# Development of GUI for Database creation, Data conversion and Data analysis for MIL-STD-1553B Data

Author, Alan Israel M

*3rd year integrated MTech in ECE*

*International Institute Of Information Technology, Bangalore*

*Intern at MCSRDC, HAL. ( 10.05.2018 – 21.06.2018 )*

*alan.israel@iiitb.org*

***Abstract*— To create GUI for ICD data handling and analysis for the MIL-STD-1553B protocol using MATLAB Tool. To enable create database, and quick data conversion and plotting for data analysis and report generation. To study the ARINC 708 protocol and analyze the weather radar data transmitted and generate display using MATLAB Tool.**

## INTRODUCTION

The flight data coming from the Mission Computer (transmitted and received from onboard sensors on the aircraft) is recorded in Flight test instrumentation in the form of hexadecimal values. These values need to be converted to engineering values or decimal values in order to enable easy analysis.

Converting hexadecimal values to the engineering format done as per the predefined ICD required. The ICD is a document which specifies the format of the information present in the hexadecimal values in line with the respective data bus protocol. The hex dump contains words received from the bus. The ICD contains the information of the different parameters being encoded in the words.

Therefore, using the ICD format information, the conversion of the hex values to engineering values is being done. This converted data is used for plotting graphs and also analysis of the different parameters of sensors.

The weather radar data received from the ARINC 708 bus is used to represent the precipitation and the turbulence present in the surrounding of the aircraft in the form of video display. The ARINC 708 data was simulated and used to generate a image of the weather radar simulation.

## PROBLEM STATEMENT

To create simple GUI tools for easier and faster analysis of the flight data. GUI has to assist the user in specifying the ICD data format. This format has to be stored in a text or comma separated value (CSV) file. This acts as the database for the raw file containing the hexadecimal values for conversion. On applying the database to the raw file, the engineering formatted values are obtained in a CSV format. This file can be used for plotting and analysis of the different parameters contained in the file.

To study the ARINC 708 protocol and access the card using the API calls. To transmit and receive data through the bus and dump the data in a file for analysis. The data should be used for plotting the precipitation and turbulence around the aircraft.

## RESEARCH

**MIL-STD-1553** is a military standard that defines the electrical and protocol characteristics for a data bus. A data bus is used to provide a medium for the exchange of data and information between various systems. It is similar to what the personal computer and office automation industry has dubbed a Local Area Network (LAN).

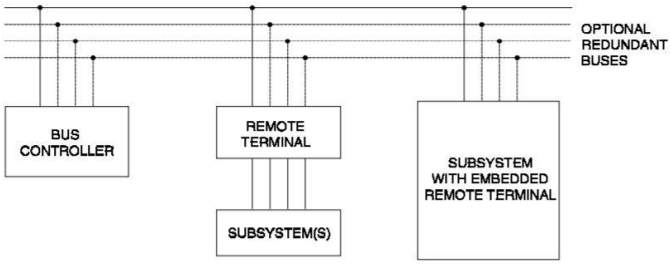
MIL-STD-1553B defines the term Time Division Multiplexing (TDM) as “the transmission of information from several signal sources through one communications system with different signal samples staggered in time to form a composite pulse train.”

MIL-STD-1553 defines certain aspects regarding the design of the data bus system and the black boxes to which the data bus is connected. The standard defines four hardware elements. These are:

- *The transmission media*
- *Remote terminals*
- *Bus controllers*
- *Bus monitors*

The *transmission media*, or data bus, is defined as a twisted shielded pair transmission line consisting of the main bus and a number of stubs. There is one stub for each terminal connected to the bus. The main data bus is terminated at each end with a resistance equal to the cable's characteristic impedance (plus or minus two percent). This termination makes the data bus behave electrically like an infinite transmission line.

*Remote terminals* are defined within the standard as "All terminals not operating as the bus controller or as a bus monitor". Therefore if it is not a controller, monitor, or the main bus or stub, it must be a remote terminal. The remote terminal comprises the electronics necessary to transfer data between the data bus and the subsystem. So what is a subsystem? For 1553 applications, the subsystem is the sender or user of the data being transferred. The figure below shows the different levels of remote terminals.

