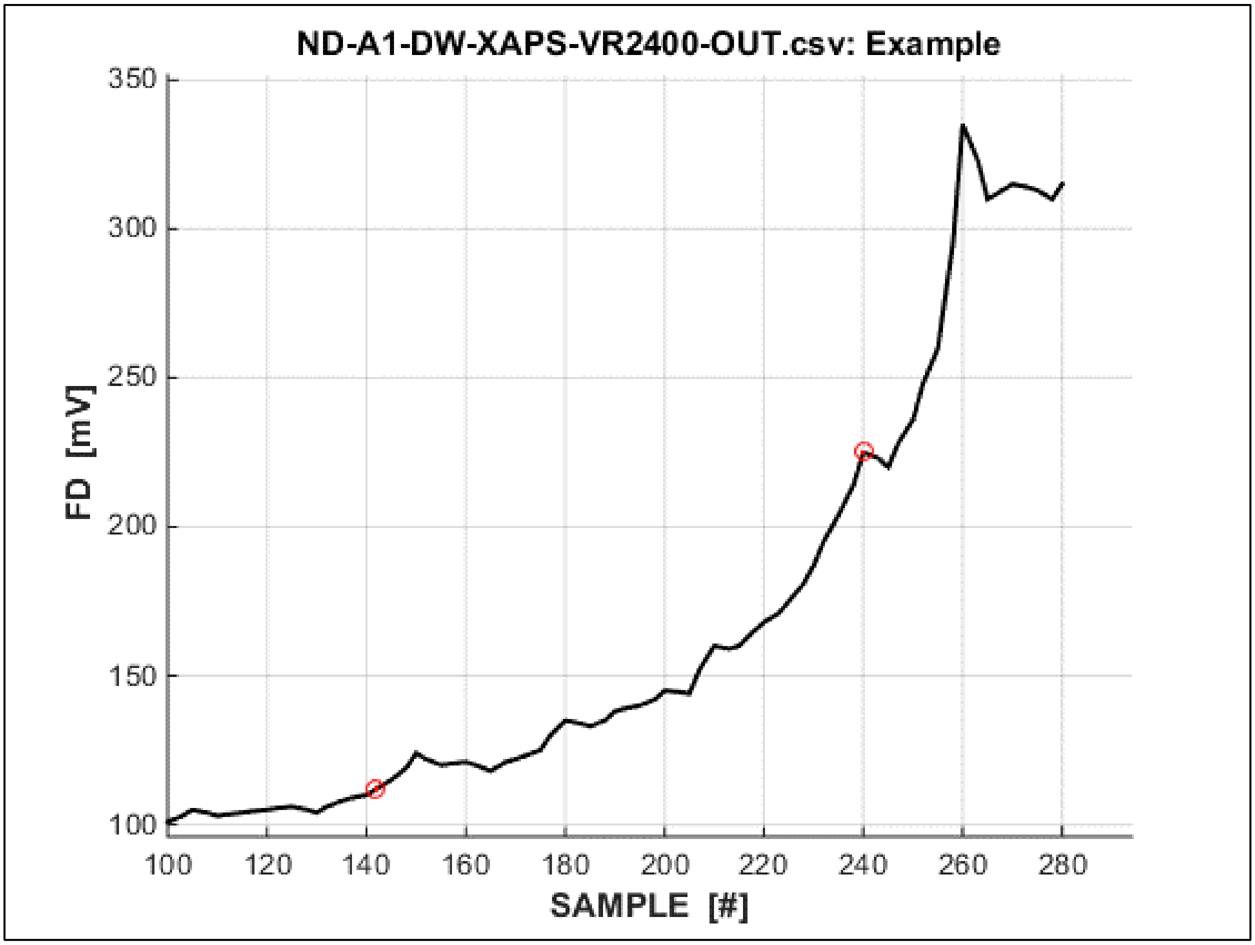


Figure 156 – Extracted FD from DS3: ripple voltage

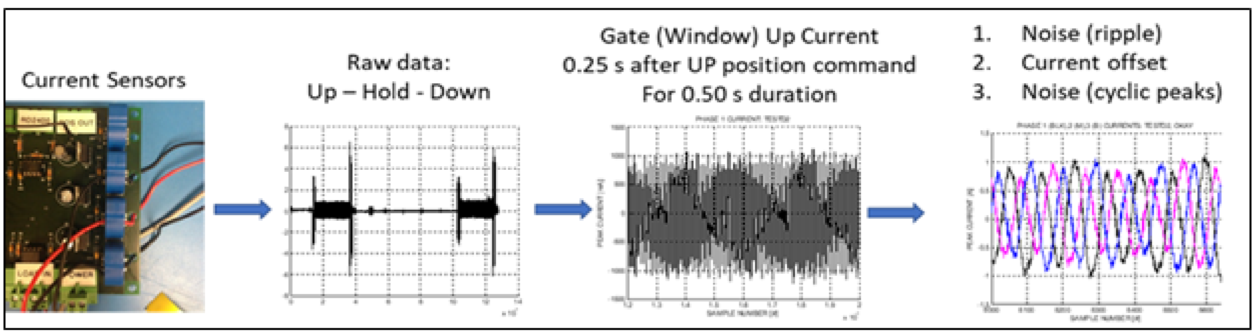


Figure 157 – Phase–current sampling: sensors, sensor output, sampled data, and examples of noise

The FD from sampling the phase currents exhibit all manner of noise: electrical (Figure 158), amplitude variability (Figure 159), current offset (Figure 160, unbalanced positive and negative current halves), and distortion (Figure 160): all of which required noise–mitigation and noise–canceling solutions).

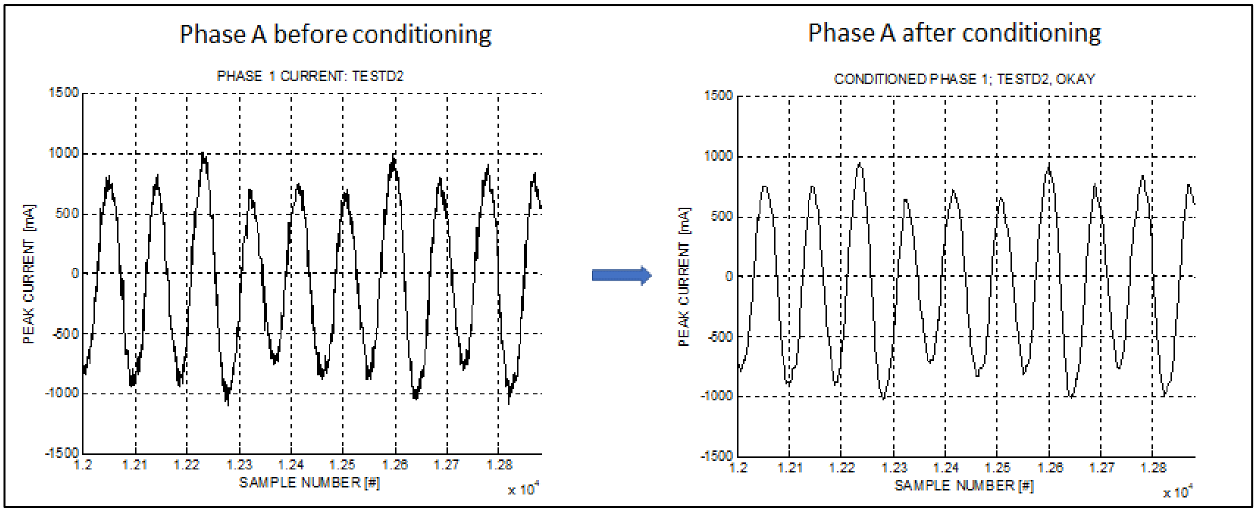


Figure 158 – Phase–current noise: electrical before (left) and after (right) conditioning

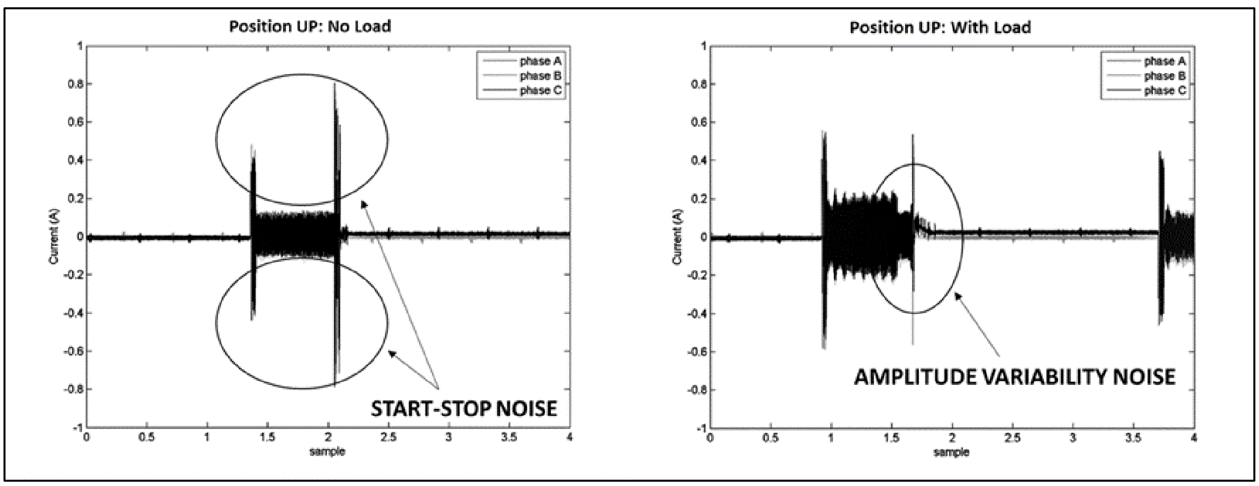


Figure 159 – Phase–current noise: start–stop (left) and load–variation (right)

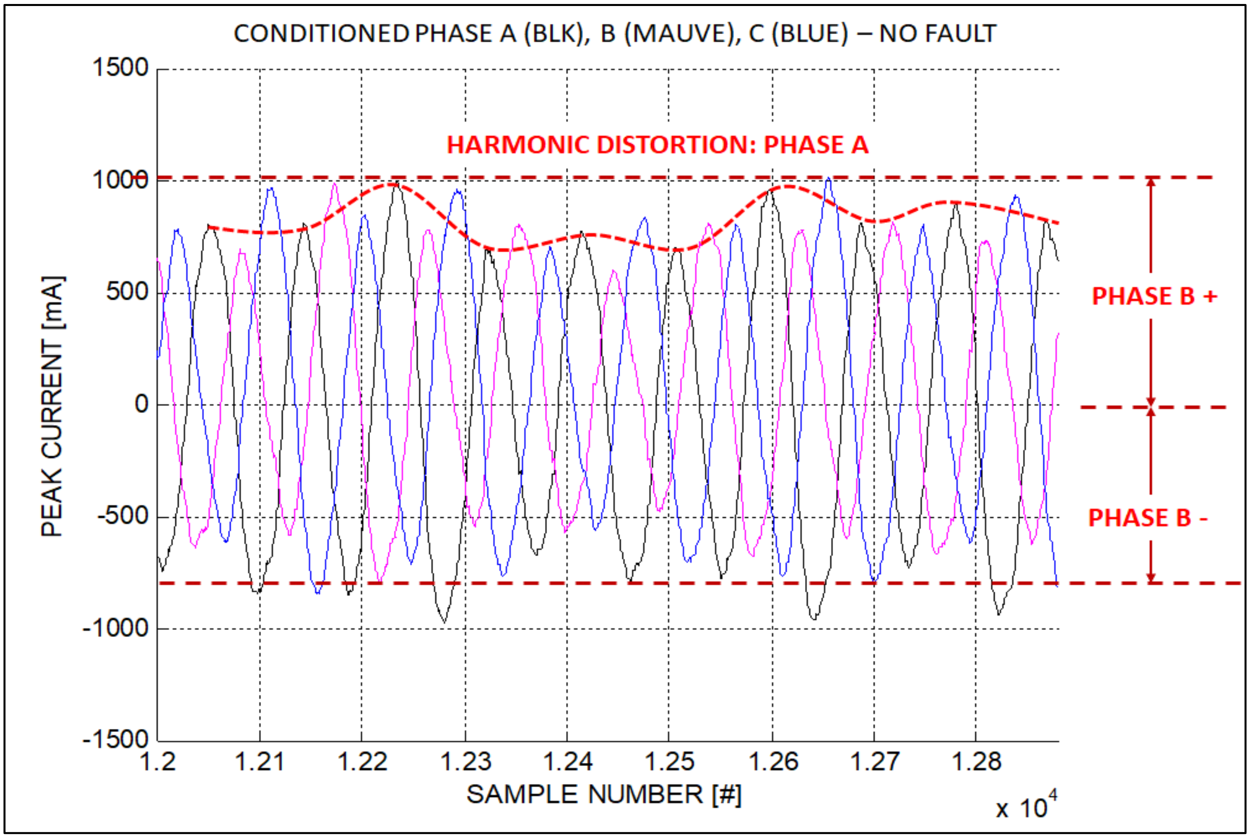


Figure 160 – Current offset and distortion

Experimentation, physics–of–operation and failure analyses, and physics–of–failure analyses revealed the following important characteristics of the EMA used in the test bed:

- The EMA motor uses a "floating" reference for the stator windings (Figure 161) so that, for example, an increase in impedance in one winding path causes all of the currents in all of the paths to change: zero sum of all of the currents into the floating node.
- The 120–degree phase shift is a primary cause of distortion: N reduction in phase–A current requires 2/3 N and 1/3 N increase in phase–B and phase–C currents.

Special methods for calculating rms and differential methods of comparison were required:

- To distinguish between failing nodes: switching–transistor (1 of 6), stator winding (1 of 3) and motor shaft (1 of 1).
- To support 10 degrees of freedom with only three sensors.
- To provide quantifiable computational values to support prognostics.

