Matlab-based GUI for Data Analysis of Real-time RFI Filtering System

Student project

By
Aishwarya Dhaigude
Ankita Desai

Electronics And Telecommunication Engineering
Vishwakarma Institute of Technology
Pune

Under the guidance of

Mr. Kaushal Buch

GIANT METREWAVE RADIO TELESCOPE

NATIONAL CENTRE FOR RADIO ASTROPHYSICS
TATA INSTITUTE OF FUNDAMENTAL RESEARCH
Narayangoan,Tal-Junnar,Dist-Pune,
State-Maharashtra PIN - 410 504
INDIA

JUNE - JULY 2017

Abstract

Radio frequency interference(RFI) adversely affects the radio astronomical signals. It thus becomes necessary to detect and filter RFI. In order to verify the RFI detection and mitigation technique,a GUI has been developed. This GUI provides the necessary plots for analysis of data. GUI based data analysis makes the process faster and also allows the user who has little to do with coding to observe and study the plots. The GUI has been designed for raw voltage, beam as well as interferometry data observation and helps in analyzing these data for real time RFI filtering. It is currently available as part of the GMRT Wideband Backend(GWB).

Acknowledgements

We would like express our special gratitude to **Kaushal Buch, Engineer E**, for his patient guidance, enthusiastic encouragement and useful critiques. His passion and pursuit for excellence has inspired us to work harder.

We are grateful to Group Coordinator (Digital Back-End) **Ajit Kumar B , Engineer F** , for giving us the opportunity to work in the digital backend group and for his motivation and suggestions during the project work.

We would like to thank **Prof. Yashwant Gupta** for giving us an opportunity to work under the scientific staff of GMRT

We express our sincere appreciation to **Kishor Naik** for his valuable and constructive suggestions during the planning and development of this project. His willingness to give time so generously has been very much appreciated.

We are grateful to **Sanjay Kudale**, for providing us with his C codes required for the analysis of beam data.

We would like to extend our sincere thanks to **Digital Backend group and the entire staff of GMRT**. It would not have been possible without the kind support and help of many individuals and departments.

Sincerely,

Aishwarya Dhaigude

Ankita Desai

Contents

List of Figures

1 Introduction

1.1 Overview of GMRT

Figure 1: Antennas at Khodad pointed towards a source

The Giant Metrewave Radio Telescope (GMRT) is located at a site about 80 km north of Pune in India. This site was selected after an extensive search in many parts of India, considering several important criteria such as low man-made radio noise, availability of good communication, vicinity of industrial, educational and other infrastructure and,a geographical latitude sufficiently north of the geomagnetic equator in order to have a reasonably quiet ionosphere and yet be able to observe a good part of the southern sky as well. GMRT consists of 30 fully steerable gigantic parabolic dishes. Each antenna is 45 m in diameter, and has been designed to operate at a range of frequencies from 50 MHz to 1450 MHz. Figure 1 shows some of the antennas at GMRT pointed towards a source. It is operated by the National Centre for Radio Astrophysics, a part of the Tata Institute of Fundamental Research, Mumbai. At the time it was built, it was the world's largest interferometric array offering a baseline of up to 25 kilometres (16 miles).

The antennas are all connected together by optical fibre, and operate in unison, to produce images with a resolution of a telescope 25km across. Fourteen of the thirty dishes are located more or less randomly in a compact Central Array in a region of about 1 sq km. The remaining sixteen dishes are spread out along the 3 arms of an approximately 'Y'-Shaped configuration (as shown in figure 2) over a much larger region, with the longest interferometric baseline of about 25 km. The antennas have been constructed using a novel technique named SMART (**S**tretch **M**esh **A**ttached to **R**ope **T**russes) and their reflecting surface consists of panels of wire mesh. These panels are attached to rope trusses, and by appropriate tensioning of the wires used for attachment the desired parabolic shape is achieved. This design has very low wind loading, as well as a very low total weight for each antenna. Consequently it was possible to build the entire array very economically.[1]

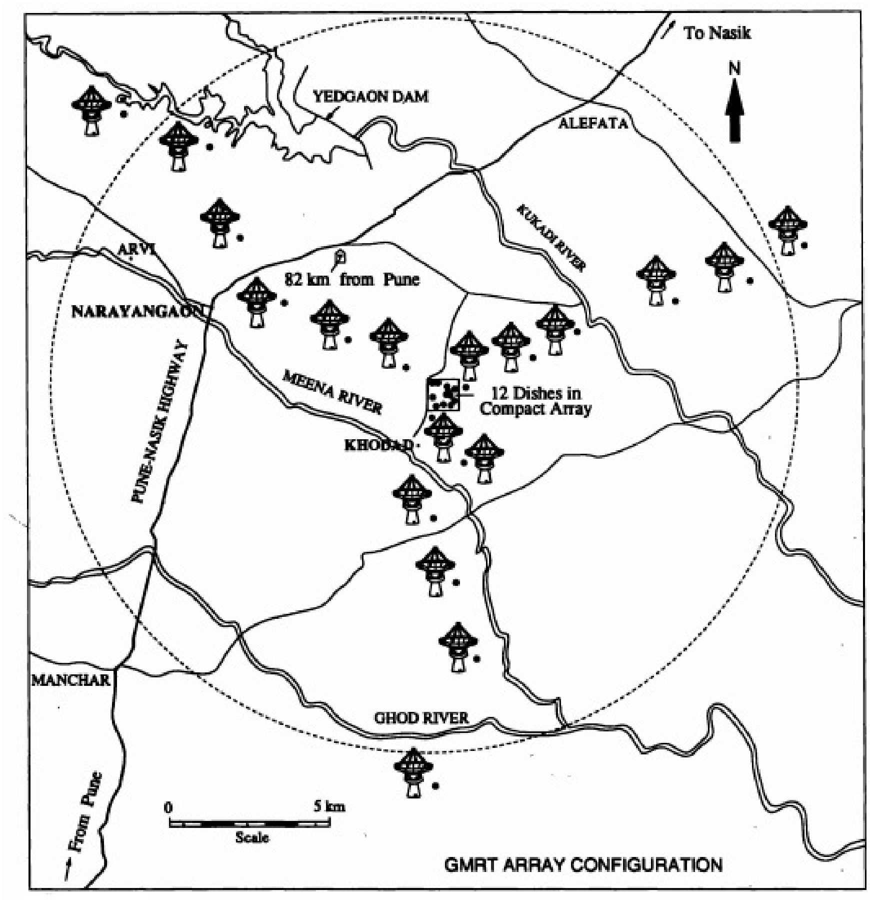

Figure 2: Y-shaped configuration of the GMRT array[1]

1.2 Overview of RFI

1.2.1 What is RFI?

Radio-frequency interference (RFI) also called, Electromagnetic interference (EMI) when in the radio frequency spectrum, is a disturbance generated by an external source that affects an electrical circuit by electromagnetic induction, electrostatic coupling, or conduction.The disturbance may degrade the performance of the circuit or even stop it from functioning. Both man-made and natural sources generate changing electrical currents and voltages that can cause EMI: automobile ignition systems, mobile phones, thunderstorms, the Sun, and the Northern Lights. EMI frequently affects AM radios. It can also affect mobile phones, FM radios, and televisions, as well as observations for radio astronomy.[2]

1.2.2 RFI in Radio Astronomy

Interference in radio astronomy is commonly referred to as radio-frequency interference (RFI).It is basically, any source of transmission that is within the observed frequency band other than the celestial sources themselves. Because transmitters on and around the Earth can be many times stronger than the astronomical signal of interest, RFI is a major concern for performing radio astronomy. Natural sources of interference such as lightning and the Sun are also often referred to as RFI.

Some of the frequency bands which are very important for radio astronomy such as the 21-cm HI line at 1420 MHz are protected by regulation. This is called spectrum management. Because of the limited spectral space at radio frequencies, these frequency bands cannot be completely allocated to radio astronomy. Therefore, observatories need to deal with RFI in their observations.