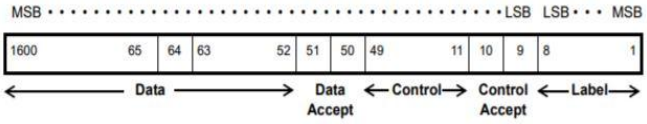



The *bus controller* is responsible for directing the flow of data on the data bus. While several terminals may be capable of performing as the bus controller, only one bus controller may be active at a time. The bus controller is the only one allowed to issue commands onto the data bus. The commands may be for the transfer of data or the control and management of the bus (referred to as mode commands).

A *bus monitor* is a terminal that listens (monitors) to the exchange of information on the data bus. The standard strictly defines how bus monitors may be used, stating that the information obtained by a bus monitor be used "for off-line applications (e.g., flight test recording, maintenance recording or mission analysis) or to provide the back-up bus controller sufficient information to take over as the bus controller." A monitor may collect all the data from the bus or may collect selected data.

**ARINC 708** is a specification for airborne pulse Doppler weather radar systems primarily found on commercial aircraft. The Weather Radar protocol consists of 1600-bit ARINC-708 words, which, when interpreted and plotted together, form a picture of weather patterns in the surrounding area. The various weather conditions are indicated by the colors displayed.

The picture is formed as follows: emanating from the center of a circle, a single radius line at a time is drawn, where each radius line is the data from a single ARINC-708 word. The data portion is organized into 512 range-bits per scan angle value. Each range bit contains a color value to indicate the intensity at that position. The ARINC 708 display data bus word format is as follows:




## RESOURCES AND METHODS

**MATLAB 2012a** was used to create the GUI for the ICD data handling. Also used for plotting the data received from the ARINC 708 bus denoting the weather radar data.

In order to tweak and debug the ARINC 708 device driver code, **Visual Studio** was used. The C codes were compiled and executed in Visual Studio and the 708 card was accessed for transmitting and receiving the data. The debugging and accessing the card was done in the Testing Rig. The Sensor Simulator system was used to experiment the simulation of weather radar data. PATS simulator was also experimented in an attempt to simulate the weather radar data.

## DESCRIPTION OF GUI TOOLS

The ICD Data Handling GUI consists of three modules:

- ICD Database Creation and Editing
- ICD Data conversion to Engineering Format
- ICD Data analysis through Graphs

The above three modules are integrated onto a single main module which enables the user to choose from the above modules accordingly.

*1) ICD Database Creation and Editing*

This GUI allows the user to add different messages and their formats specified in the ICD, in a comma separated format ( .csv ). As the user gives in the entries of different messages, these are being updated in the database and the user can also open the same to view the contents of the database. The fields that need to be entered in the GUI are as follows:

- **MSG_ID** - Message ID, e.g., MSG_1.
- **From** - From which Source, e.g., DMC.
- **To** - To which Destination, e.g., FTI.
- **Parameter_name** - Parameter name, e.g., Track Angle.
- **Start_word** - Word number for conversion.
- **End_word** - Word number for conversion. Same as Start_word if single word is considered for conversion. If not same, then the word extends from Start_word: End_word for conversion, e.g., Start_word = 0xA345, End_word = 0xF325. Final word = 0xA345_F325.
- **Signed** - Indicates whether word is signed or unsigned. 1 – signed, 0 – unsigned.
- **RT_no** - RT number which is transmitting or receiving. Value = ( 0 – 31 ).
- **Tx_SA** - Transmit Sub-Address. Value = ( 0 – 31 ). If not required then indicate with -1.
- **Rx_SA** - Receive Sub-Address. Value = ( 0 – 31 ). If not required then indicate with -1.
- **Unit** – Units of engineering value, e.g., deg ( degree).
- **Start_bit** – From which bit in the word the information is present. Value = ( 0 – 15) for 16-bit word.
- **End_bit** - Till which bit in the word the information is present. Value = ( 0 – 15) for 16-bit word.
- **LSB** – Value of the least significant bit specified in the ICD. E.g., 0.005493 for Track Angle.
- **MSB** - Value of the most significant bit specified in the ICD. E.g., 180 for Track Angle.

The components used in the GUI are:

- *Create New* File Button – This allows the user to create a new database file by specifying the name of the file in the given text field.
- *Open Existing File* Button – Allows the user to open any existing database and view its contents. User can also add new records or edit existing records.
- *Prev* and *Next* Button – Allows the user to navigate through all records in the database file when open.
- *Add Record* Button – To append new record in the database file which is open.
- *Save Record* Button – To save the record in the database file.
- *Close File* Button – To close the file open for either creating new file or opening existing file.
- *View* Radio Button – To enable Read Only mode on opening existing database.
- *Edit* Radio Button – To enable Read/Write mode on opening existing database.