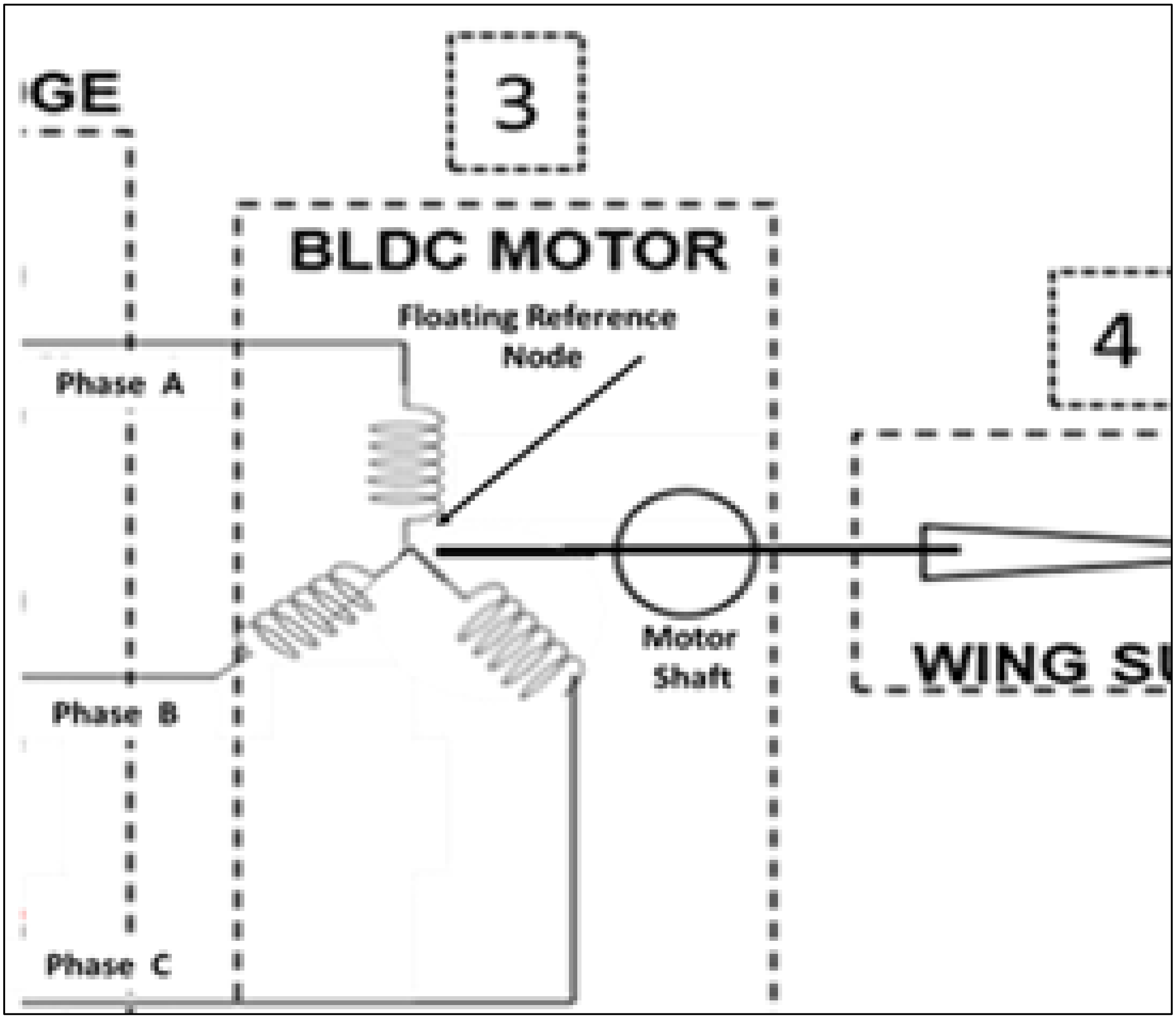


Figure 161 – Brushless DC motor with floating reference for the stator windings

The special methods for processing phase currents results in 10 sets of extracted FD signatures (refer back to Figure 155) as plotted in Figure 162, Figure 163, and Figure 164.

In Figure 162, the positive and negative halves of the phase–B current and phase–C current diverge in opposite directions while they diverge in the same (negative) direction for phase–A. All three of the phase currents diverging is indicative of degradation in a switching–transistor path: in this case it is the negative path for phase A in the H–bridge controller.

When there is (1) no divergence of the negative and positive halves of the phase currents, the fault tree is the following:

- There is no significant change in any of the three phase currents: no degradation.

- There is a significant change in one of the phase currents: stator–winding fault (Figure 163).
- There is significant change in all three phase currents: motor–shaft fault (Figure 164).

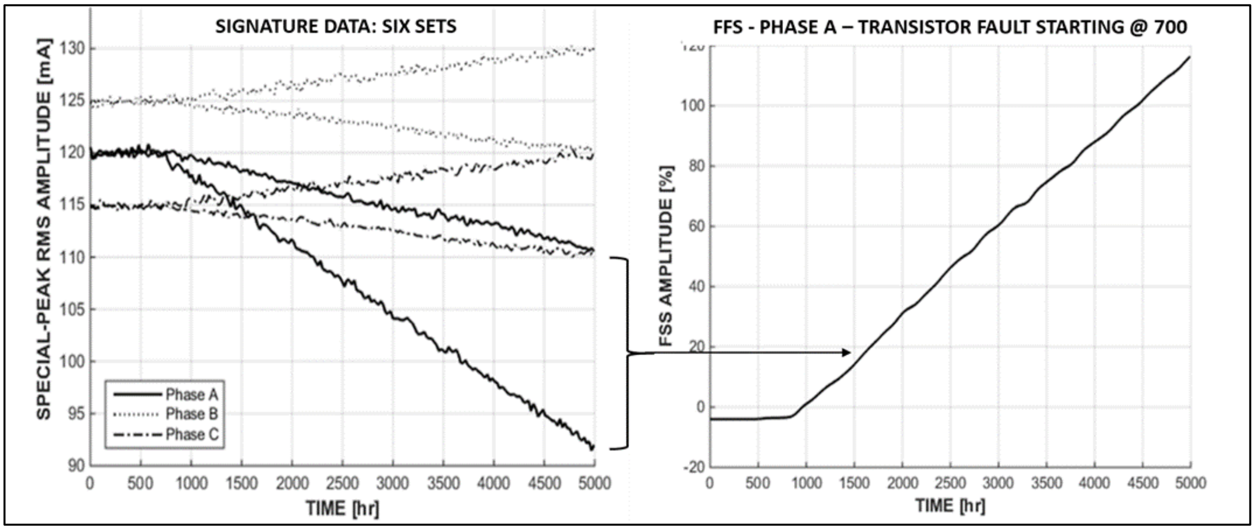


Figure 162 – Degradation in one–of–six transistor paths (phase A, negative path)

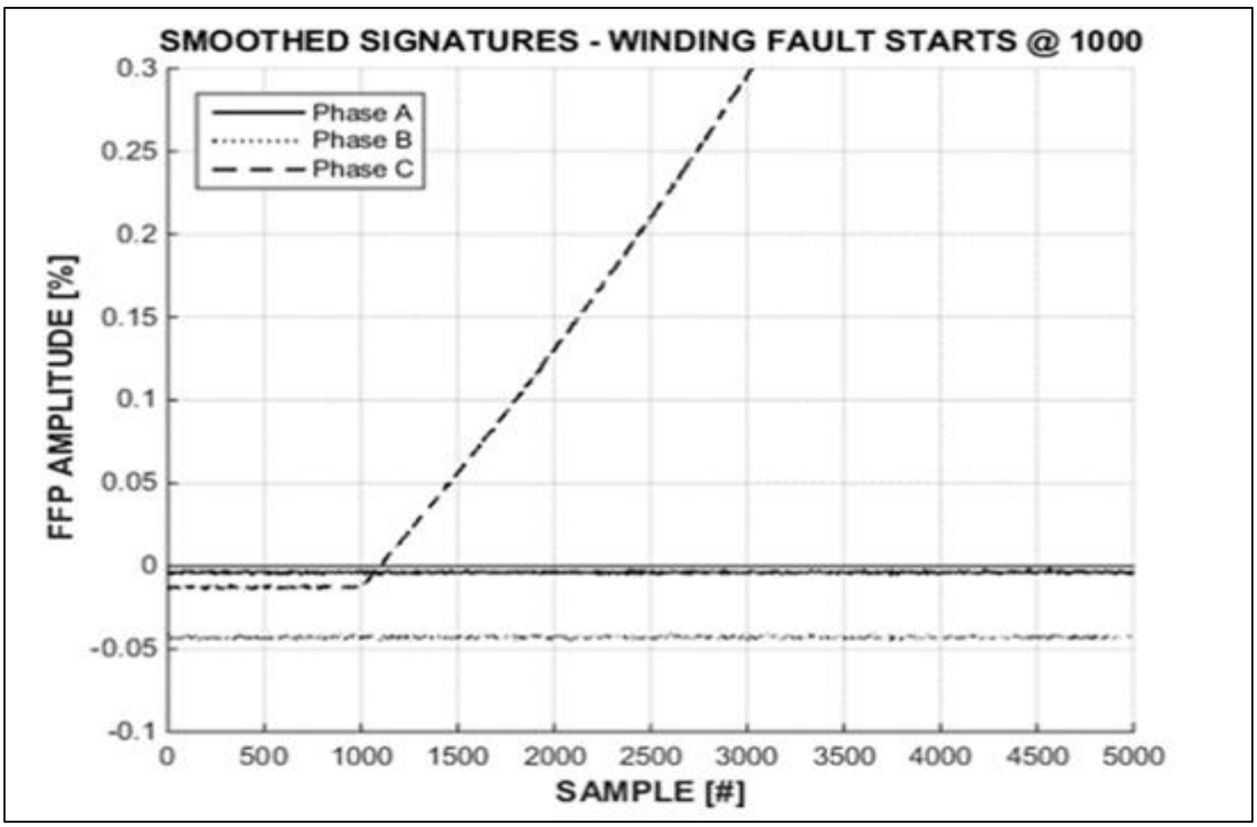


Figure 163 – Degradation in one–of–three stator windings

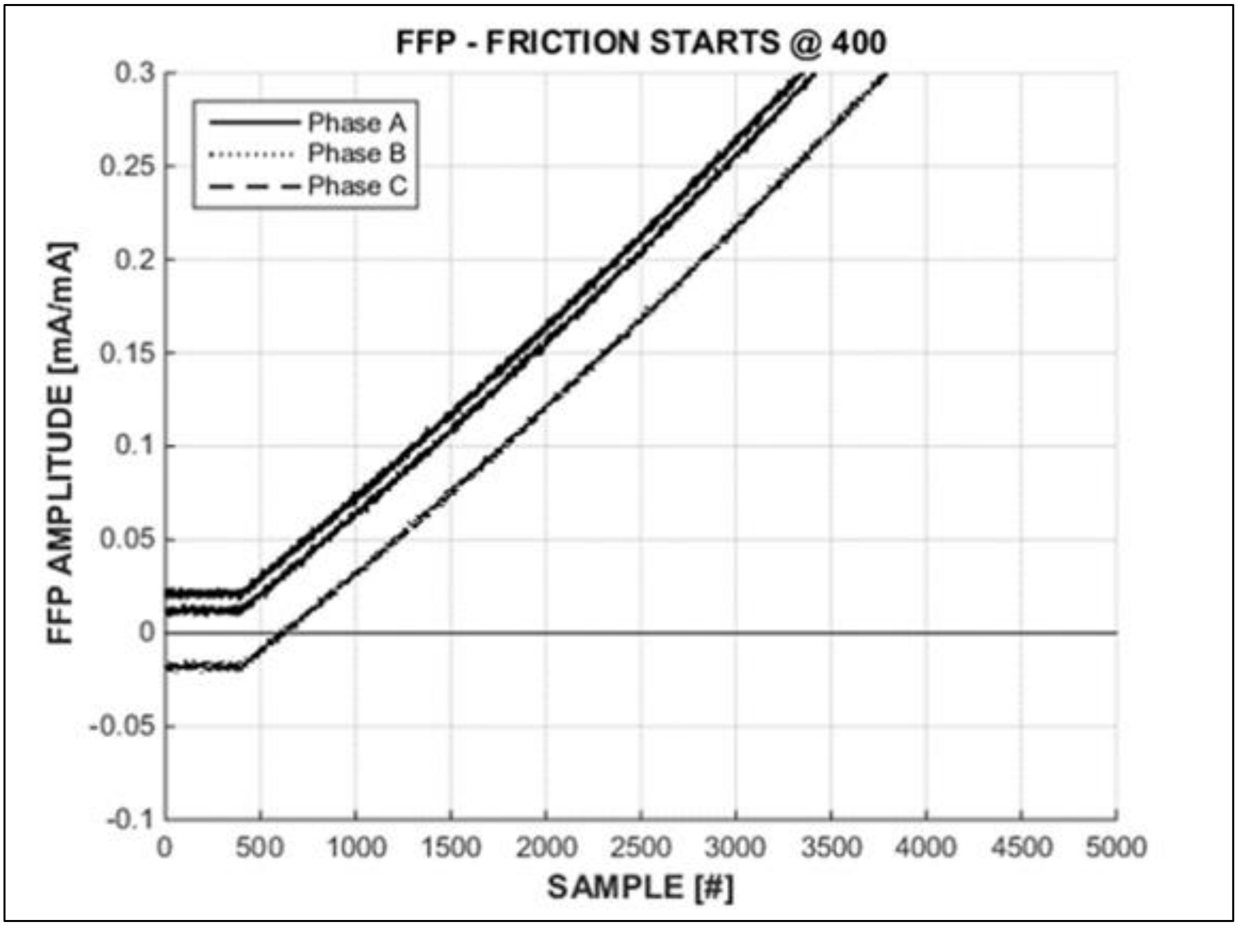


Figure 164 – Excessive friction/load on EMA rotating shaft

6.11.1 XFD9: Definitions

For illustrative and demonstration purposes, four examples of NDEFs for an EMA are created: XAPS for the SMPS, XAST for a switching transistor, XASW for a stator winding, and XAMS for a motor shaft: see Figure 165.

The input data plotted in Figure 162, Figure 163, and Figure 164 are simulated based on experimental results. The simulation results and experimental results were combined to create the example data used as input for this example.