1.2.3 RFI Filtering

Techniques to deal with RFI range from filters in hardware to advanced algorithms in software. One way to deal with strong transmitters is to filter out the frequency of the source completely. However, filtering out a frequency band implies that these frequencies can never be observed with the instrument.

A common technique to deal with RFI within the observed frequency bandwidth, is to employ RFI detection techniques in the digital system (hardware or software). Such a technique can find samples in time, frequency or time-frequency space that are contaminated by an interfering source. These samples are subsequently ignored in further analysis of the observed data. This process is often referred to as data flagging.

The spectrum observed by a radio telescope may be either collocated or shared with various other communication services which can severely affect the observations. To mitigate this source of RFI, the radio telescope site is usually built in an area which has sparse population density and is devoid of industrial development. The majority of RFI that is found in such remote sites is due to extra-terrestrial sources or radio propagation through the ionosphere and troposcatter from remotely located transmitters.

To prevent pollution of the radio spectrum, a radio quiet zone spanning several tens of kilometres in diameter may be established around the radio telescope in coordination with the domestic spectrum allocation authorities. The spectrum in and around the observatory is regularly monitored by the observatory authorities to check the power levels at the frequency bands of interest. Construction of future radio telescopes is planned in remote geographical regions either in the deserts or on the mountains. This provides an environment free from terrestrial RFI, enabling the radio telescopes to achieve their expected sensitivity.To prevent observatory-generated interference, the building needs to have appropriate shielding. The forgoing are the initial steps towards RFI mitigation in a radio telescope. Advanced signal processing techniques have been proposed and used for locating sources of RFI around the observatory. Over time, due to urbanization and industrialization of the once remote observatory site, the radio observatory starts facing increasing levels of RFI, which is the situation with most existing radio telescopes. Where regulatory efforts fail to provide adequate protection against RFI, it becomes necessary to implement techniques for RFI mitigation at various stages in the receiver chain and in the post-processing of data.[3]

Figure 3 shows the block diagram of Real time RFI filtering.

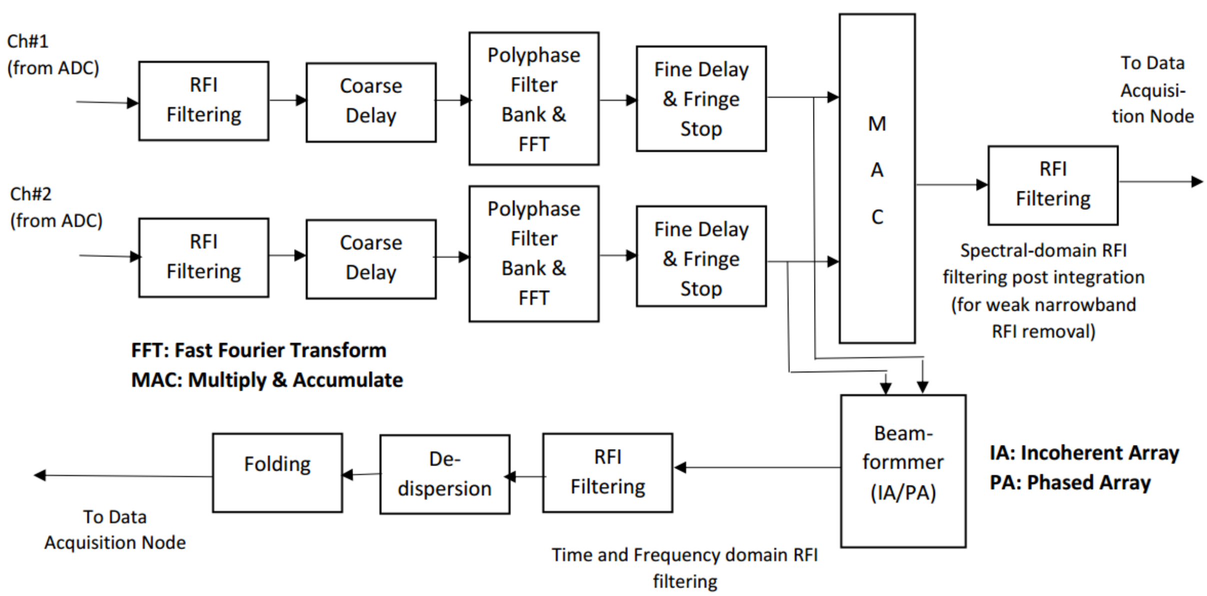

Figure 3: Block diagram of Real time RFI filtering[4]

1.2.4 Real time RFI Mitigation technique for the GMRT Wideband Backend (GWB)

Of the numerous RFI mitigation techniques, GMRT Wideband Backend(GWB) uses RFI excision. In this technique, it is assumed that the power in RFI signals is much greater than the actual astronomical signal. This assumption mostly holds true as most of the RFI sources are terrestrial and hence have large power. Time domain excision using robust estimates of variance is used for mitigating impulsive broadband RFI and has been implemented using the Median Absolute Deviation(MAD) estimator on an FPGA for real-time removal of power-line RFI in the Giant Metrewave Radio Telescopes(GMRT) wideband backend. Excision of time-domain RFI resulting from high voltage power-line transmission lines has also been effectively demonstrated using the MAD estimator in the offline mode of the GMRT software backend(GSB) implemented using a CPU cluster. This robust estimation technique is effective when the RFI is much stronger than the astronomical signal. Upon computation of the excision threshold through a robust estimator, the RFI outside this specified threshold is selectively removed (or replaced with a particular value or noise).

The basic classification of RFI mitigation techniques is as show in figure 4

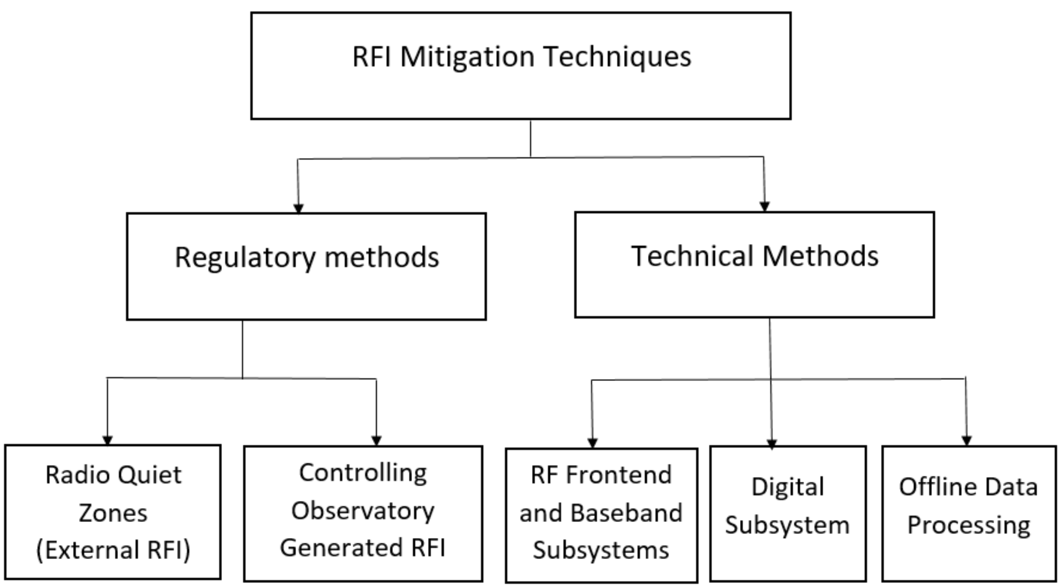

Figure 4: Basic Classification of RFI Mitigation Techniques[3]

2 Motivation for developing the GUI

Monitoring the RFI helps in locating and removing of RFI sources. RFI can be avoided if it is well characterised in terms of direction and frequency. However a good knowledge of the properties of RFI will help in RFI detection and removal from the collected data using RFI filtering techniques.

At the beginning of this project the observatory had many tools which were only used for visualization purpose but now with the help of this tool the observatory can visualize and analyse the signals.