**Note** : On creating a new file, user can only create new records. The user cannot view the records he has added. If the user wants to add a message whose ID is '8-0-1-0' where the first digit indicates the RT number, the second digit indicates whether transmit/receive, the third bit indicates sub-address, the fourth indicates word count, the user must specify Tx_SA as -1 because the second digit is 0 which indicates receive mode. The user must specify the correct LSB value as this is considered for the engineering value conversion. On filling all entries, the user must press the Save Button to write the record in the database. Only the Save button should be used for saving the records in the new database. After finishing the entries, the user should not forget to close the file as it may lead to error in file operations.

On opening existing file, the user can view all the records available in the database, and also edit them by enabling the Edit radio button. Once the user has edited, the Save button has to be pressed in order to update in the database. The buttons enabled during the Open mode are – Add Record button, Prev/Next button, Save Button. User can add new record by clicking on the Add New Record button which will append the new record to the end of file. After which the user has to hit the Save button to update the database.

*2) ICD Data conversion to Engineering Format*

This module converts the hexadecimal data in the raw file to engineering values by taking input the raw file and the database file. The user has to choose the raw file and the database file which he wants to apply to the raw data. The different message IDs available in the raw data will be displayed in the drop down menu. The user has to choose the message ID which he wants to analyze, and

hit the Convert button. A new file with the same name as that of the message ID is created and it contains all the engineering values pertaining to that message ID.

This module reads from the database, the RT number, the Tx/Rx sub-address and other credentials related to the word and parameter, and applies it on the columns of the raw file. The file generated can be used for analysis of the flight data.

**Note**: The database has to be in the format as created using the above Database Creation GUI. The conversion GUI does not support any other form of database. If the message selected to be written is not present in the database, then an empty file of that message is created.

Also note that the hexadecimal values containing 'E' is being written as exponent values in the csv file. The user must ensure such fields are replaced with valid hexadecimal values.

*3) ICD Data analysis through Graphs*

This module enables the user to load the engineering file for plotting and analyzing different parameters with the help of graphs. The parameters list is loaded in a drop-down menu. The user is given the option of choosing the axes parameters. The user can also write mathematical equations of different parameters and plot the graph. 3-D plotting is also enabled. There are three different types of plots the user can view. Line, Scatter and 3-D plot. Buttons for these are provided and the user can plot graphs by a simple press of the button.

**Note**: A few instructions are written as static text fields on the GUI window. The user must follow these to avoid any possible errors. The user must ensure to that 2-D plot and 3-D plots do not overlap each other. Whenever there is a switch between both types of plots, the Clear button has to clicked. In order to Save the plots, the user must right-click on an empty space on the graph. This will save the plot in a word document.

The above three modules were integrated onto a single module which call these three modules on clicking their respective buttons. The figures of the GUI windows are attached in the ***Appendix***.

The GUI modules are packaged with the MATLAB Compile Runtime Installer ( MCRInstaller ) and is ready to be deployed on any PC with or without MATLAB. The user has to copy the package and execute it. On execution the user must follow the steps to install the MCRInstaller on the PC and the standalone application (executable) is ready to be used.

## ISSUES FACED

The 2012a version of MATLAB does not support table operations. This was an issue since file operations had to be used to read and write into file. On improper handling of file, problems were caused in reading and writing. Automatic filling of database entries is not enabled. The GUI textfield callback function was not executed as expected and the user is expected to enter all fields in the database record entries.

Any entry with the hexadecimal value of 'E' in the csv file was considered to be an exponent value. These values are to be replaced with some other hexadecimal values in order for proper reading of the raw file. For example, the value **'5E03'** corresponds to **'5.00E+03'**. This is not being read by the GUI. But a value '5E0A' is being accepted. So the user has to ensure that such exponent values are removed and replaced with valid hexadecimal values.

## RESULTS

The GUI tool was successfully created and packaged to convert to executable. With the help of MATLAB compile runtime executable, the package was built and a standalone Windows application was built. The user can now use the application across different PCs by copying the GUI package and executing it.

On experimenting with ARINC 708 bus protocol, simulated data flow between transmitter and receiver was achieved by accessing the card through API calls. The received data was dumped into a file and was analysed using polar plots.