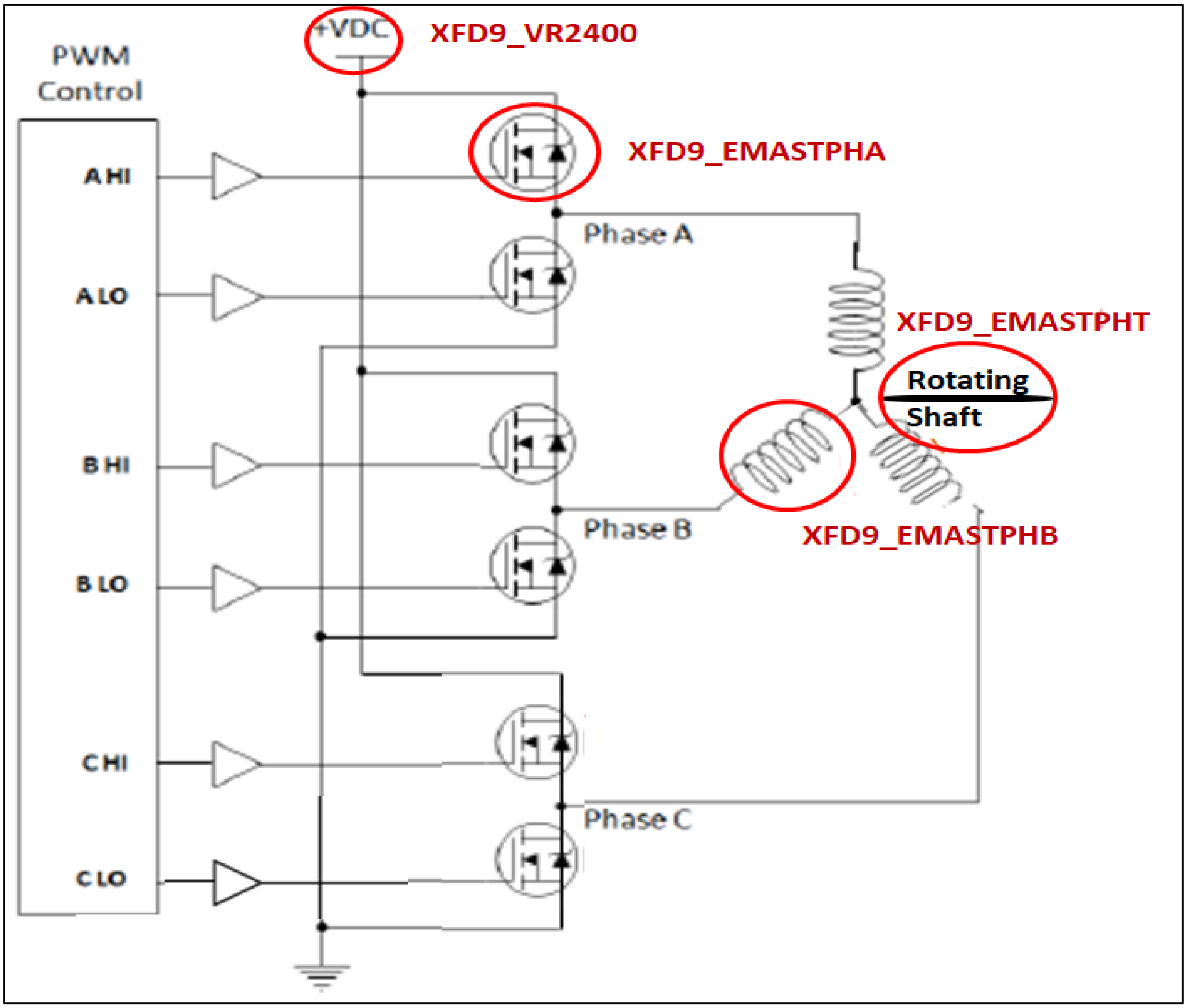


Figure 165 – EMA block diagram showing selected locations of four failures for XFD9

The full solution included a fault–tree analysis routine that compared prognostic estimates, relative magnitudes, and signature directions to determine whether or not degradation was detected and, if so, the likely source: power supply, phase–current branch (out of six), stator winding (out of three), and motor shaft. This example only presents failure for each of four major sources represented by the following NDEFs in XFD9 (see Figure 166):

- **XFD9_VR24000** SMPS for the EMA – Ripple Voltage
- **XFD9_EMASTPHA** Switching Transistor – Phase A
- **XFD9_EMASWPHB** Motor Stator Winding – Phase B
- **XFD9_EMARSPHT** Motor Rotating Shaft – Phase Total

Figure 167, Figure 168, Figure 169, and Figure 170 show the NDEF parameters and keywords used for XFD9.