This tool helps in doing the regular data analysis. The diagnostic tool GUI helps the user in understanding the RFI very closely and know its properties.

For the observatory the tool can be used for long term RFI characterization and filtering strategy. This tool is also useful for the astronomer observing from the GMRT to have a look at few things like how the technique performs on the given day and frequency band before taking observations. This helps the astronomers in deciding the filtering options during the observation.

3 Requirements for GUI and overall layout

The GUI is developed in MATLAB and is compatible with both windows and Linux operating system. This section gives an idea about the system flow of all the GUIs.

3.1 Design flow for Main GUI

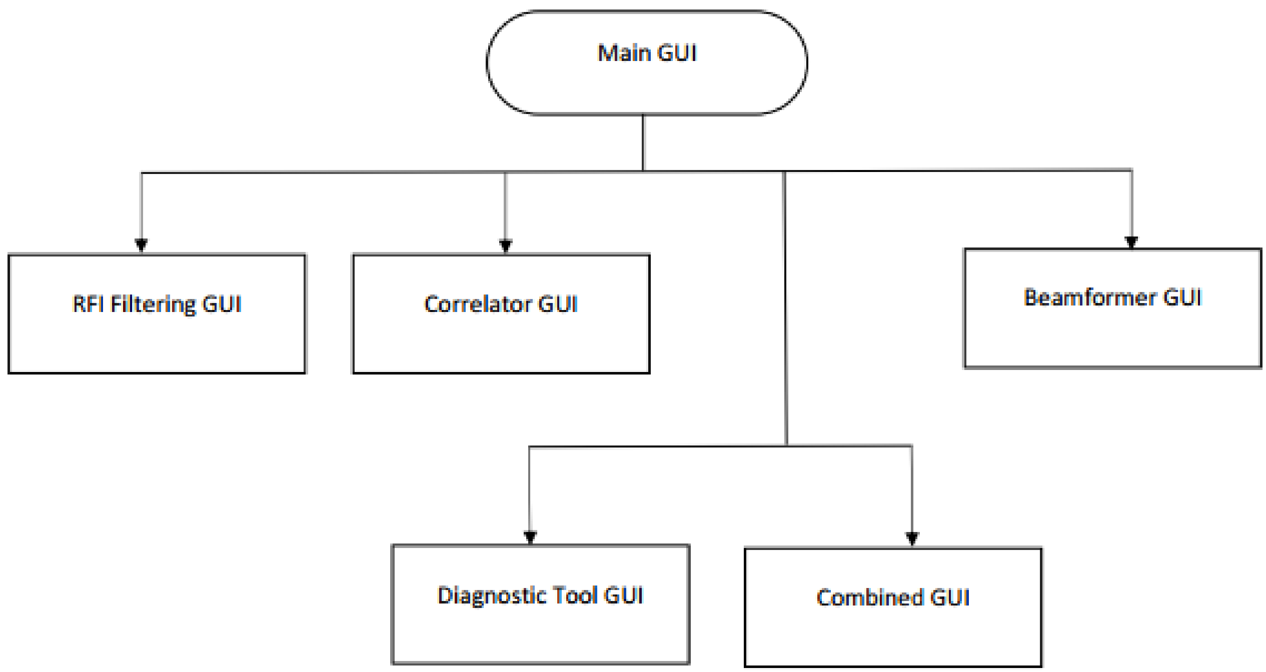

Figure 5: Design flow of Main GUI

3.2 Design flow for Raw Voltage GUI

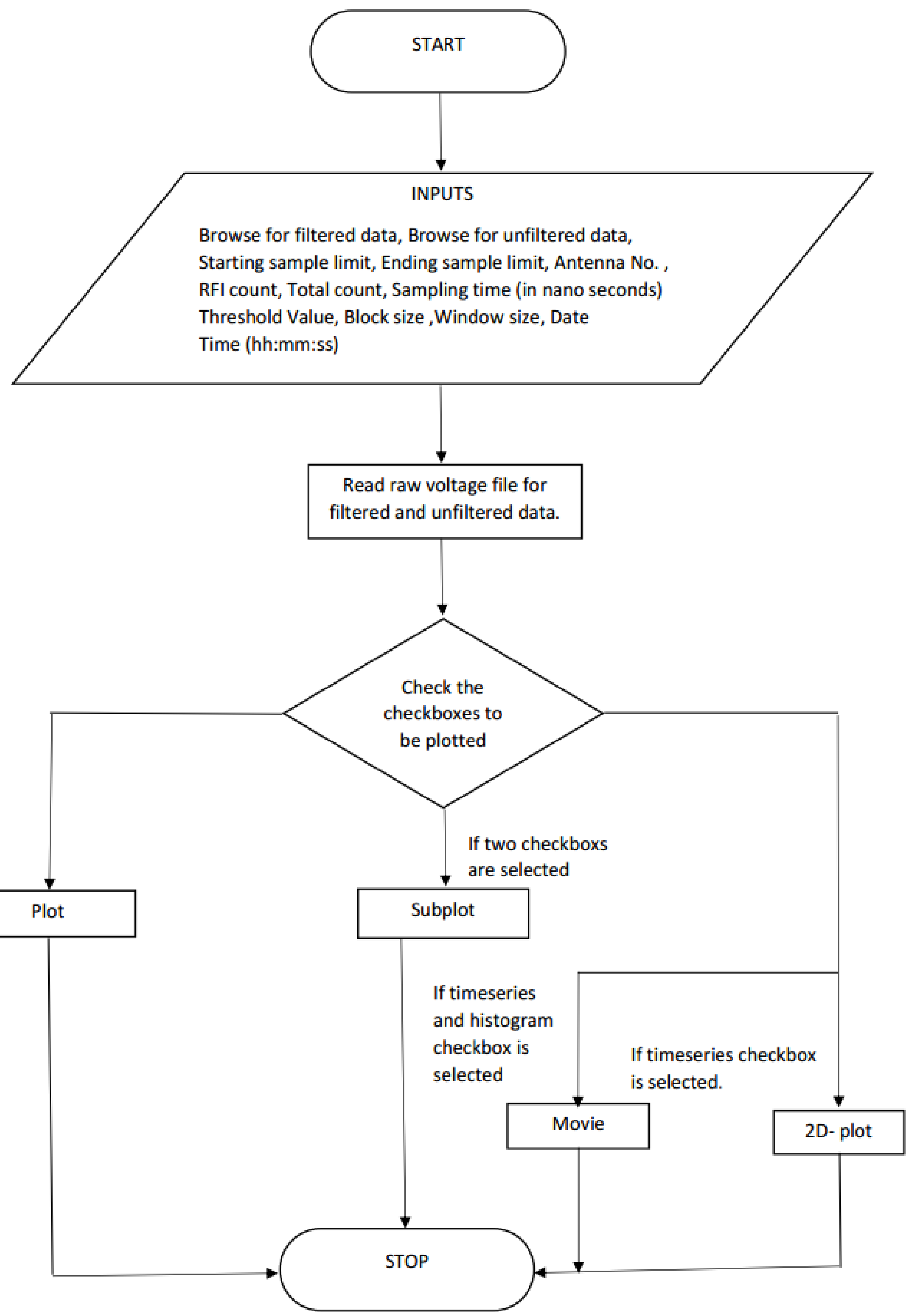

Figure 6: Flow chart of raw voltage GUI

3.3 Design flow for Diagnostic Tool GUI

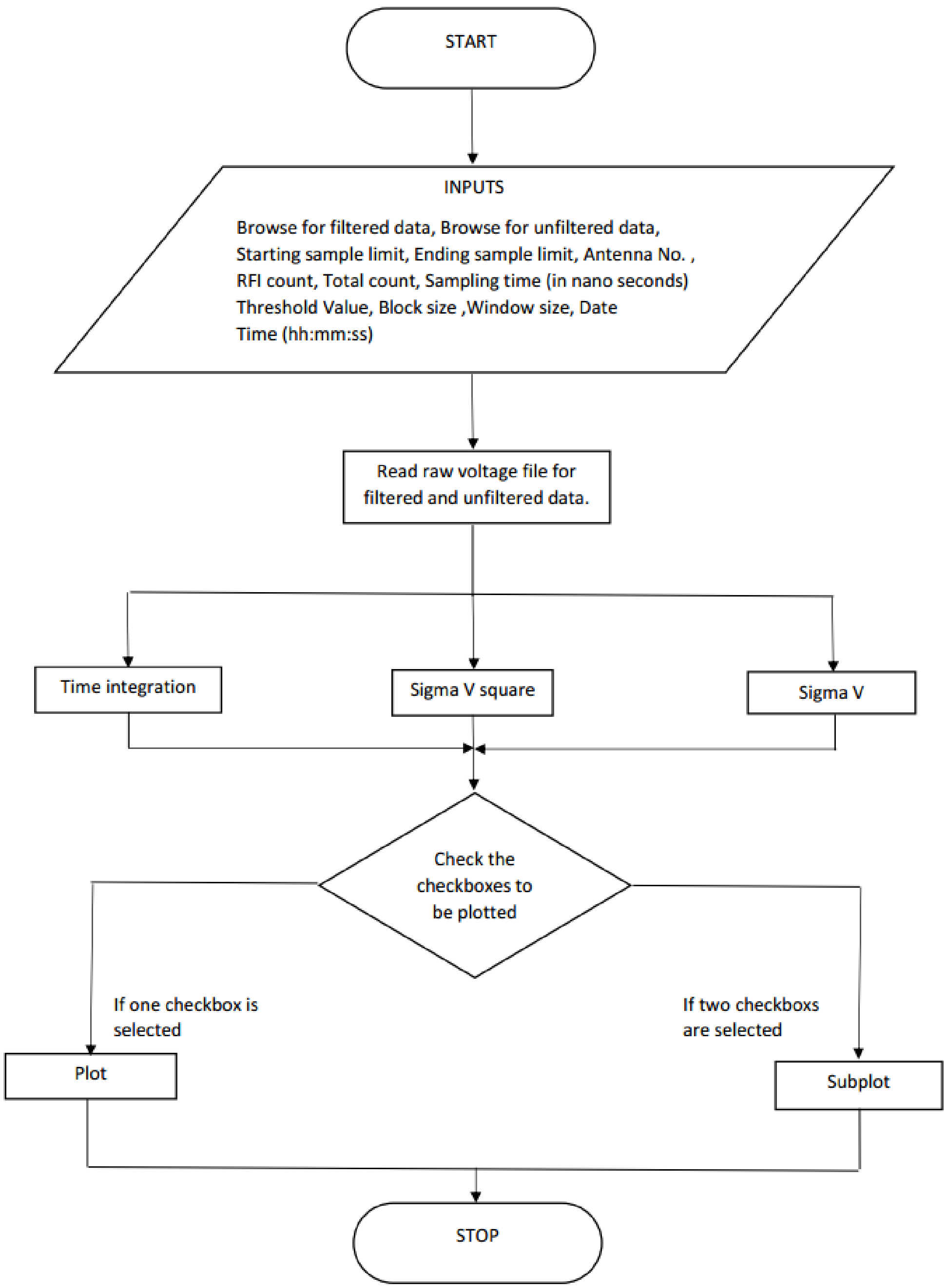

Figure 7: Flow chart of Diagnostic Tool GUI

3.4 Design flow for Beamformer GUI

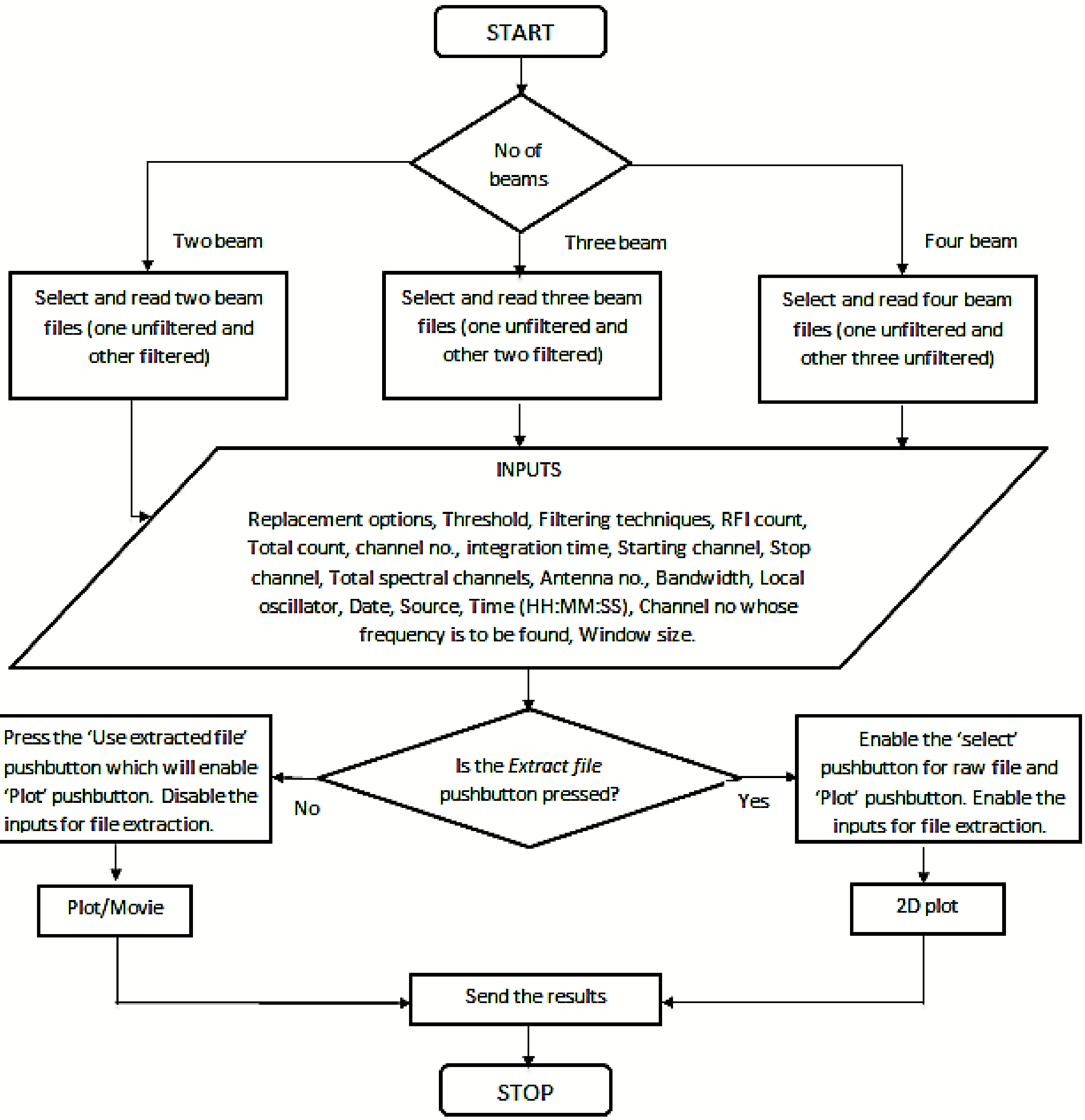

Figure 8: Flow chart of Beamformer GUI

3.5 Design flow for Correlator GUI

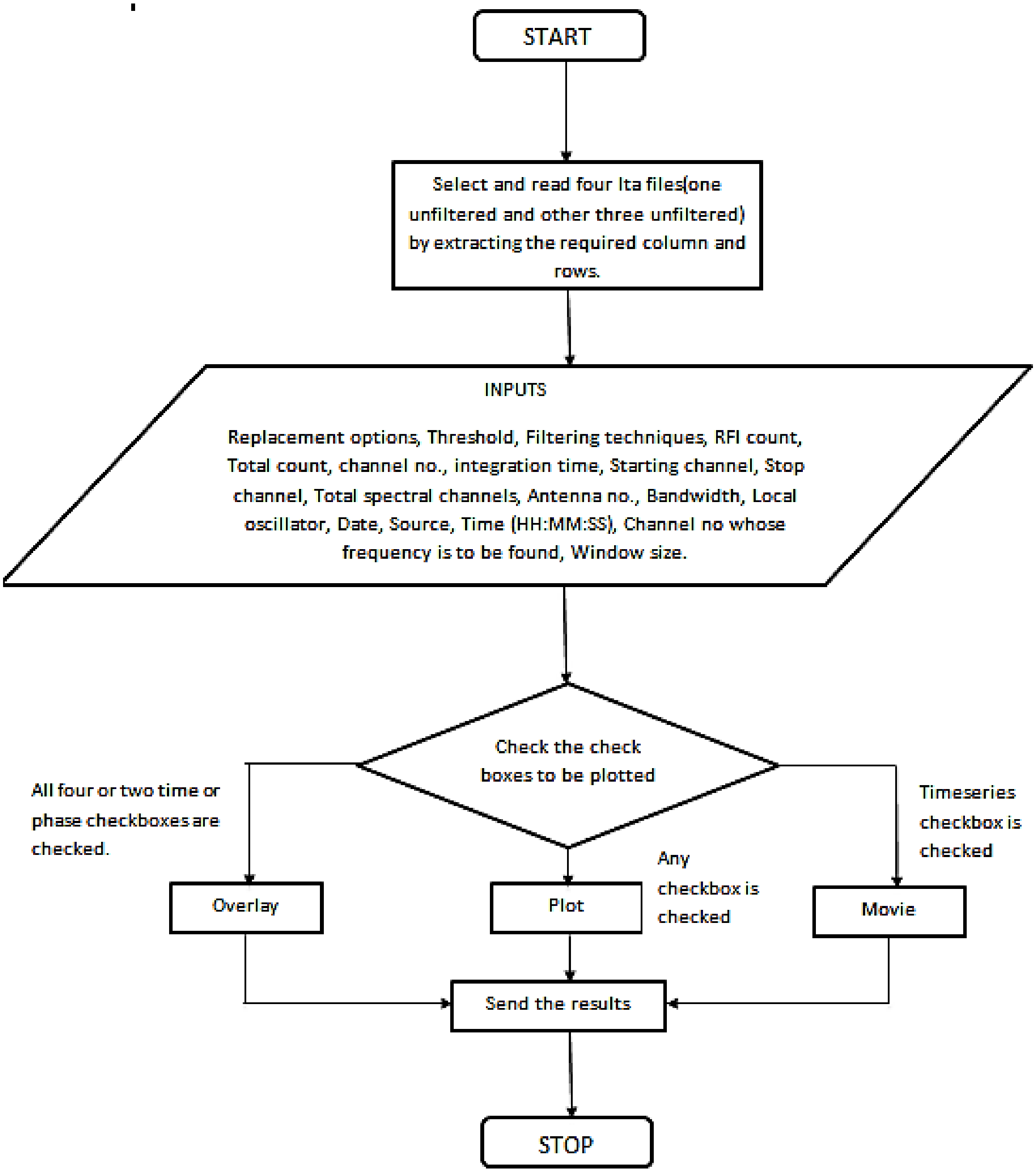

Figure 9: Flow chart of Correlator GUI

3.6 Design flow for Combined GUI

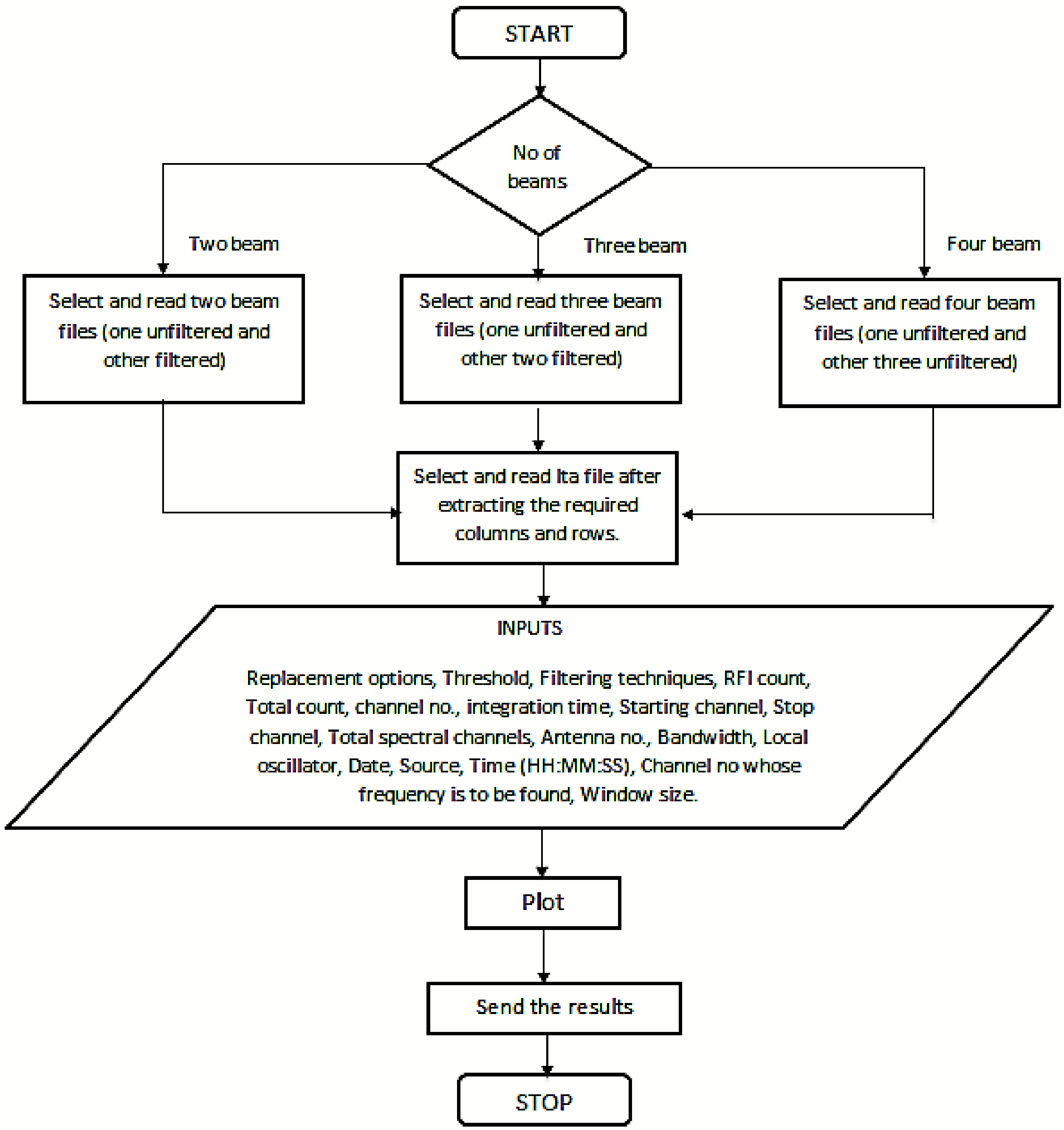

Figure 10: Flow chart of Combined GUI

4 Implementation

This section will give an idea of various features included in the Raw voltage, Diagnostic tool, Beamformer, Correlator and Combined GUI and how they are implemented. Some of the common features in GUIs are also listed in this section.

4.1 Some of the common features in all GUIs

The GUI is handling and processing intensive data files, this means that the GUI is memory intensive. Using *Memory Usage* we can see how much memory the MATLAB is currently using so that if it gets full then the user can clear the memory. Here we are monitoring the Java memory which is a part of overall MATLAB process memory. Representation of the free and used memory is done using a pie chart figure 11.