## REFERENCES


*1) MATLAB Documentation*

*2) MIL-1553 ICD Data*

*3) ARINC 708 Documentation*

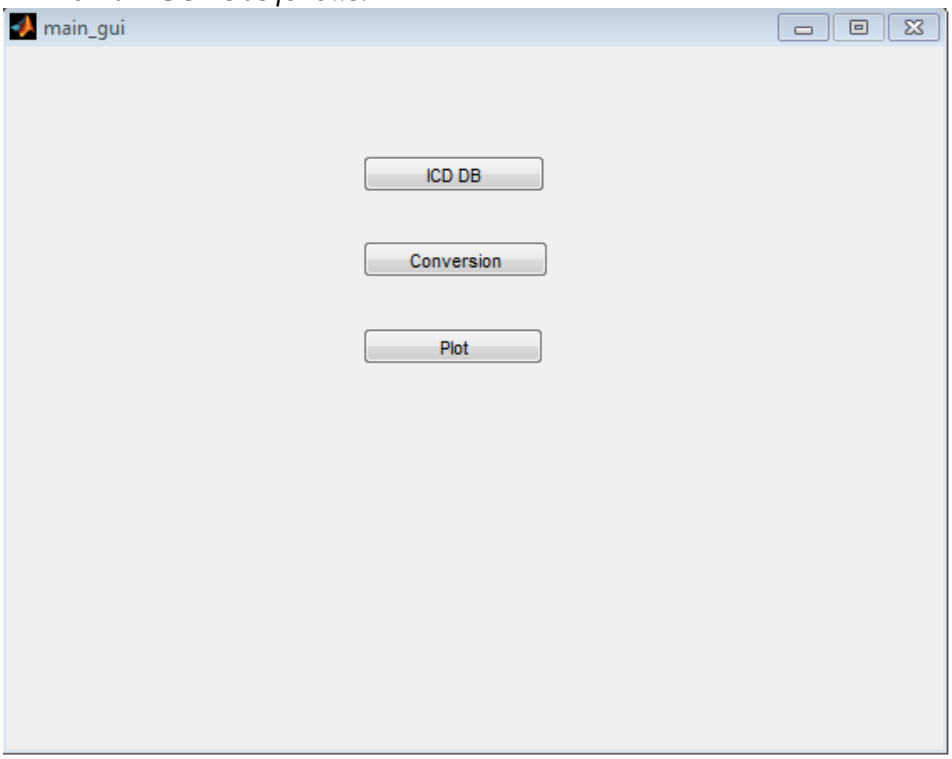
*4) Wikipedia*


## ACKNOWLEDGEMENT

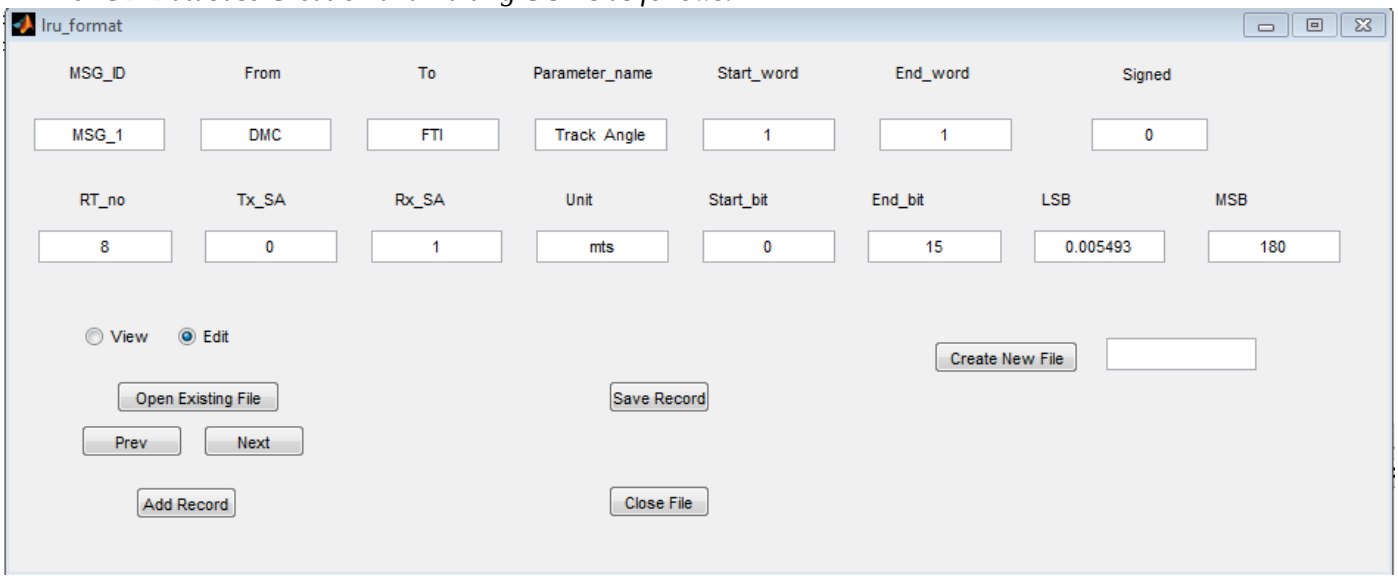

I thank Mr. R. Gopalakrishnan, Senior Manager (Design) for giving me the opportunity to work on this project and his able guidance, constant support and motivation.