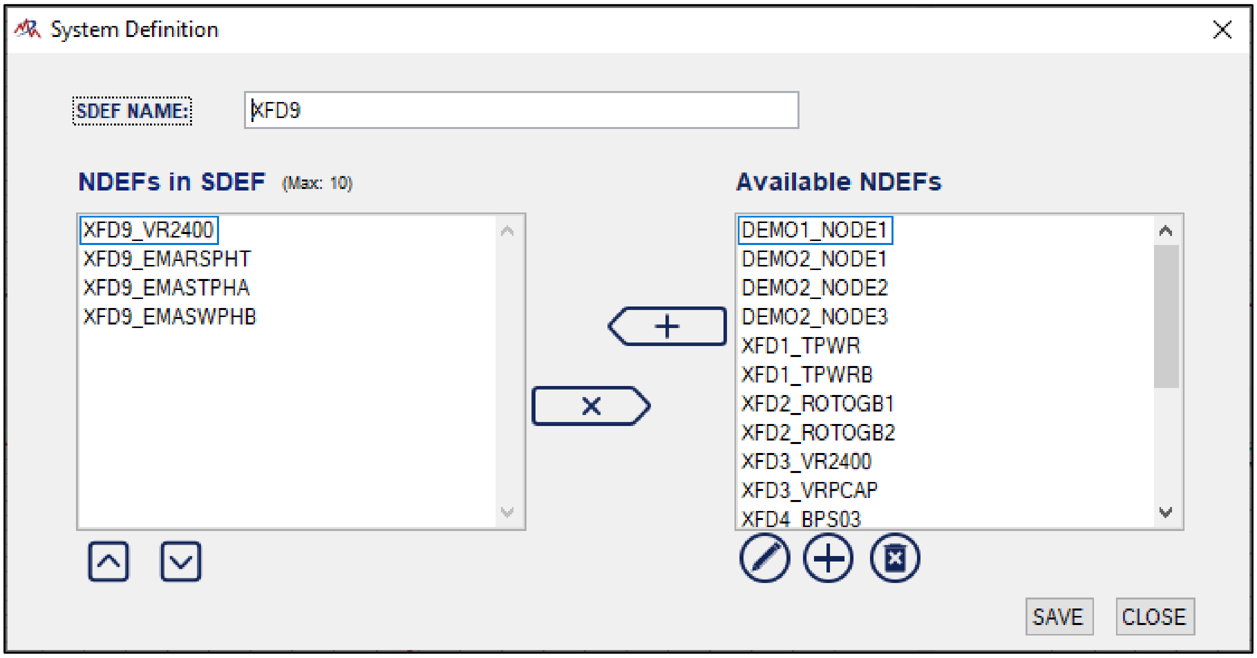


Figure 166 – XFD9 System Definition (SDEF) User Form

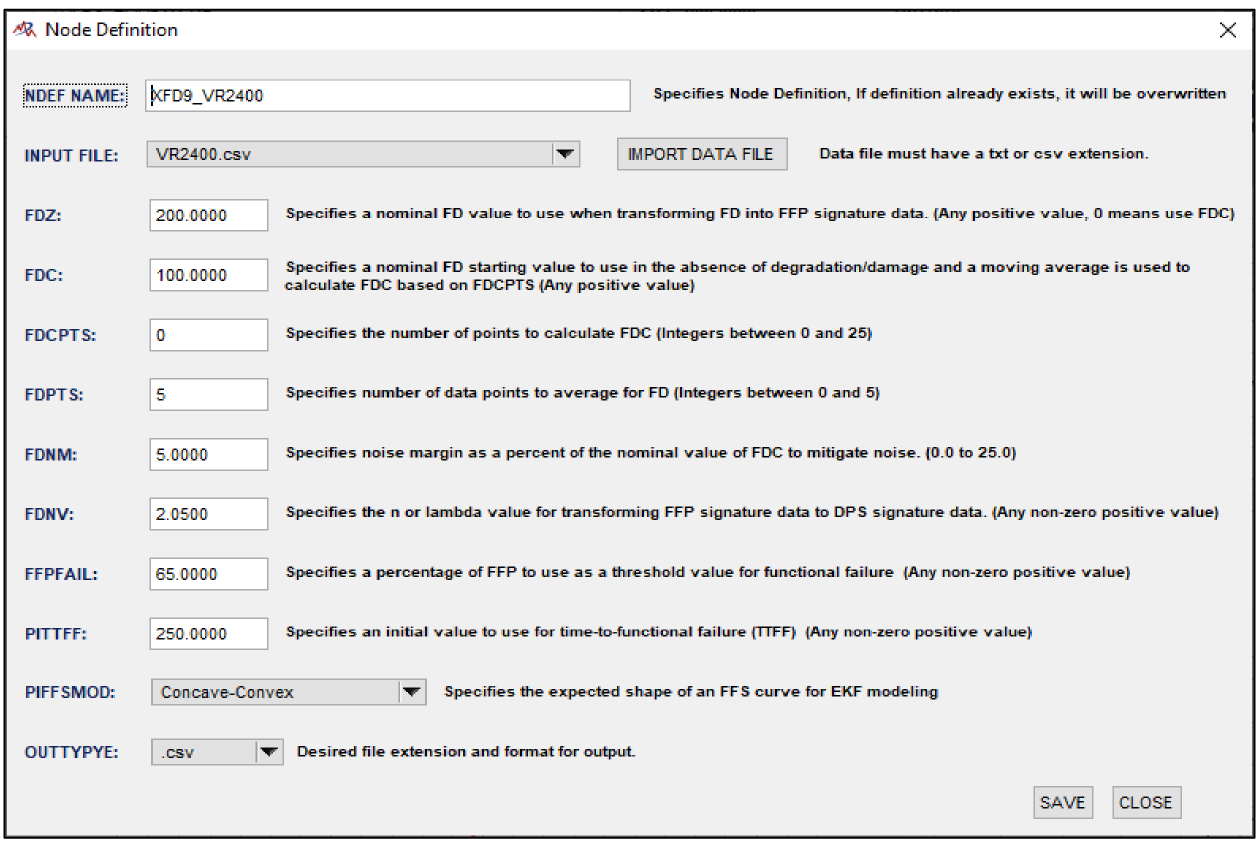


Figure 167 – Node Definition (NDEF) User Form for First Node in XFD9

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

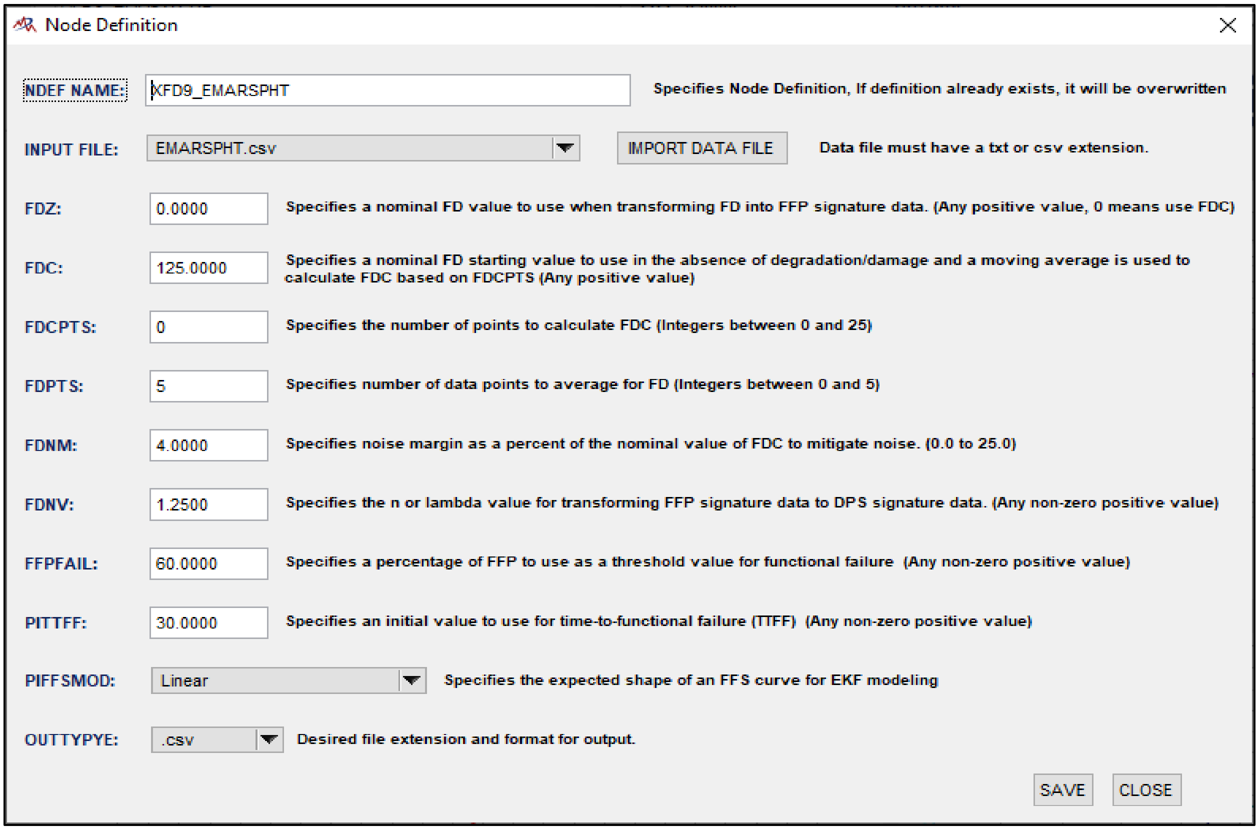


Figure 168 – Node Definition (NDEF) User Form for Second Node in XFD9

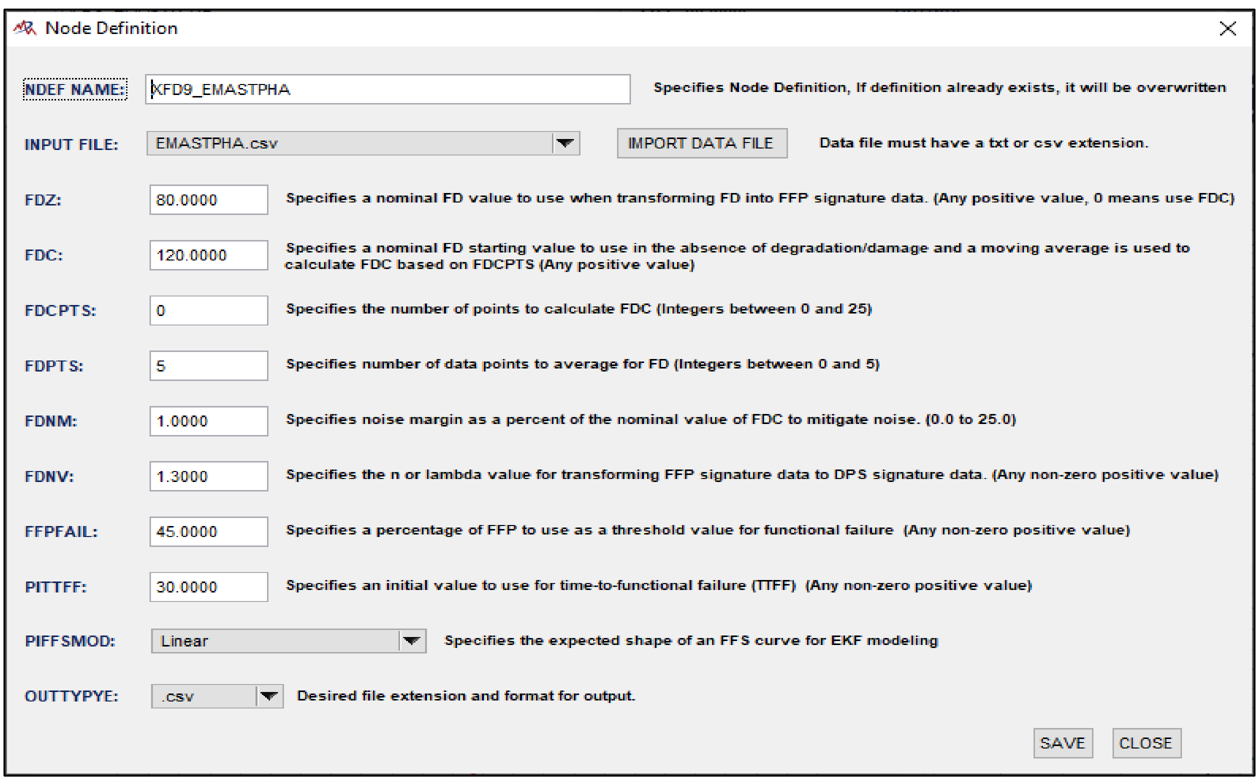


Figure 169 – Node Definition (NDEF) User Form for Third Node in XFD9

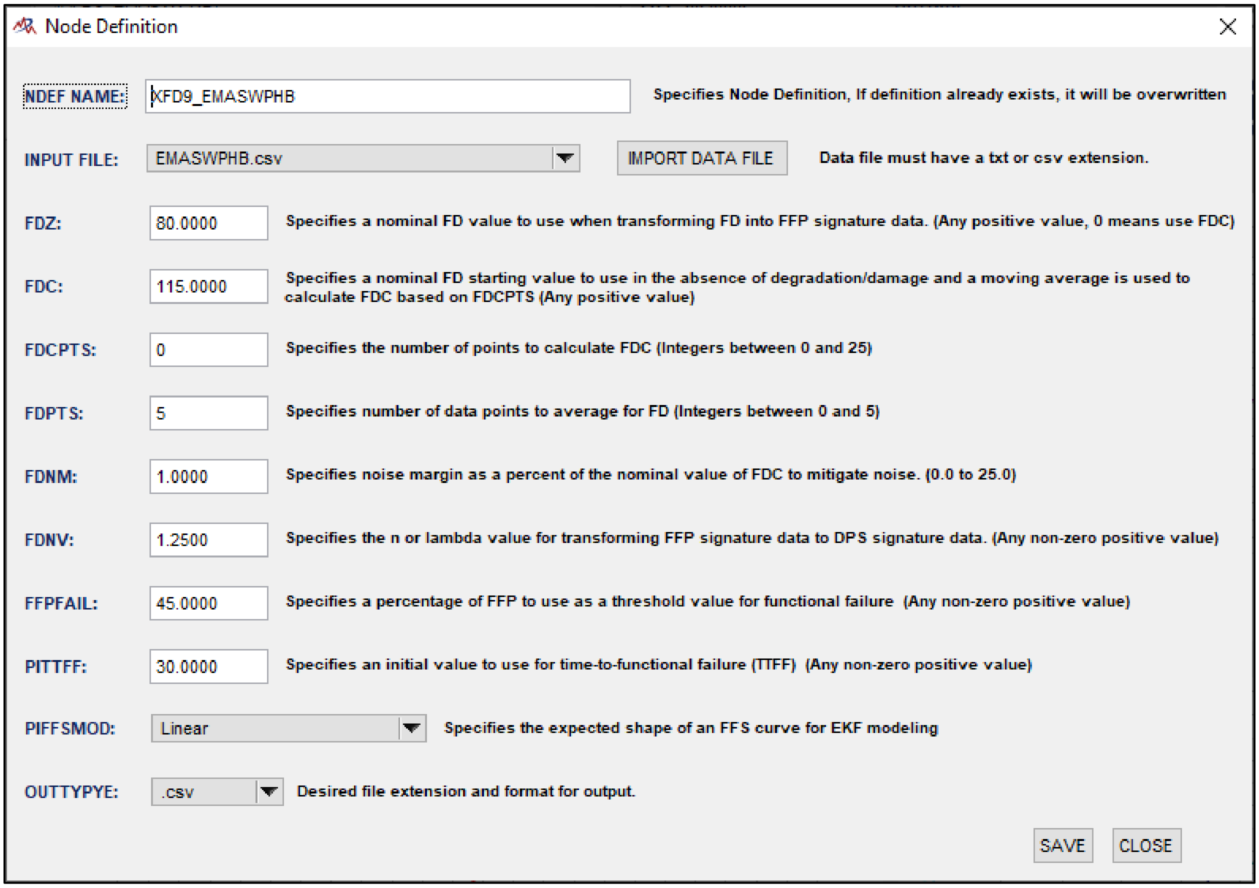


Figure 170 – Node Definition (NDEF) User Form for Fourth Node in XFD9

6.11.2 XFD9: Input Feature Data

Figure 171, Figure 172, Figure 173, and Figure 174 show the Condition–Based Data Input plots for **XFD9_VR24000, XFD9_EMASTPHA, XFD9_EMASWPHB,** and **XFD9_EMARSPHT** respectively.

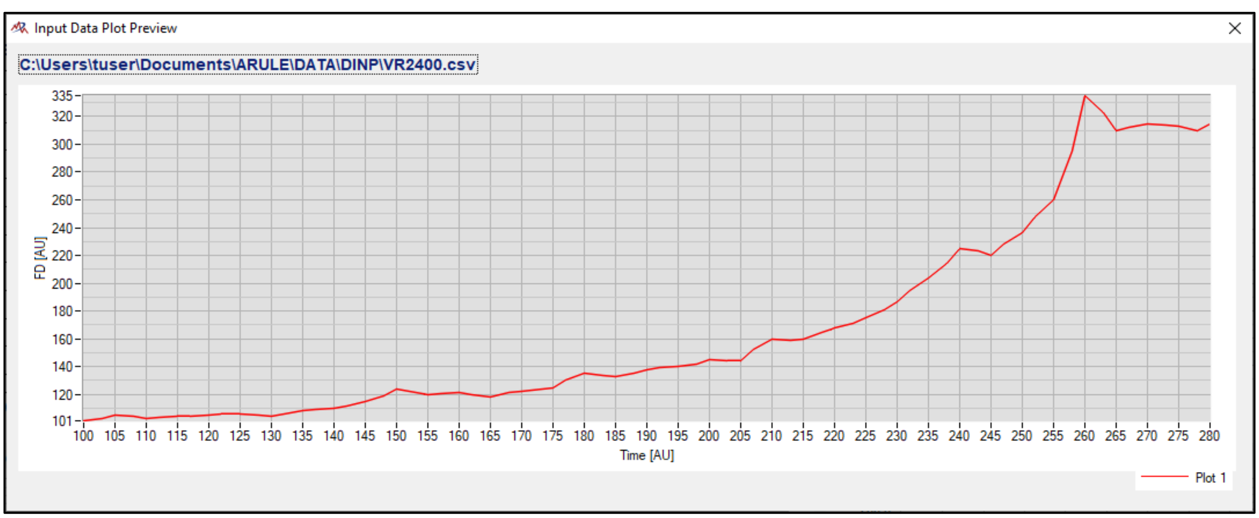


Figure 171 – XFD9 Input Data Plot Preview for Node 1

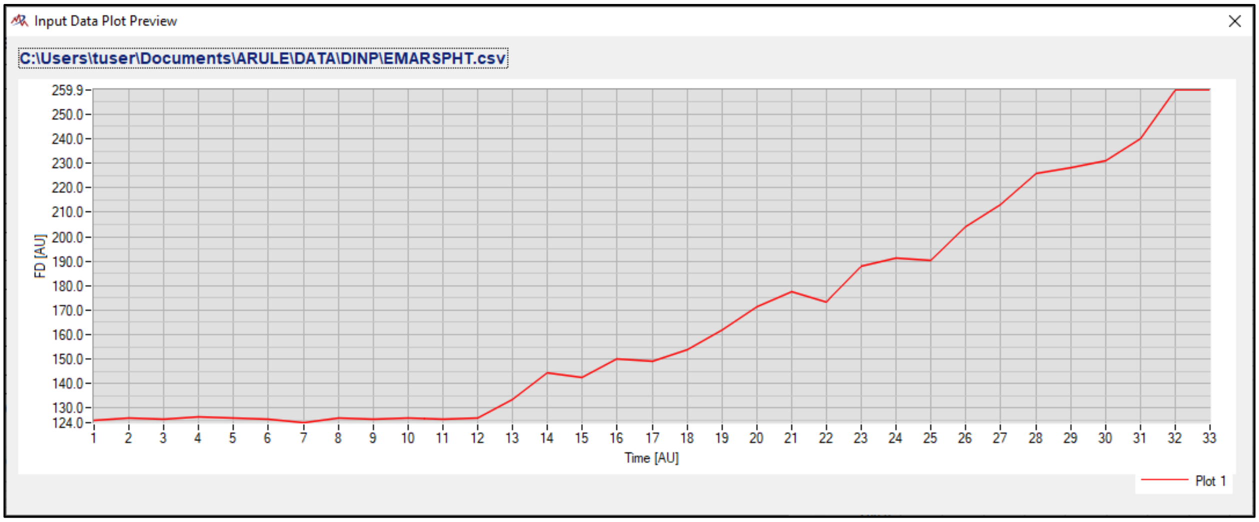


Figure 172 – XFD9 Input Data Plot Preview for Node 2

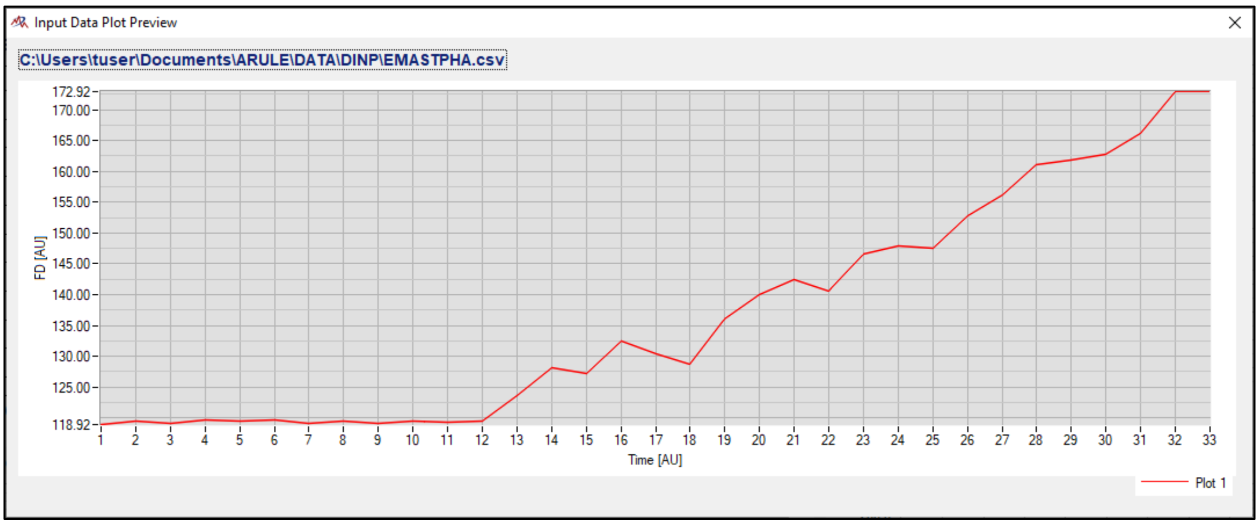


Figure 173 – XFD9 Input Data Plot Preview for Node 3

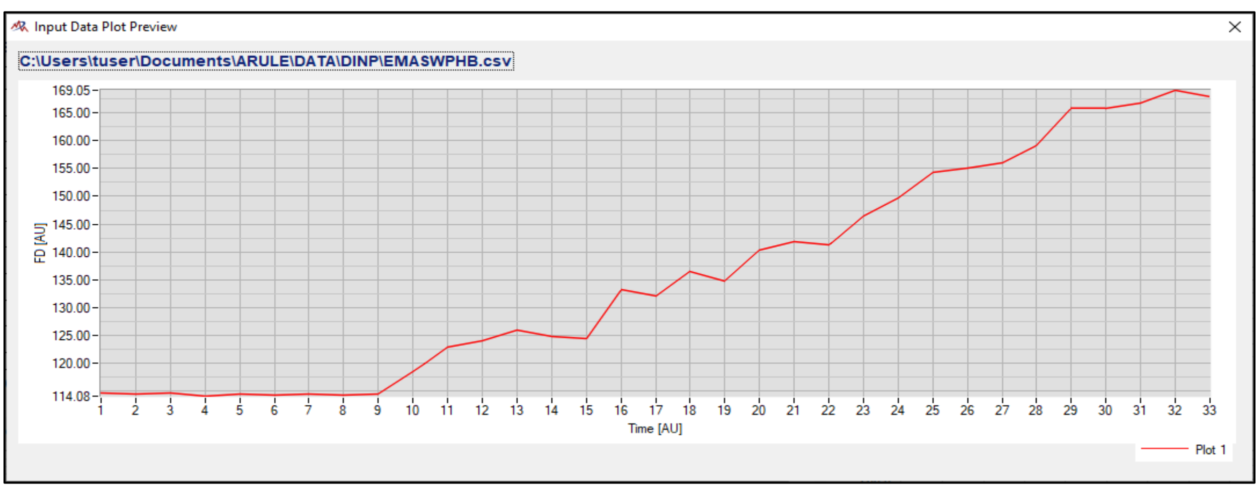


Figure 174 – XFD9 Input Data Plot Preview for Node 4

6.11.3 XFD9: Data Outputs and Prognostic Information

Figure 175, Figure 176, Figure 177, and Figure 178 show the Data Output for the two nodes in XFD9: This includes the FD, FFS Progression, along with SoH, RUL, and PH.