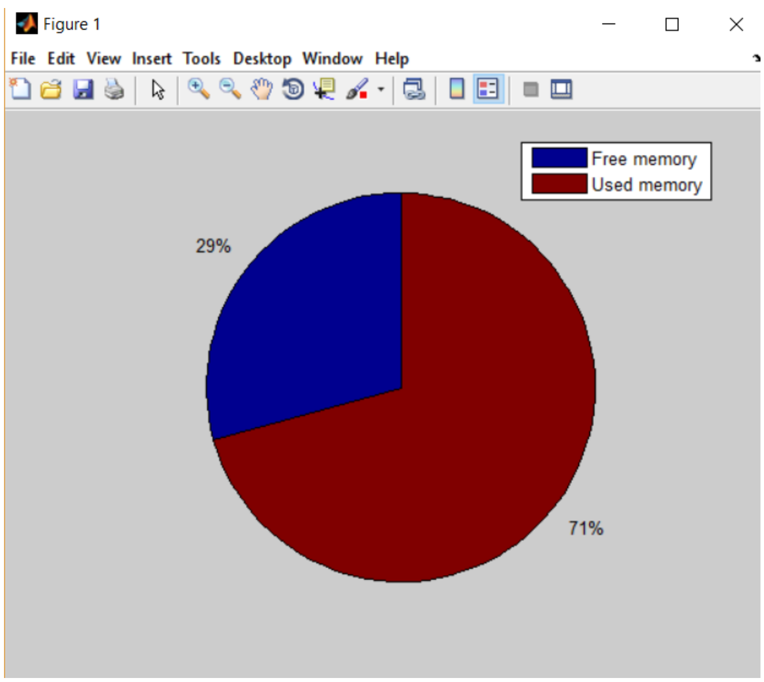

Figure 11: Representation of memory

The *Clear memory* will clear the workspace and command window and also sets the static box containing the filenames to their default text. Once the memory is cleared then an exception pops up saying "Command window and workspace cleared" figure 12.

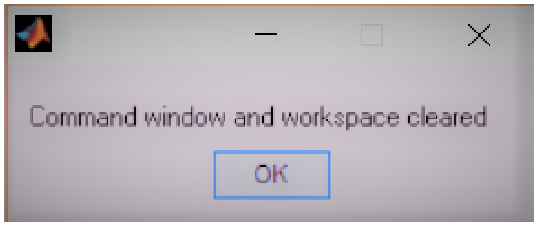

Figure 12: Exception handling for *Clear memory*

The *Help* pops a help window which provides some information about each of the inputs, push buttons and other parameters of the GUI. The callback of this push buttons displays a text file in the Help window as shown in figure 13. Horizontal scrolling option is also provided so that there is no limit for the number of lines in the text file.

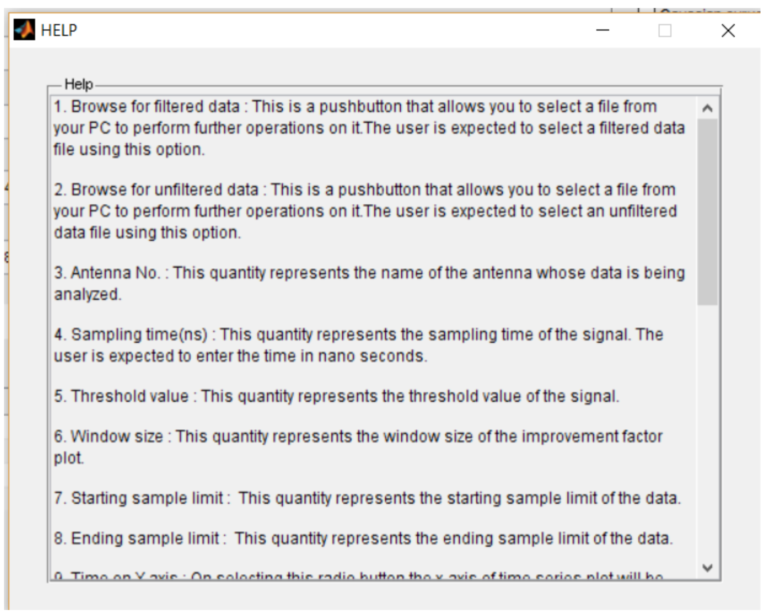

Figure 13: *Help* window

The *Back to main GUI* will call back the main GUI from the current GUI and a new main GUI window will be opened. The user need not have to look for the main GUI window in MATLAB instead click on the push button.

The *Grid* will have the grid lines on the respective plots on its selection.

4.2 Main GUI

The main GUI consists of four push buttons and are linked with another GUI. Out of these four push buttons two bare same name called RFI Filtering and will open a same GUI. So these two push buttons on selection will open a complete

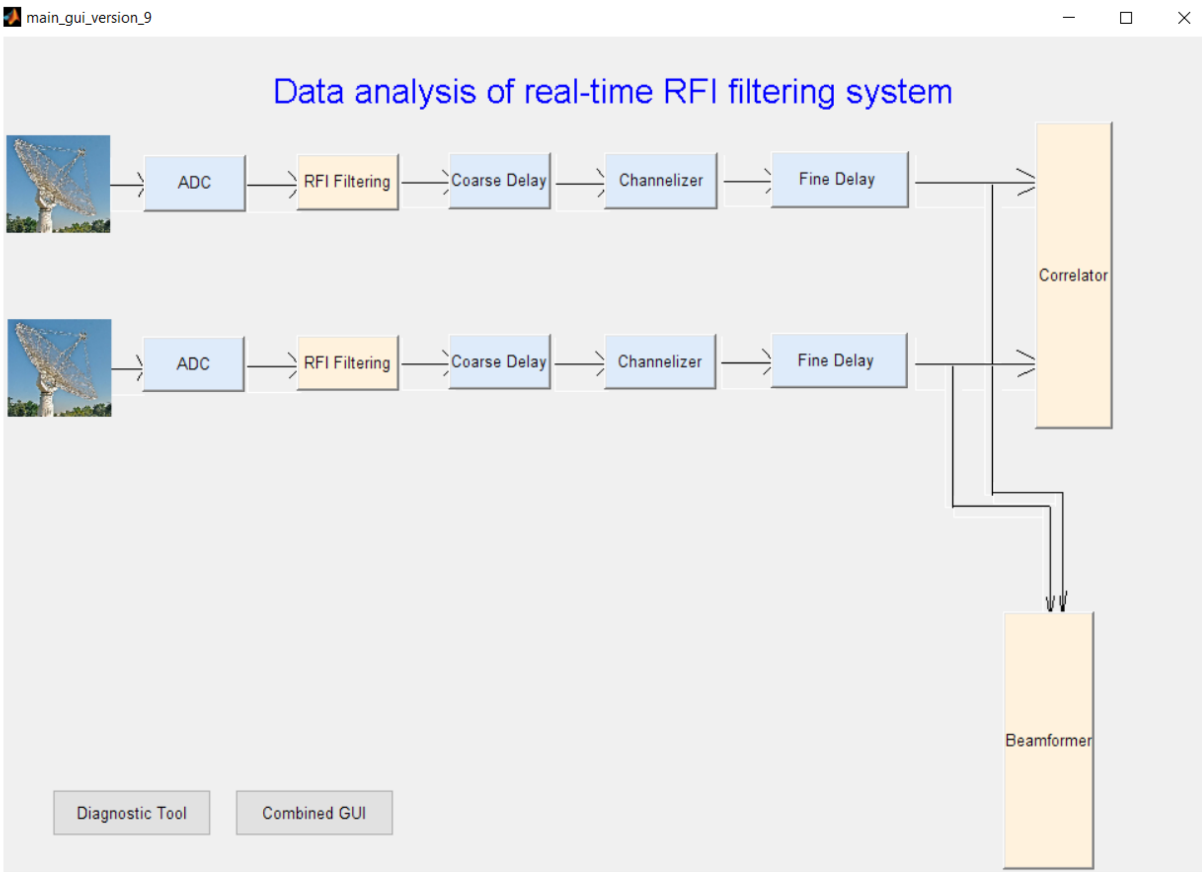

Figure 14: Main GUI

new and independent raw voltage GUI. Similarly the remaining two push buttons named Correlator and Beamformer will open two different and independent correlator and beam data GUI respectively. Figure 14 shows the main GUI.