## APPENDIX

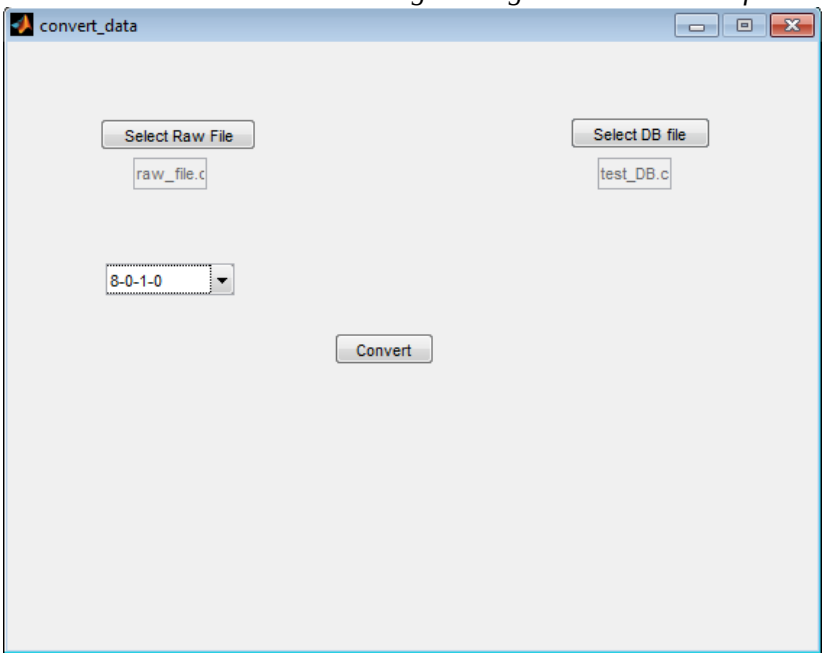
*The main GUI is as follows:*



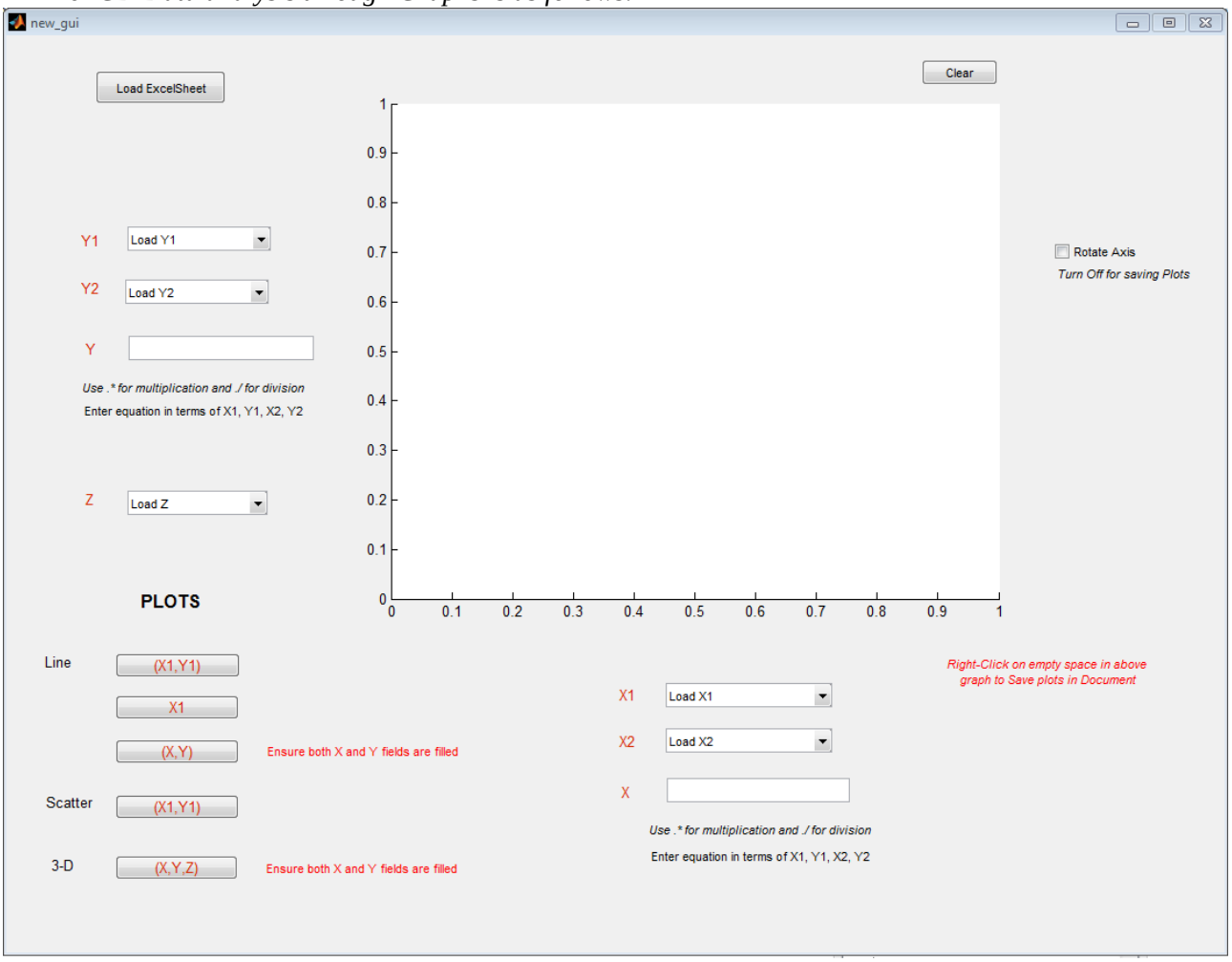

*The ICD Database Creation and Editing GUI is as follows:*



*The ICD Data conversion to Engineering Format GUI is as follows:*

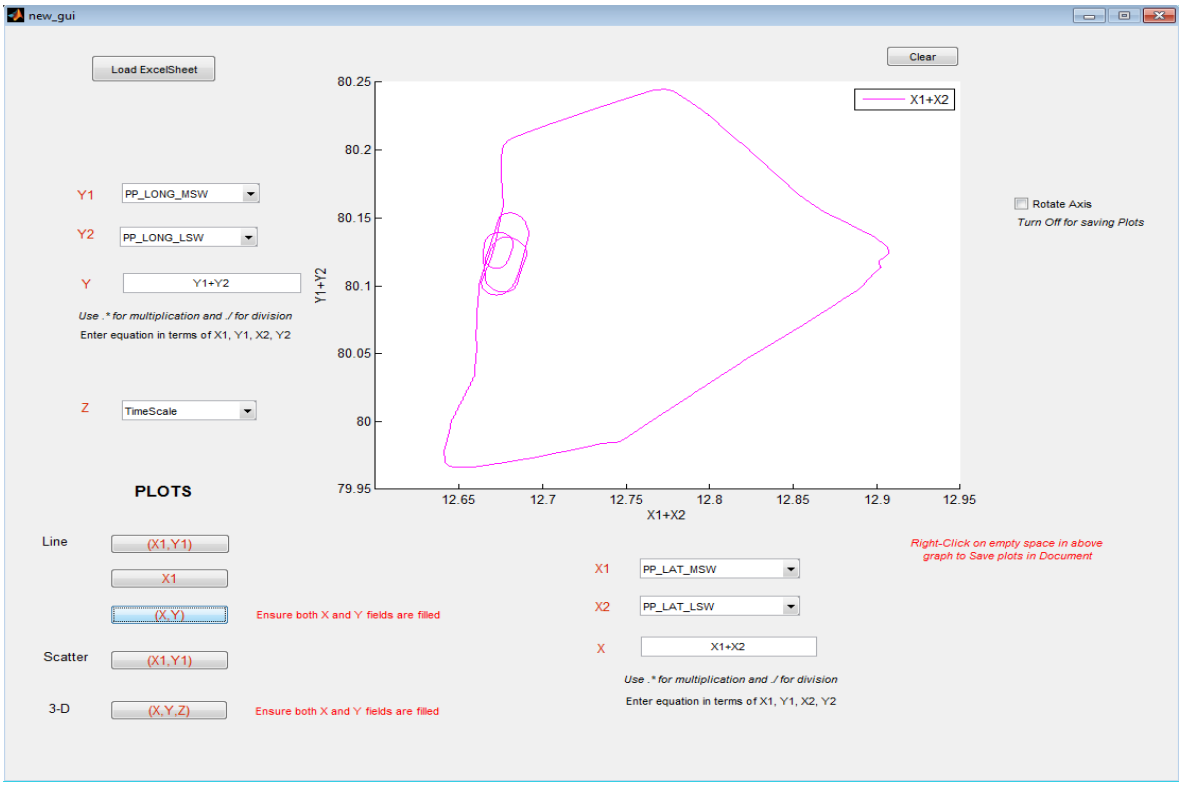