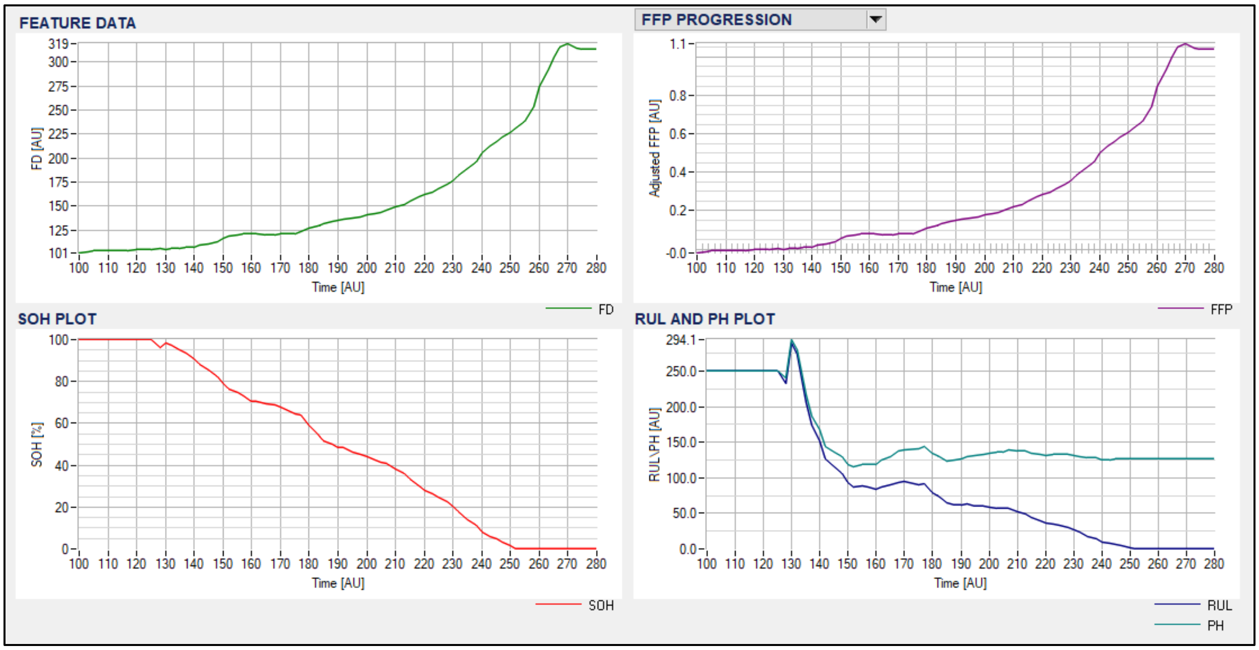


Figure 175 – Plots of the XFD9 Prognostic Information for the First Node

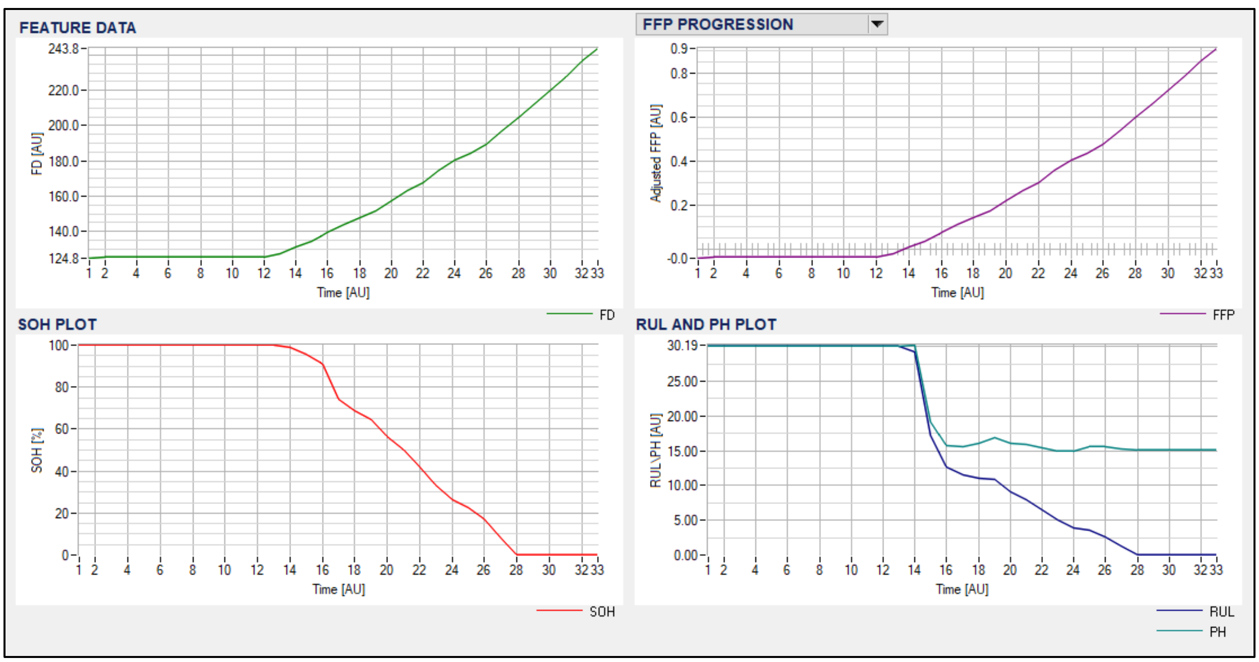


Figure 176 – Plots of the XFD9 Prognostic Information for the Second Node

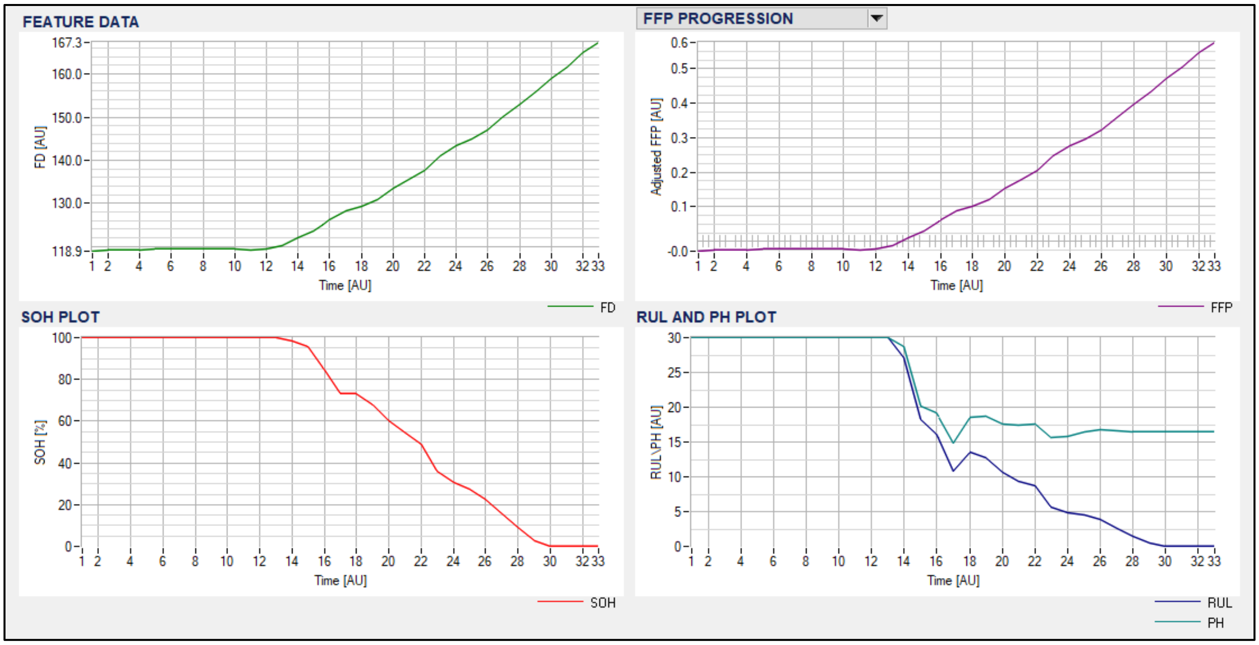


Figure 177 – Plots of the XFD9 Prognostic Information for the Third Node

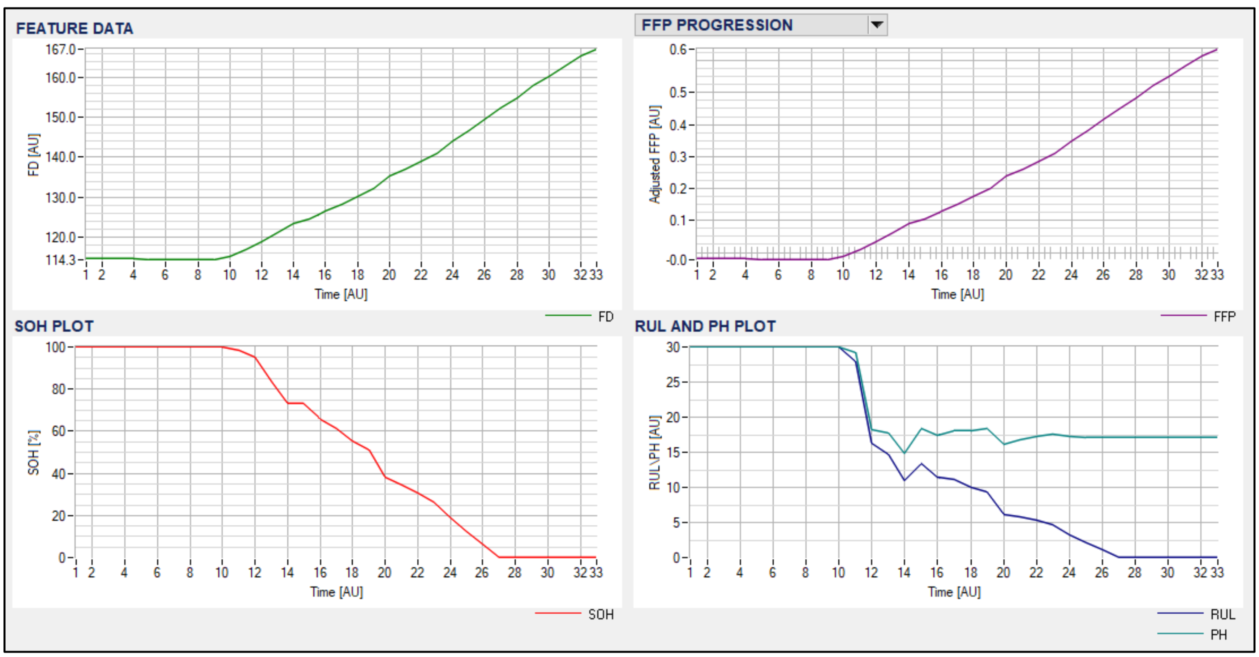


Figure 178 – Plots of the XFD9 Prognostic Information for the Fourth Node

6.11.4 XFD9: Discussion

XFD9_VR2400: A complete PHM solution would comprise 11 different sets of FD analyses: one for the power supply, one for each of the six switching transistor paths, one for each of the three stator windings, and one for the rotating shaft. This demonstration is for a failing case only. This demonstration is for the case when the power supply degrades (refer back to Figure 156, page 124).

XFD9_EMASTPHA: This demonstration is for the case where one of two sets of switching transistors for the Phase A current degrades. Referring back to Figure 162, page 128, the summation of the positive and negative halves for any undamaged switching path is essentially zero plus background noise. However, when the switching path is damaged, the positive and negative halves of the current change in the same direction – as shown for Phase A in the figure. If the change is in the positive direction, the positive half of the transistor pair is damaged. In this case, the change is in the negative direction because that was the location of the damaged switching transistor.

XFD9_EMASWPHB: This demonstration is for the case where the stator winding for the Phase B current degrades. Referring back to Figure 163, page 128, when a stator winding is damaged, instead of a change in amplitude of both halves of all three currents, there is a change in both halves of only one phase and the change is in the same direction and of equal magnitude: in this changes, the damaged stator winding is for Phase B.

XFD9_EMARSPHT: Referring back to Figure 164, page 129, when there is excessive load (weight) and/or friction (binding) on the motor shaft, all halves of all three phases change in the same direction.

7. How to Analyze Your Own Data with ARULE™?

A Sequence of Operations to Create an SDEF and NDEF(s):

In order to process your own condition-based data (CBD) set with ARULE™, the user should follow the sequential flow of operations outlined below:

1. Collect Condition-Based Data (CBD) or Feature Data (FD) that is associated with a known failure mode (simulated, experimental, historical) and then analyze the FD signature to characterize the data as corresponding to a power value that matches Figure 179.