4.3 Raw Voltage GUI

This GUI is for observation and analysis of the raw voltage data. It is also used to display some statistical parameters like the mean, standard deviation, skewness, kurtosis, median, mean absolute deviation (MAD), and robust standard deviation of the data. The data can be only filtered or only unfiltered or both filtered and unfiltered. Figure 15 shows the GUI for raw voltage.

The pushbuttons named as *Browse for filtered data* and *Browse for unfiltered data* allows the user to browse for the filtered .out file or unfiltered .out file respectively. The callback of both *Browse for filtered data* and *Browse for unfiltered data* displays a dialog box that lists files in the current working directory and enables the user to select or enter the name of file. To enable this process an in-built MATLAB function is used called uigetfile. If the file name is valid that means if the file name exists in the current working directory then the uigetfile

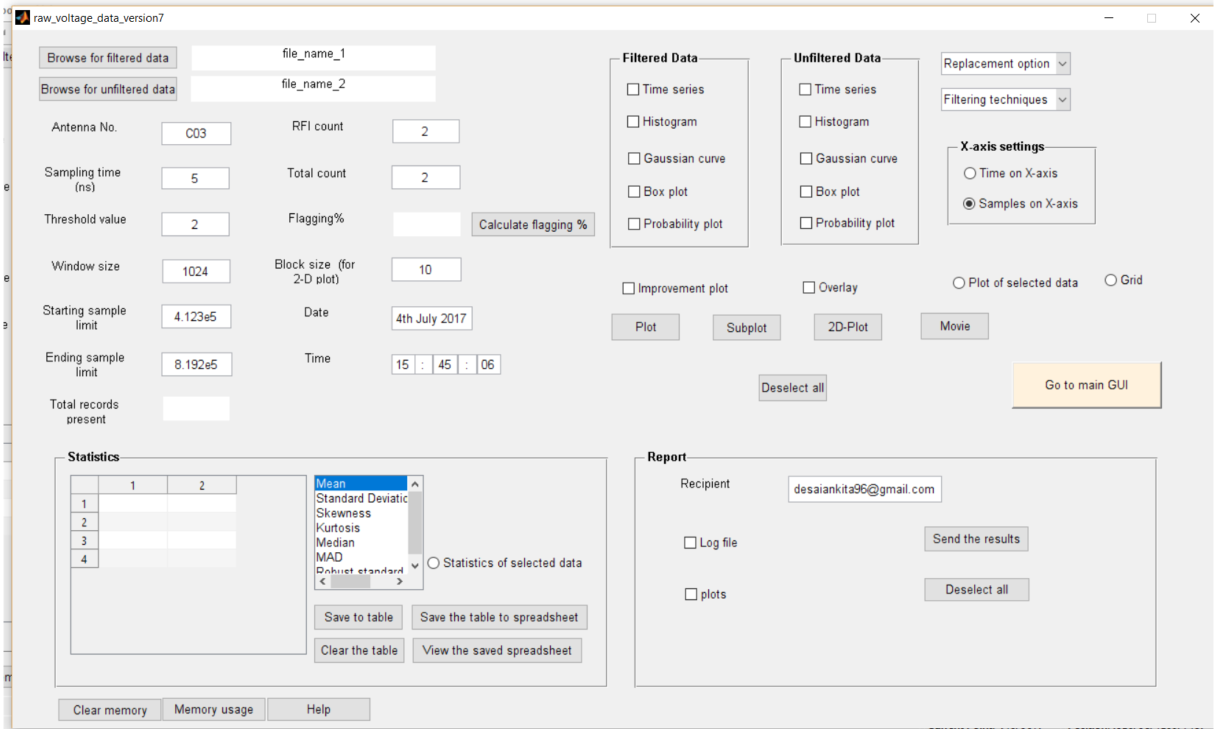

Figure 15: Raw Voltage GUI

returns the file name when the user clicks on *Open* and if the user clicks on *Cancel* or closes the dialog box then the uigetfile will return 0. The callback also sets the static textbox with the filename of the browsed file and also reads the file into a variable called filename.

The inputs such as ‘Antenna No.’, ‘Sampling time (ns)’, ‘Date’and ‘Time’are used in the title of the plots whereas the ’Filtering technique‘, ‘Replacement options’and ‘Threshold value’are specified in the legends. *Total record present* will display the length of raw data file when the filtered file is loaded.

The inputs ’Total count’ and ’RFI count’ is used as an input to calculate the flagging percentage. Flagging is calculated as;

$$Flagging\% = \frac{RFIcount}{Totalcount} \times 100$$

Flagging percentage will be displayed in the static text box only clicking the push button named Calculate flagging%.The inputs Starting sample limit and Ending sample limit are used as an input to analyse the histogram plot, fitting gaussian curve in histogram plot, box plot and probability plot in the user defined limits of data.

The *Plot* pushbutton will display the respective plots of filtered and unfiltered data. So each figure will contain only one plot.The Subplot push button will display the plots of selected checkboxes for filtered and unfiltered data in the same figure but one below the other. The upper plot is of the unfiltered data and the below one is of filtered data.The 2D-Plot push button will display a two dimensional plot of either filtered time series data or unfiltered time series data. In the callback first the data is reshaped using in-built function reshape.The Overlay checkbox will give a plot of overlaid time series plot of filtered data over the time series plot of unfiltered data.The Improvement plot checkbox will give a mean by rms plot. However the plot takes a long time to generate the plot due to large computations.

The *Movie* pushbutton will generate a movie for the selected checkbox. As the movie option is only available for time series and histogram plot, it will give an error if any of checkboxes is selected.The *Deselect all* pushbutton deselects all the checkboxes. This feature is added so that the user need not have to deselect each of the checkboxes.The Plot of selected range of data will generate a plot for user defined *Starting sample limit* and *Ending sample limit.*The panel named *X-axis settings* allows the user to display either time or samples on X-axis. This functionality is available for all the plots.

The panel named *Statistics* contains a pushbutton named *Save to table* which on selection will display all the displayed parameters in the listbox to the uitable. The radio button named *Statistics of selected data* will display all the statistical parameters between the user defined *Starting sample limit* and *Ending sample limit*. The pushbutton named *Save the table to spreadsheet* will save the uitable to excel spreadsheet, the user can also view the spread sheet on selecting the *View the saved spreadsheet* pushbutton. The pushbutton named *Clear* the table will clear the uitable.

4.4 Diagnostics tool

The diagnostic tool computes statistical parameters like MAD, Kurtosis and provides 2-D plots for raw voltage and square of raw voltage for a given block size and accumulation. Figure 16 shows the diagnostic tool GUI.