*The ICD Data analysis through Graphs is as follows:*

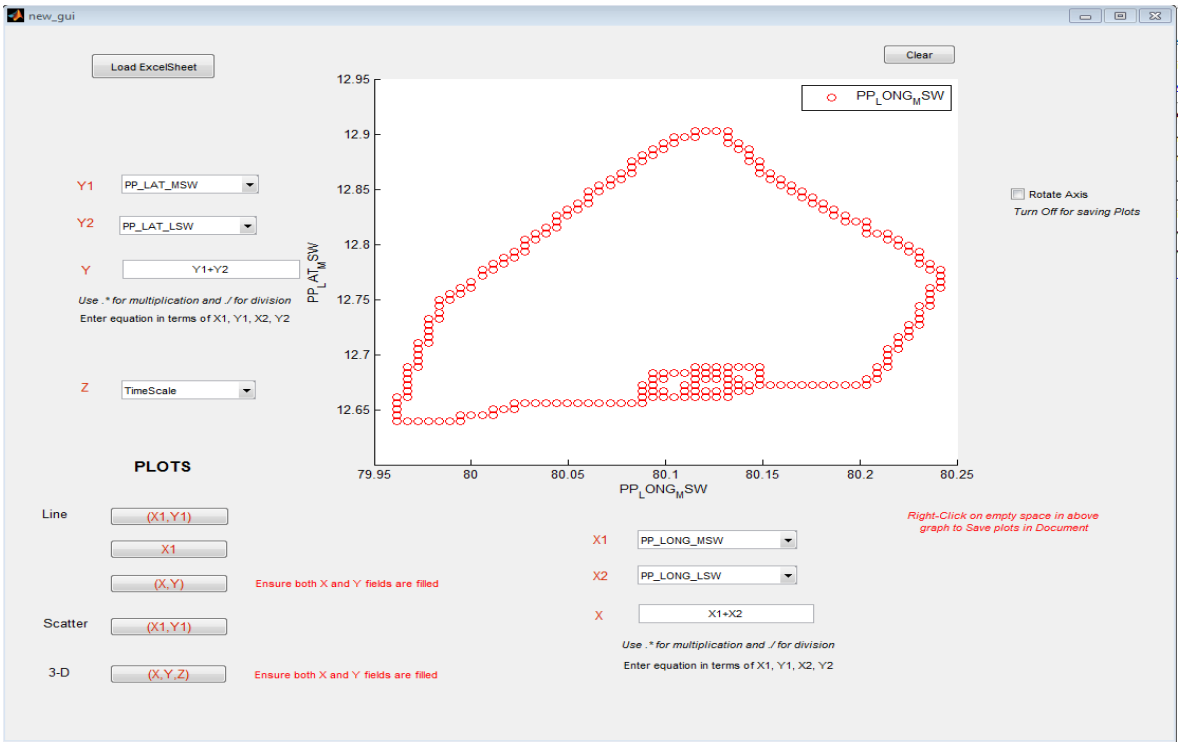


Sample Plots from the Data Analysis Tool

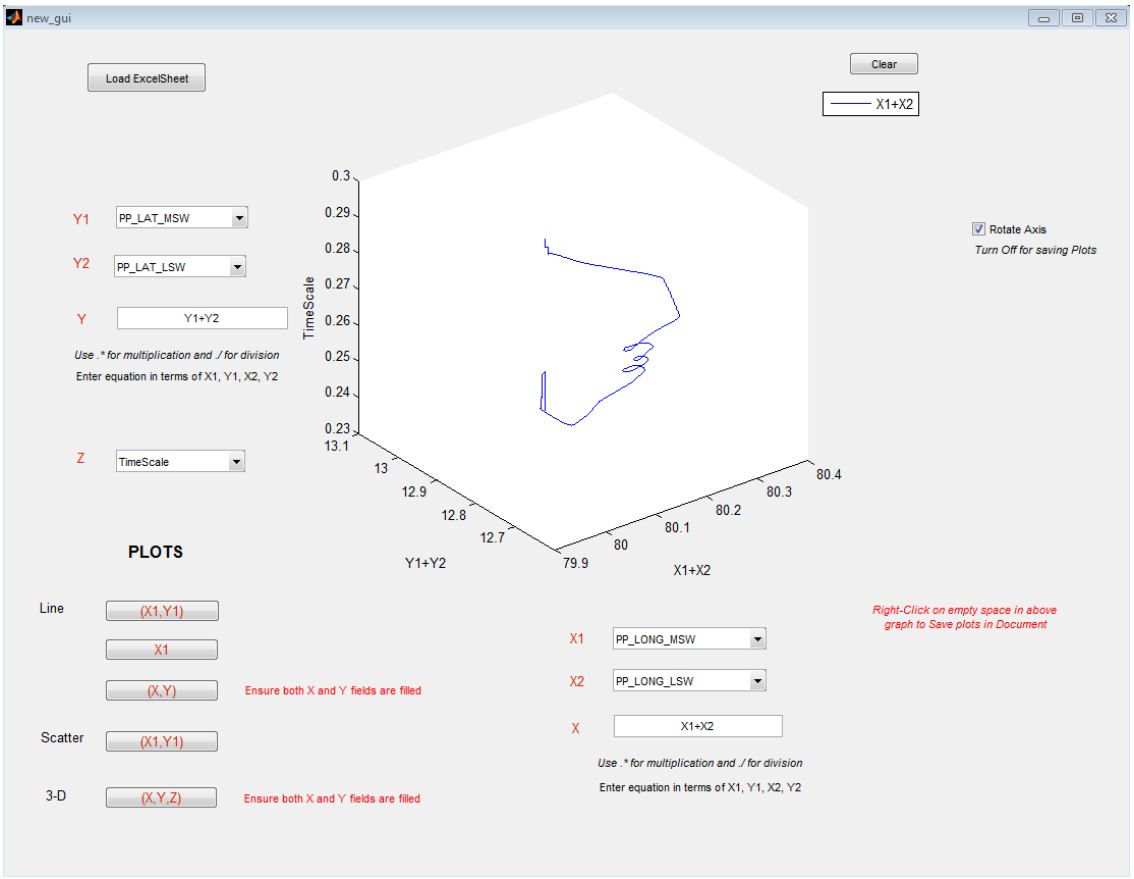
*Line Plot*




*Scatter Plot*



*3-D Plot*


new_gui
Load ExcelSheet
Clear
X1+X2
Y1
PP_LAT_MSW
Y2
PP_LAT_LSW
Y
Y1+Y2
Use .* for multiplication and ./ for division
Enter equation in terms of X1, Y1, X2, Y2
Z
TimeScale
Rotate Axis
Turn Off for saving Plots
TimeScale
0.3
0.29
0.28
0.27
0.26
0.25
0.24
0.23
13.1
13
12.9
12.8
12.7
Y1+Y2
79.9
80
80.1
80.2
80.3
80.4
X1+X2
PLOTS
Line
(X1,Y1)
X1
(X,Y)
Ensure both X and Y fields are filled
Scatter
(X1,Y1)
3-D
(X,Y,Z)
Ensure both X and Y fields are filled
X1
PP_LONG_MSW
X2
PP_LONG_LSW
X
X1+X2
Use .* for multiplication and ./ for division
Enter equation in terms of X1, Y1, X2, Y2
Right-Click on empty space in above graph to Save plots in Document