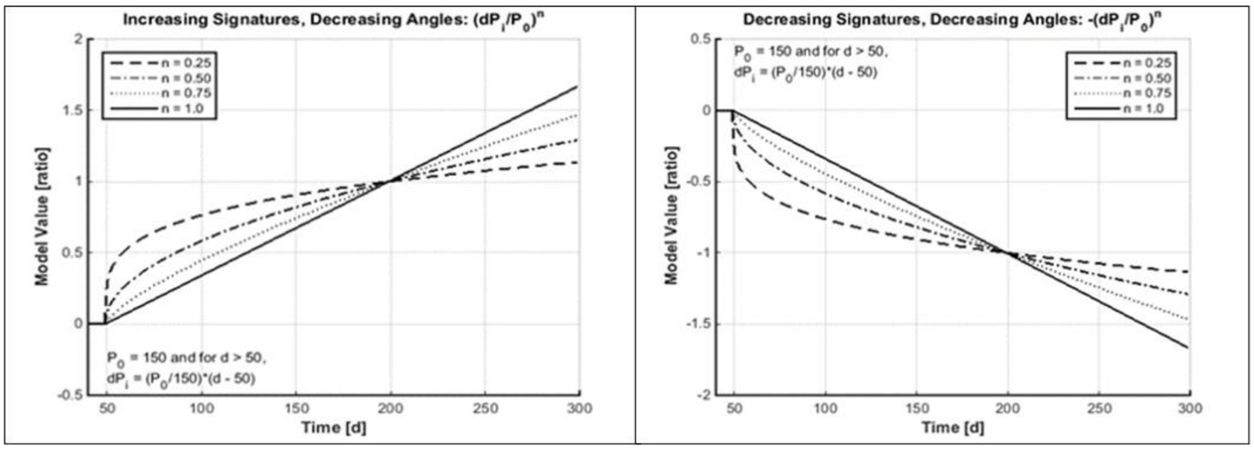


Figure 179 – Degradation signatures: increasing (left) and decreasing (right)

2. Create a New System Definition (SDEF) file by clicking on the **Circle Plus Button** located on the lower–left side of the **SDEF List** section to open the System Definition User Form. Enter an SDEF file name and then specify any existing Node Definition (NDEF) file names by using the **+ Button** (to add) and **X Button** (to remove) and filtering the NDEF(s) from the **Available NDEFs** section. An example SDEF file is displayed in Figure 180. If there is no existing NDEF file, create a new one by clicking on the **Circle Plus Button** located on the lower–left side of the **Available NDEFs** section to open the Node Definition User Form. An example of the Node Definition User Form where the new NDEF file is created, is displayed in Figure 181.

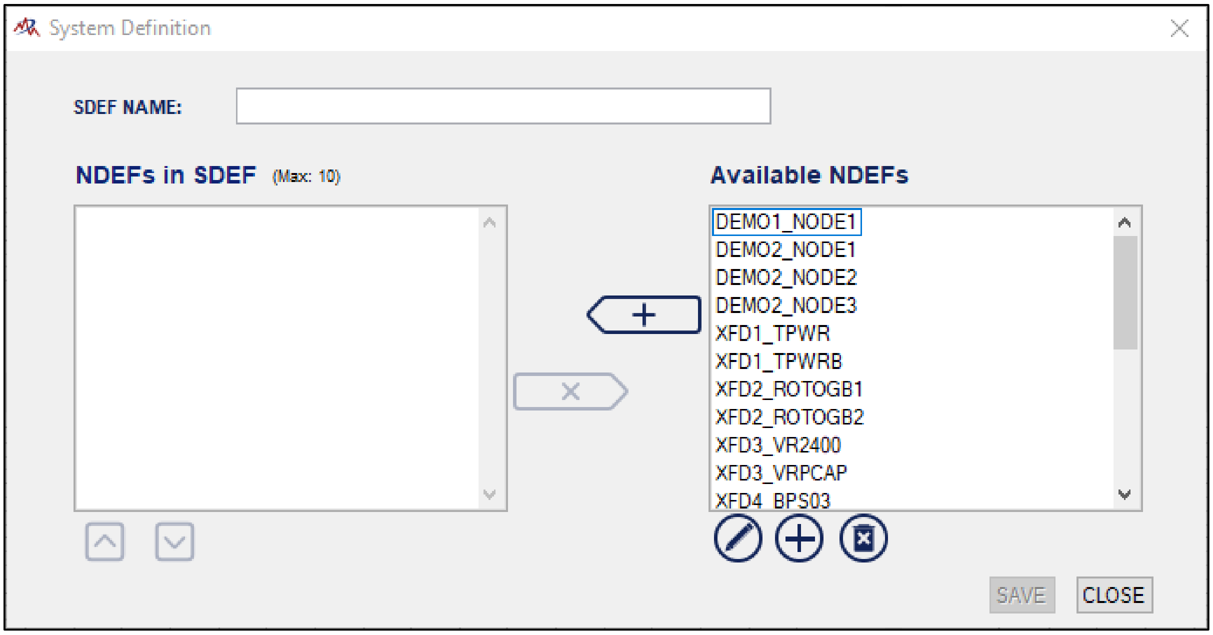


Figure 180 – Example of a new SDEF file: System Definition User Form

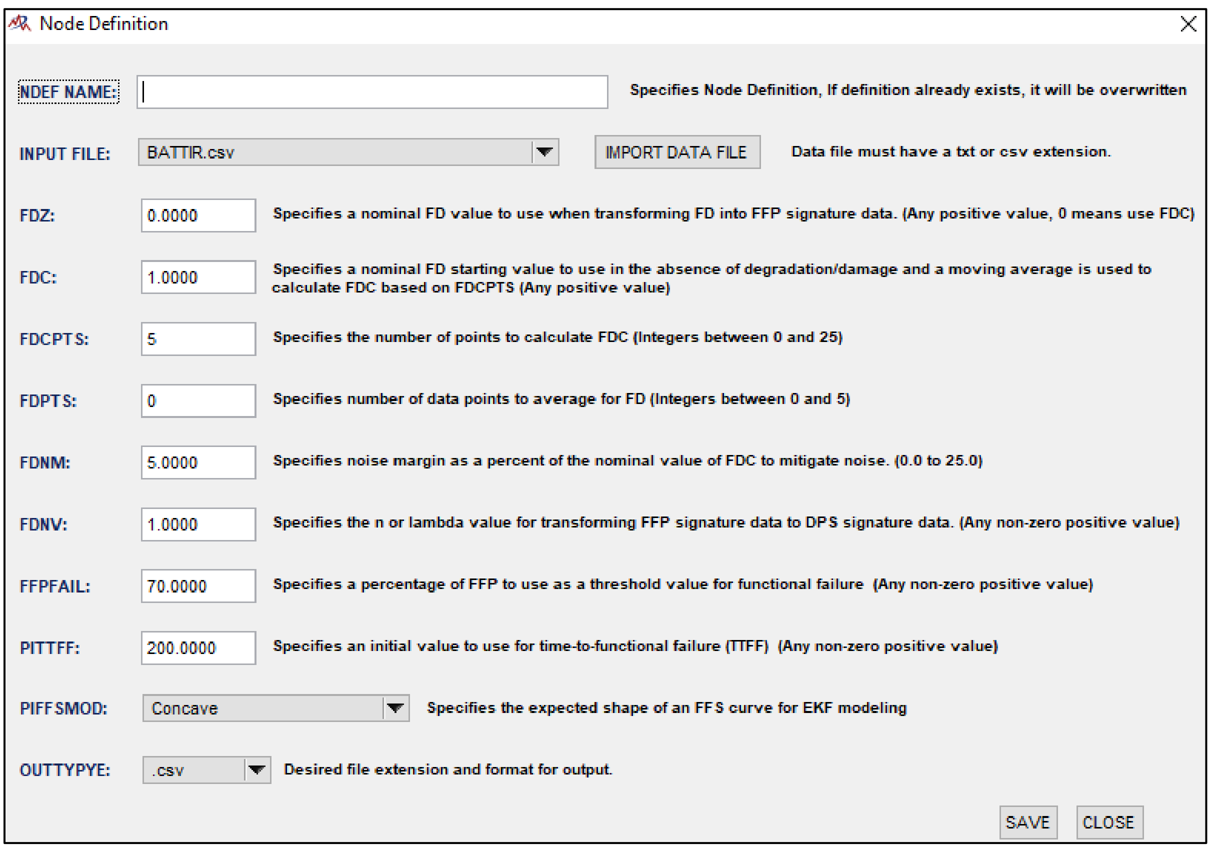


Figure 181 – Example of a new NDEF file and Node Definition User Form

In the Node Definition User Form – Name the NDEF. For this example, the NDEF will be **EXAMPLE**. Select the **IMPORT DATA FILE Button** to select an input data file using the Window's File Explorer. Note that for this example, the default input data file is set to **VRPCAP.csv** from the following default directory: **...\ARULE\DATA\DINP**, which is where a new input data file in .txt or .csv format should be placed and selected during the time of import. Figure 182 provides an example of selecting and setting the input data file to **VRPCAP.csv**, which was previously used with the XFD3 example in Section 5.5.

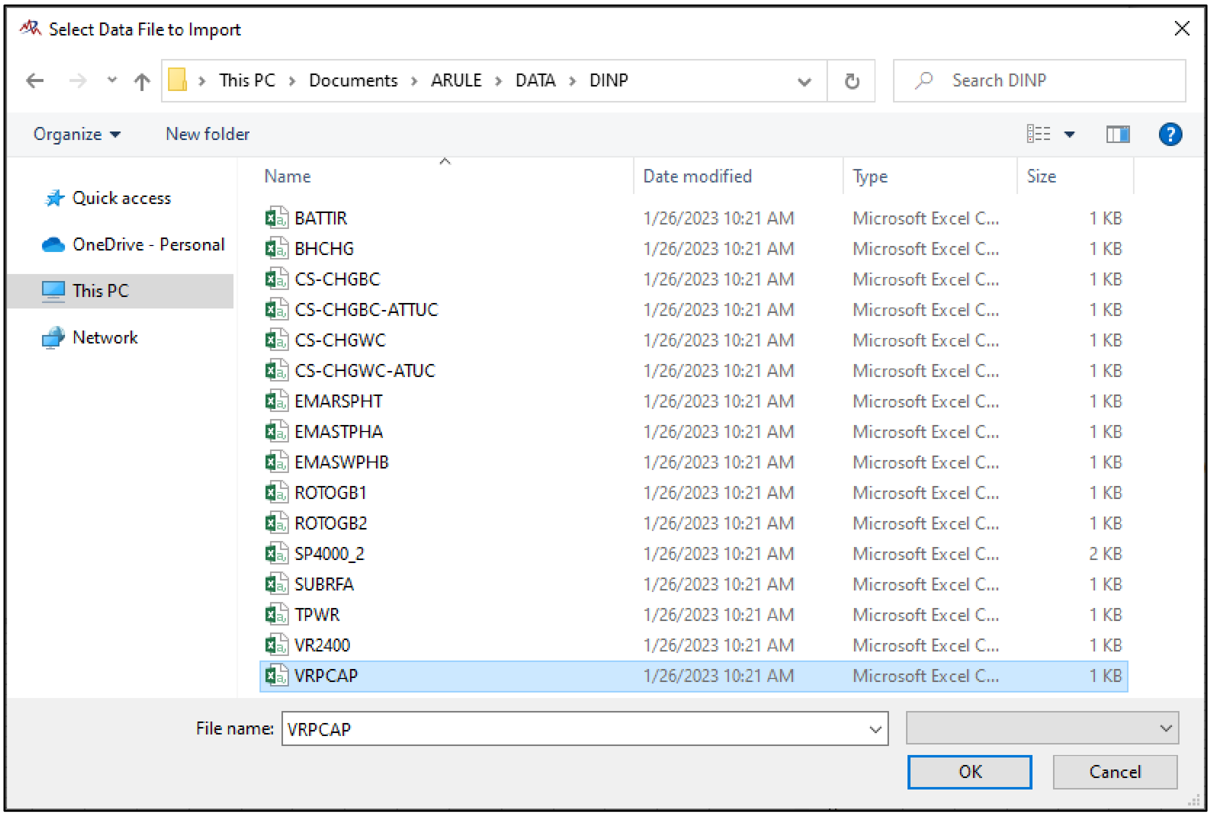


Figure 182 – DINP directory with VRPCAP.csv selected as the input data file

Important Note: The data input must be specified in either .txt or .csv format where the first value corresponds to Time, Sample ID, etc., and the second value corresponds to sensor data value, amplitude, and so on. An example for the VRCAP values in both .txt and .csv format is displayed in **Figure** 183.

	A	B	C	D
1	0	1		
2	55	2.5		
3	180	4.5		
4	320	5.1		
5	940	6.9		
6	1010	9.2		
7	1680	11.3		
8	2440	12.6		
9	2800	13.6		
10	3200	14.3		
11	3700	15.6		
12	4250	16.1		

VRPCAP.txt - Notepad

File Edit Format View Help

```
0       1
55      2.5
180     4.5
320     5.1
940     6.9
1010    9.2
1680    11.3
2440    12.6
2800    13.6
3200    14.3
3700    15.6
4250    16.1
```

Figure 183 – Example of input data: .csv (left) and .txt (right)

3. Now that the input data file has been selected, the user must now specify the ARULE™ data input processing parameters in the Node Definition User Form.

   Important Note: The default values are automatically populated, but the user may edit these parameters according to the following procedure and referencing the parameter definitions outlined in Section 4.2.2. For this example, we will set the values to match XFD3.

   Begin this process by setting the **Feature Data, Data Conditioning, Degradation signatures, and Prognostics Parameters** based on the analyses and experimentation for XFD3. The user should specify the following keyword parameters as shown in **Figure 184**, and then hit the **SAVE Button** to save this new NDEF file.

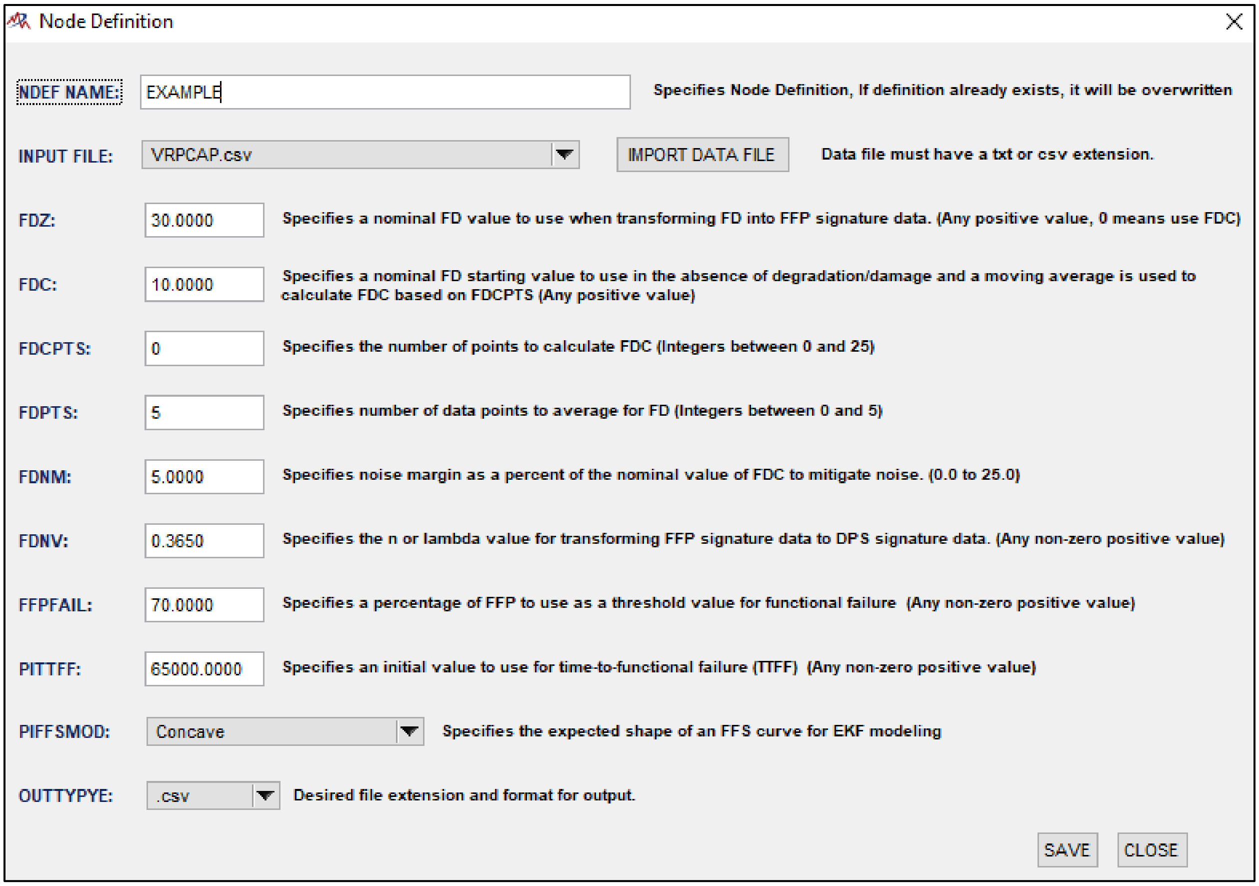


Figure 184 – Default: Values for Feature Data, Data Conditioning, Degradation signatures, and Prognostics Parameters and an example of a completed NDEF file

4. Once all NDEF file parameters have been entered and the file has been saved, click the **PLOT INFILE Button** in the **SELECTED NODE DEFINITION** section on the main window, and verify that the Condition–Based Data Input Plot populates in the **Input Data Plot Preview** tab. (**Figure** 185).

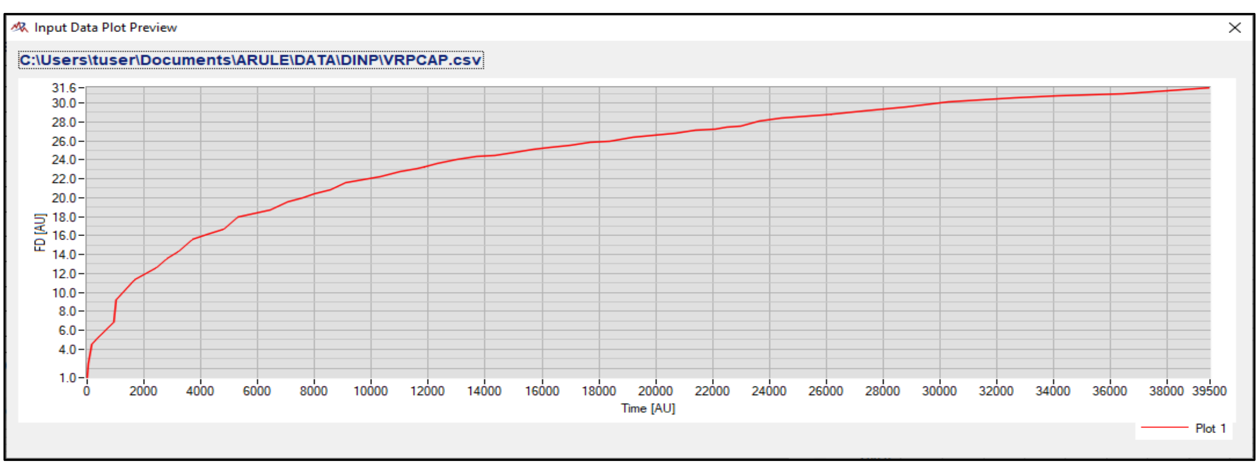


Figure 185 – Review of the CBD Input Plot

5. Moving on, the user can now specify the following ARULE™ Run Arguments in the **ARULE RUN ARGUMENTS** section on the main window:
    1. Argument 1 is auto set as a label for the name of the SDEF file you that will be processed.
    2. Argument 2 is a toggle button labeled as "Run" to specify the mode of operation.
        a. When the toggle button is disabled, the program is in the Default Processing Mode and will process all data points for the input data files that were defined in the NDEF file.
        b. When the toggle button is enabled, the program is in Check Point Restart (CPT) Mode. In CPT mode, the user has the ability to process N# of data points between 1–10000 in the text input.

    Important Note: For this example, we will not use any ARULE™ Run Arguments to match XFD3.

6. Once all ARULE™ Run Arguments have been verified (if need be), the user can now select the RUN ARULE Media Button in the ARULE RUN Arguments section on the main window and verify success in the SYSTEM MESSAGES section (Figure 186).

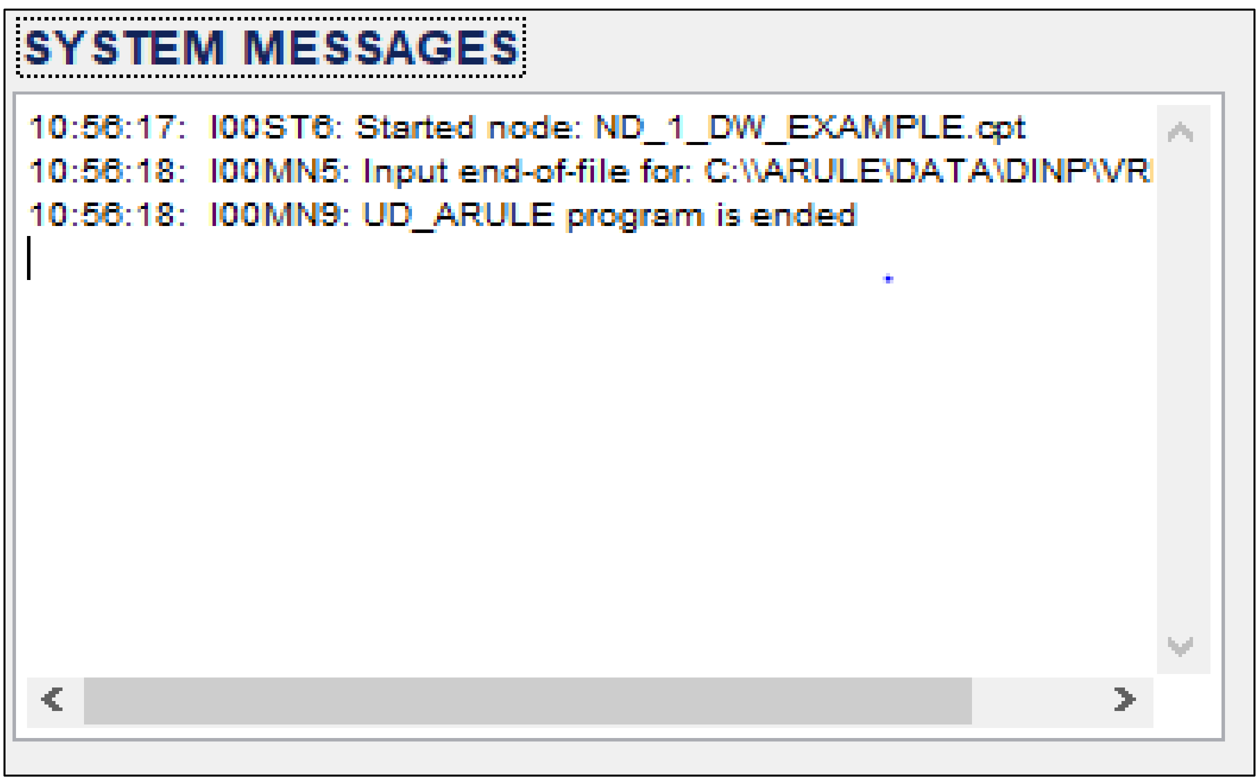


Figure 186 – Example message for successfully processing of the input data by ARULE™

Important Note: If there is an error message, please verify the data input file contents, format, and NDEF File input parameters. Any errors, program logs, and GUI operation issues will be displayed here.

7. To view the data output files, the program will automatically display the data output plots for the first NDEF. If multiple NDEFs are processed, then the user must select the data output file associated with those NDEFs in the **CURRENT ARULE RUN** drop–down section on the main window. Note that there should be a data output file for each NDEF that was specified. Once a data output file is selected, the user can see the four plots on the main window of the GUI (Figure 187): FD, FFS Progression, along with SoH, RUL, and PH.
    - Feature Data (FD) Plot – This plot shows the Condition–based Data (CBD) input to ARULE™.
    - Fault to Failure Signature (FFS) Progression Plot – This plot provides the transformation data input to ARULE™ from the original Condition–based Data (CBD) plot and can be selected from the dropdown selection box.
    - State–of–Health (SoH) Plot – This provides a visual for the SoH % vs. Time plot for the monitored system based on the input data set, and shows the onset of degradation when the SoH starts decreasing from 100%.
    - Remaining Useful Life (RUL) and Prognostic Horizon (PH) Plot – This plot provides a visual for Remaining Useful Life and Prognostic Horizon vs. Time.

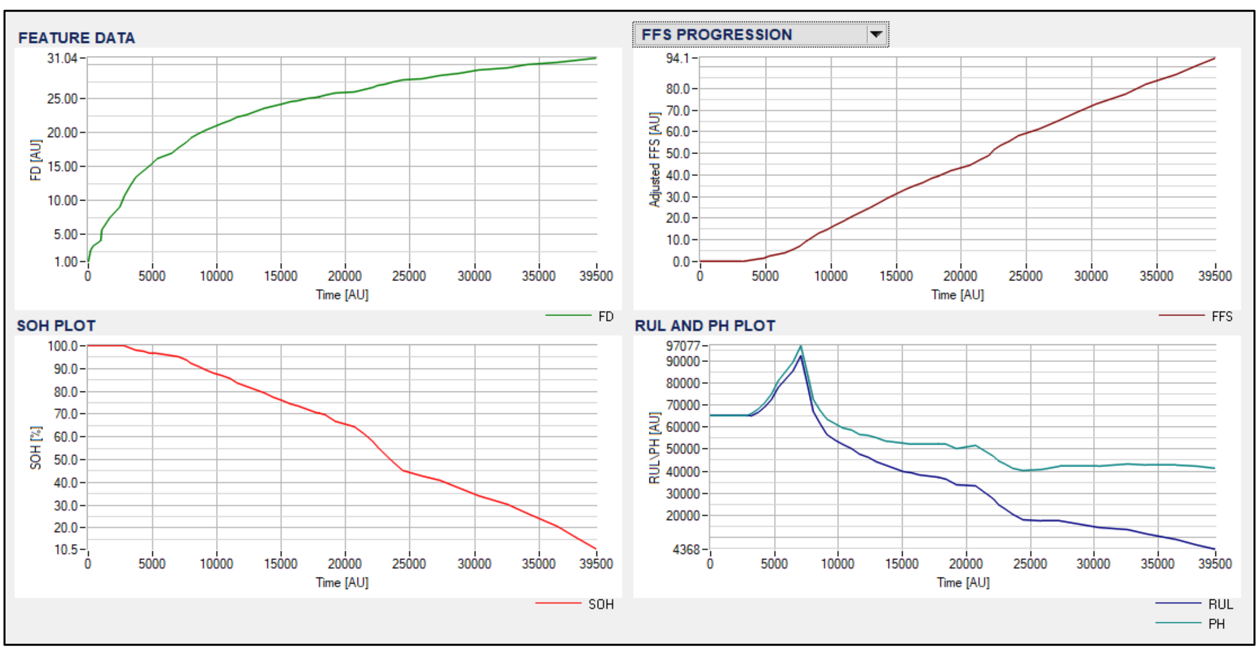


Figure 187 – Example Data Output plots: FD, FFS Progression, along with SoH, RUL, and PH

8. Once the user has previewed the data output plots, the GUI provides an option to save individual plot data to the following directory: **...\ARULE™\DATA\DINP** (**SAVE OUTPUT Button** below the **CURRENT ARULE RUN** drop–down section on the main window).
9. Upon completing this step, you have successfully processed your own input data set with ARULE™. You can exit the program, import a new data set, or revisit the program examples. If you are interested in embedding ARULE™ in your own CBM, PHM, or IVHM software system please contact an ARULE™ Applications Engineer at Ridgetop Group.

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com

8. Appendix

8.1 Frequently Asked Questions (FAQ)

1. **Question: Are there any prerequisites for installing and running the ARULE™ GUI?**
   Answer: No, all prerequisites for the ARULE GUI are included with installation wizard.

2. **Question: Does Ridgetop Group offer software training programs?**
   Answer: Yes, Ridgetop Group offers on–site and remote training programs for PHM, CBM, and IVHM applications. Contact a Ridgetop Group directly or through one of its representatives for additional details.

3. **Question: What types of prognostic solutions does Ridgetop Group offer?**

   Answer: Ridgetop Group's vision statement is to provide best in class CBM, PHM, and reliability engineering products and services for Aerospace, Defense, Transportation, Energy, Medical, and Industrial applications. The real world examples that documented in Section 5, utilize Ridgetop's innovative product lines Sentinel Motion and Sentinel Power. Please visit the Ridgetop Group website at www.ridgetopgroup.com to find details pertaining to other product lines.

8.2 Table of Figures

Figure 1 – Framework diagram for a PHM system 2
Figure 2 - Diagram of a 3-stage PHM system: (1) SENSING STAGE, (2) FEATURE EXTRACTION STAGE, and (3) PROGNOSING STAGE to Produce Prognostic Information with ARULE 3
Figure 3 - General operational view of how the sensors, health monitoring, and assessment functions inherent within the electronic system enable the PHM capability operational processes [IEEE 1856-2017] 4
Figure 4 - Examples of noisy CBD: (A) phase currents and (B) y-direction accelerometer forces 5
Figure 5 - Continual (top) and periodic sampling (bottom) 6
Figure 6 - Data sampling at 1/TP rate with burst mode at (s=7)/TB rate 6
Figure 7 - SMPS output: six extractable features (A) and an extracted damped-ringing response (B) 8
Figure 8 - Idealized damped-ringing response showing other extractable features 8
Figure 9 - Example of an FD signature (resonant frequency) 11
Figure 10 - Annotated FD signature: noise, degradation, linear degradation (feedback), and curvilinear degradation (feedback no longer effective) 12
Figure 11 - Operational diagram: external program using ARULE through its API 13
Figure 12 – Framework for a GUI–based ARULE™ 14
Figure 13 - An example instantiation of defined set of API keywords and values to use the ARULE kernel 18
Figure 14 - FD signature data transformed into an FFP signature data 19
Figure 15 - FFP signature data transformed into DPS data 20
Figure 16 - Power Function: FFP on the left, DPS on the right 20
Figure 17 - Examples of FD input, their characteristic curves, and initial evaluation of probable n values 21
Figure 18 - Examples of DPS transforms for specified FDNV values for the FD input in Figure 17 21
Figure 19 - DPS data transformed into FFS data – ideal transforms are red-colored lines 22
Figure 20 - EKF: adapt data point to 1st model, adapt 2nd model to adapted data, then use adapted 2nd model to predict RUL, PH, and SoH. 23
Figure 21 - Piecewise-linear 2nd EKF model: random-walk of data through multiple data spaces 24
Figure 22 - Plots of the final signature models specified by the PIFFSMOD input parameter for ARULE to perform EKF processing of FFS data 25
Figure 23 - Adjusted FFS plot (left) of the transformed input to final prediction stage (Extended Kalman Filtering) and plots of the resulting RUL/PH estimates. 25
Figure 24 - Plots of RUL/PH estimates (left) and SoH estimates (right): 25% &10% margins of accuracy (blue dot-dashed lines) and ideal estimates are included (red-dashed lines). 26
Figure 25 - Ideal plots of RUL, PH, and SOH values with 100% prognostic accuracy 27
Figure 26 - Example of a failure distribution 28
Figure 27 - Idealized plots of RUL and PH estimates given initial values having +/- 50% variation 29
Figure 28 - Example plots of RUL given three values of initial estimates: 200, 115; and 50 days 29
Figure 29 – Example of pseudo–ideal prognostic information. Ideal input FD signature data, non–ideal (300 days) estimate of time–to–failure (200 days) – an initial estimate error of 50 percent 31
Figure 30 - Plot of RUL and PH estimates where the horizontal dashed lines represent 25% and 10% margins of accuracy 32
Figure 31 - Ideal and non-ideal plots for terminology and metrics explanation. 35
Figure 32 - Example plots to illustrate prediction information (except for SOH) 36
Figure 33 – ARULE™ installation folder saved in This PC > Downloads 41
Figure 34 – Folder is located at ...\Downloads\ARULE_01.27.2023 42

Figure 35 – ARULE™ installation wizard initializing ... 42
Figure 36 – View of the installation wizard for the ARULE™ installation directory. Click Next >> to continue ... 43
Figure 37 – View of the installation wizard that prompts the user to select the National Instruments Software License Agreement. Click Next>> to continue ... 43
Figure 38 – View of the installation wizard to start the installation. Click Next >> to continue ... 44
Figure 39 – View of the installation wizard installing the software ... 44
Figure 40 – View of the installation wizard when the installation has completed ... 45
Figure 41 – Installation directory of ARULE™ ... 45
Figure 42 – License verification prompt asking user to retrieve a license from the server ... 46
Figure 43 – Logging into the license server ... 46
Figure 44 – Message that software license is activated and installed correctly ... 47
Figure 45 – Selecting the ARULE Directory location ... 48
Figure 46 – View of the ARULE™ software being opened after license installation ... 48
Figure 47 – Settings, System, Display, Scale and Layout: recommended values ... 49
Figure 48 – Single Window GUI: System Definition Input and Data Output ... 50
Figure 49 – Visual of the System Definition Input ... 51
Figure 50 – Visual of the SDEF List section for an SDEF file on the GUI ... 52
Figure 51 – Visual of the System Definition User Form ... 53
Figure 52 – The corresponding SDEF file in .txt format ... 53
Figure 53 – Visual of the NDEF file scroll box on the main window ... 54
Figure 54 – Visual of the NDEF file scroll box ... 55
Figure 55 – Visual of the Node Definition User Form ... 56
Figure 56 – Example of an NDEF being edited in raw .txt by Notepad ... 58
Figure 57 – Plot Input File feature in the SELECTED NODE DEFINITION section ... 59
Figure 58 – Visual of CBD Input Data Plot Preview tab ... 59
Figure 59 – ARULE™ Run Arguments section ... 60
Figure 60 – Example of enabling CPT mode and wishing to process 500 data points ... 61
Figure 61 – Example of a system message: successful processing ... 62
Figure 62 – Example of a Data Output section ... 63
Figure 63 – View of Data Output dropdown options. ... 63
Figure 64 – View of dropdown option for SAVED ARULE RUN ... 64
Figure 65 – Example of downloaded plot information in a time stamped folder ... 64
Figure 66 – View of System Messages ... 68
Figure 67 – View of Global Menu Options ... 68
Figure 68 – First pop-up window when the Reset ARULE Directory option is selected ... 69
Figure 69 – File Explorer window to specify the new location for the ARULE directory ... 69
Figure 70 – View of About ARULE option when selected from the Menu ... 70
Figure 71 – Prompt to exit the ARULE™ application ... 70
Figure 72 – Flow–diagram for transforming CBD signature data into FFS data ... 71
Figure 73 – Example of high–rate burst–mode sampling at a low–rate periodic–mode sampling ... 72
Figure 74 – DEMO1 System Definition (SDEF) User Form ... 73
Figure 75 – DEMO1 Node Definition User Form ... 73
Figure 76 – DEMO1 Input Feature Data and ARULE RUN ARGUMENTS section ... 74
Figure 77 – Data Output section for DEMO1 ... 74

Figure 78 – DEMO2 System Definition (SDEF) User Form ... 76
Figure 79 – Node Definition (NDEF) User Form for the First Node in DEMO2 ... 76
Figure 80 – Node Definition (NDEF) User Form for the Second Node in DEMO2 ... 77
Figure 81 – Node Definition (NDEF) User Form for Third Node in DEMO2 ... 77
Figure 82 – DEMO2 Input Feature Data for the First and Second Node ... 78
Figure 83 – DEMO2 Input Feature Data and for Third Node ... 78
Figure 84 – Data Output for the First Node in DEMO2 ... 79
Figure 85 – Data Output for Second Node in DEMO2 ... 79
Figure 86 – Data Output for Third Node in DEMO2 ... 80
Figure 87 – Examples of differences between the input data in SP4000–1.csv and SP4000–2.csv ... 80
Figure 88 – XFD1 System Definition (SDEF) User Form ... 81
Figure 89 – Node Definition (NDEF) User Form for the First Node in XFD1 ... 82
Figure 90 – Node Definition (NDEF) User Form for the Second Node in XFD1 ... 83
Figure 91 – XFD1 Input Feature Data for the First and Second Node. ... 83
Figure 92 – Plots of the XFD1 Prognostic Information for the First Node ... 84
Figure 93 – XFD1 plots of the Prognostic Information for the Second Node ... 84
Figure 94 – RotoSense diagram, unit, and assembly in gearbox system. ... 85
Figure 95 – XFD2 System Definition (SDEF) User Form ... 86
Figure 96 – Node Definition (NDEF) User Form for the First Node in XFD2 ... 86
Figure 97 – Node Definition (NDEF) User Form for the Second Node in XFD2 ... 87
Figure 98 – XFD2 Input Data Plot Preview for Node 1 ... 87
Figure 99 – XFD2 Input Data Plot Preview for Node 2 ... 88
Figure 100 – Plots of the XFD2 Prognostic Information for the First Node ... 88
Figure 101 – Plots of the XFD1 Prognostic Information for the Second Node ... 89
Figure 102 – Pinion gear after final notching @ 161 hours (left) and after the test was ended ... 90
Figure 103 – Original data collection used to create input data for XFD2 ... 90
Figure 104 – Examples of Feature Data (FD) from power supply output ... 91
Figure 105 – XFD3 System Definition (SDEF) User Form ... 92
Figure 106 – Node Definition (NDEF) User Form for First Node in XFD3 ... 92
Figure 107 – Node Definition (NDEF) User Form for Second Node in XFD3 ... 93
Figure 108 – XFD3 Input Data Plot Preview for Node 1 ... 93
Figure 109 – XFD3 Input Data Plot Preview for Node 2 ... 94
Figure 110 – Plots of the XFD3 Prognostic Information for the First Node ... 94
Figure 111 – Plots of the XFD3 Prognostic Information for the Second Node ... 95
Figure 112 – XFD4 System Definition (SDEF) User Form ... 96
Figure 113 – XFD4 Node Definition (NDEF) User Form ... 97
Figure 114 – XFD4 Input Data Plot Preview ... 97
Figure 115 – Plots of the XFD4 Prognostic Information ... 98
Figure 116 – XFD5 System Definition (SDEF) User Form ... 99
Figure 117 – Node Definition (NDEF) User Form for First Node in XFD5 ... 100
Figure 118 – Node Definition (NDEF) User Form for Second Node in XFD5 ... 100
Figure 119 – XFD5 Input Data Plot Preview for Node 1 and Node 2 ... 101
Figure 120 – Plots of the XFD5 Prognostic Information for the First Node ... 101
Figure 121 – Plots of the XFD5 Prognostic Information for the Second Node ... 102
Figure 122 – XFD6 System Definition (SDEF) User Form ... 103

Figure 123 – XFD6 Node Definition (NDEF) User Form 104
Figure 124 – XFD6 Input Data Plot Preview 104
Figure 125 – Plots of the XFD6 Prognostic Information 105
Figure 126 – Published data: battery capacity [Goebel, et.al, 2008, Prognostics in Battery Health Management, IEEE Instrumentation and Measurements Magazine] 106
Figure 127 - CellSage screen shot showing loss of charge: annual temperature profile for Tucson, AZ for worst-case, average-case, and best-case simulation conditions 107
Figure 128 – XFD7 System Definition (SDEF) User Form 108
Figure 129 – Node Definition (NDEF) User Form for First Node in XFD7 108
Figure 130 – Node Definition (NDEF) User Form for Second Node in XFD7 109
Figure 131 – Node Definition (NDEF) User Form for Third Node in XFD7 109
Figure 132 – XFD7 Input Data Plot Preview for Node 1 110
Figure 133 – XFD7 Input Data Plot Preview for Node 2 110
Figure 134 – XFD7 Input Data Plot Preview for Node 3 111
Figure 135 – Plots of the XFD7 Prognostic Information for the First Node 111
Figure 136 – Plots of the XFD7 Prognostic Information for the Second Node. 112
Figure 137 – Plots of the XFD7 Prognostic Information for the Third Node 112
Figure 138 – Virginia class submarine with sensor mast attached to the sail 113
Figure 139 – Universal modular mast (UMM) cartridge and cabling system 113
Figure 140 – Twisted triplet 24–AWG segment showing a kink; microscopic view of conductor failure 114
Figure 141 – SMRT Sweeping Probe block diagram 115
Figure 142 – SMRT Sweeping Probe and SPU. 115
Figure 143 – CBD: Plots of the IFF transform – undamaged cable (left) and damaged cable (right) 115
Figure 144 – ARULE™ Application 116
Figure 145 – XFD8 System Definition (SDEF) User Form 116
Figure 146 – Node Definition (NDEF) User Form for First Node in XFD8 117
Figure 147 – Node Definition (NDEF) User Form for Second Node in XFD8 117
Figure 148 – XFD8 Input Data Plot Preview for Node 1 and Node 2 118
Figure 149 – Plots of the XFD8 Prognostic Information for the First Node 118
Figure 150 – Plots of the XFD8 Prognostic Information for the Second Node 119
Figure 151 – EMA (controller and motor) test bed 120
Figure 152 – Test Bed diagram 121
Figure 153 – Collect and process data from a switch–mode power supply: RD2400 in Figure 154 121
Figure 154 – Collect and process data from the EMA 122
Figure 155 – Diagram: experimental solution 123
Figure 156 – Extracted FD from DS3: ripple voltage 124
Figure 157 – Phase–current sampling: sensors, sensor output, sampled data, and examples of noise 124
Figure 158 – Phase–current noise: electrical before (left) and after (right) conditioning 125
Figure 159 – Phase–current noise: start–stop (left) and load–variation (right) 125
Figure 160 – Current offset and distortion 126
Figure 161 – Brushless DC motor with floating reference for the stator windings 127
Figure 162 – Degradation in one–of–six transistor paths (phase A, negative path) 128
Figure 163 – Degradation in one–of–three stator windings 128
Figure 164 – Excessive friction/load on EMA rotating shaft 129
Figure 165 – EMA block diagram showing selected locations of four failures for XFD9 130

Figure 166 – XFD9 System Definition (SDEF) User Form 131
Figure 167 – Node Definition (NDEF) User Form for First Node in XFD9 131
Figure 168 – Node Definition (NDEF) User Form for Second Node in XFD9 132
Figure 169 – Node Definition (NDEF) User Form for Third Node in XFD9 132
Figure 170 – Node Definition (NDEF) User Form for Fourth Node in XFD9 133
Figure 171 – XFD9 Input Data Plot Preview for Node 1 133
Figure 172 – XFD9 Input Data Plot Preview for Node 2 134
Figure 173 – XFD9 Input Data Plot Preview for Node 3 134
Figure 174 – XFD9 Input Data Plot Preview for Node 4 134
Figure 175 – Plots of the XFD9 Prognostic Information for the First Node 135
Figure 176 – Plots of the XFD9 Prognostic Information for the Second Node 135
Figure 177 – Plots of the XFD9 Prognostic Information for the Third Node 136
Figure 178 – Plots of the XFD9 Prognostic Information for the Fourth Node 136
Figure 179 – Degradation signatures: increasing (left) and decreasing (right) 138
Figure 180 – Example of a new SDEF file: System Definition User Form 139
Figure 181 – Example of a new NDEF file and Node Definition User Form 139
Figure 182 – DINP directory with VRPCAP.csv selected as the input data file 140
Figure 183 – Example of input data: .csv (left) and .txt (right) 141
Figure 184 – Default: Values for Feature Data, Data Conditioning, Degradation signatures, and Prognostics Parameters and an example of a completed NDEF file 142
Figure 185 – Review of the CBD Input Plot 142
Figure 186 – Example message for successfully processing of the input data by ARULE™ 143
Figure 187 – Example Data Output plots: FD, FFS Progression, along with SoH, RUL, and PH 144

3850 W. Ina Rd., Suite 200
Tucson, AZ 85741
www.ridgetopgroup.com