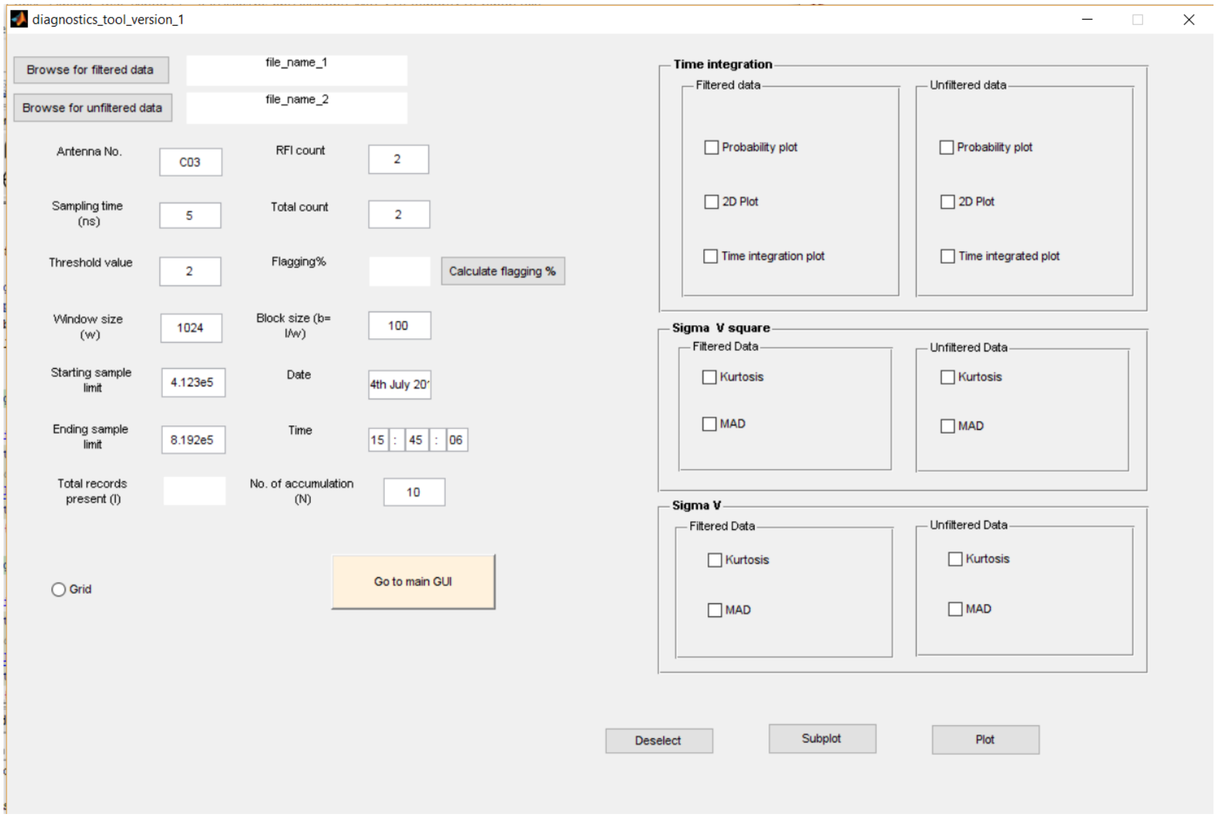

Figure 16: Diagnostic Tool GUI

This GUI contain some of the common features to raw voltage GUI. For example some common inputs are, ‘Antenna No. ’, ‘Sampling time (ns) ’, ‘Threshold value ’, ‘Window size ’, ‘Starting sample limit ’, ‘Ending sample limit ’, ‘RFI count ’, ‘Total count ’, ‘Date ’, ‘Time ’, ‘Block size ’. Here the block size is;

$$Blocksize = \frac{Totalrecordspresent}{Windowsize}$$

Input ‘No. of accumulation ’is the number of blocks integrated.

The *Time integration* panel contains two panels namely *Filtered data* and *Unfiltered data*. Each of the panel consists of three checkboxes namely *Probability plot, 2D Plot, Time integration plot*. The user can have the individual plots of filtered and unfiltered data by selecting the push button *Plot* and can also have a subplot of filtered and unfiltered data in the figure for comparison.

Similarly the *Sigma V square* panel contains two panels and will provide *Kurtosis* and *MAD* plot.On the similar lines the *Sigma V* panel contains two panels and will provide *Kurtosis* and *MAD* plot

So basically this tool will help the user to understand the RFI more closely and understand its properties.

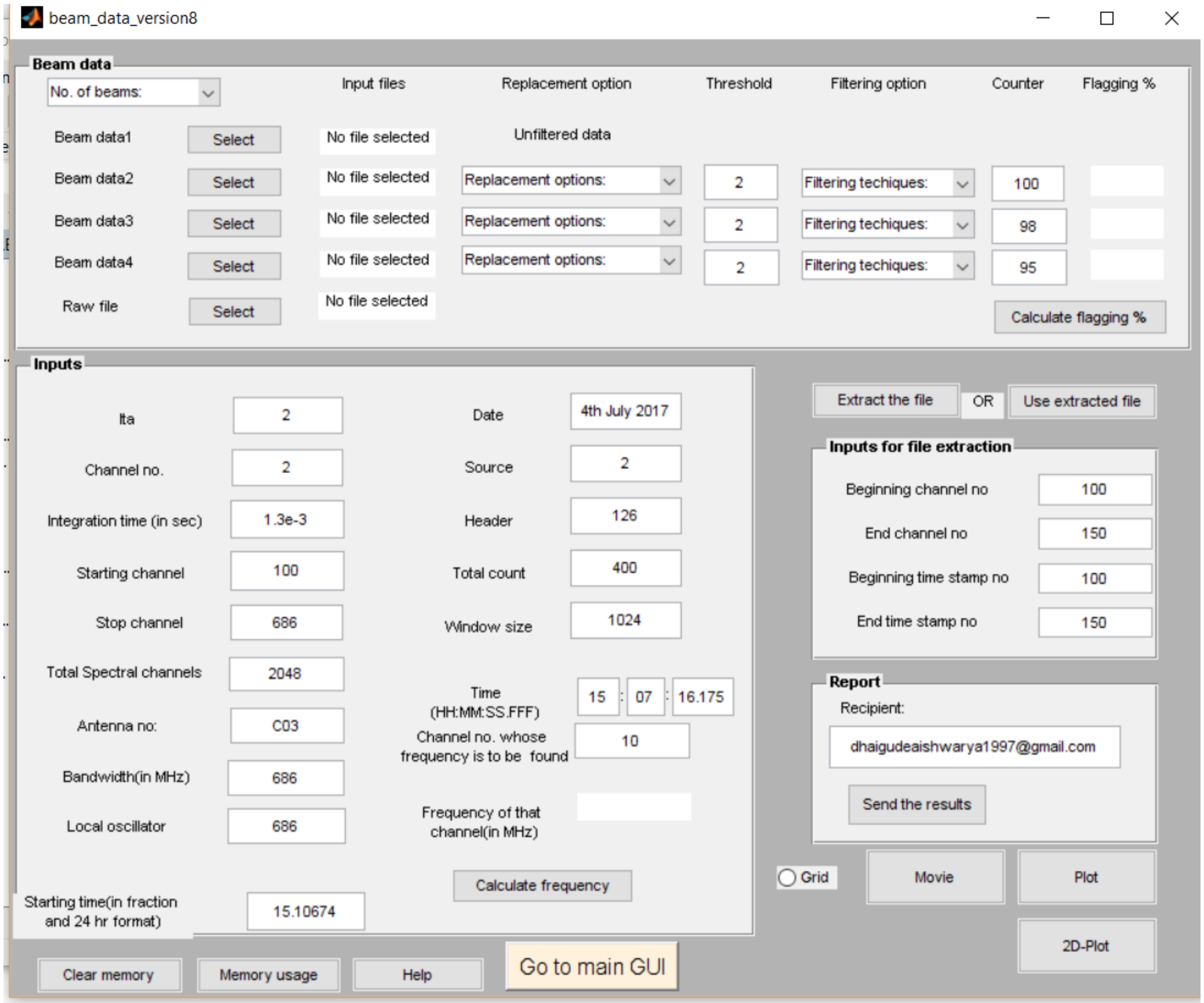

Figure 17: Beamformer GUI

4.5 Beamformer GUI

This GUI is for the observation and analysis of the beam data. The GUI provides an option for two, three and four beam data analysis. It also provides an option for the analysis of multiple channels for multiple timestamps using the pushbutton named *2D plot*. Figure 17 shows the beamformer GUI and the figure 8 shows the design flow for beamformer GUI.

When the GUI is run, the four of the five select pushbuttons in the panel named *Beam data* are disabled. The user first needs to select the *No of beams* from the pop up menu. As soon as the *No of beams* are selected, the required number of *Select* pushbuttons will be enabled. The *Select* pushbutton for the unfiltered file is already enabled as it is irrespective of the number of beams.If the user selects Two beam, then the second select pushbutton will be enabled. If the user selects Three beam, then the second and third pushbuttons will be enabled. Similarly all the four select pushbuttons will be enabled for Four beam input. In the callback of each select pushbutton, the in-built MATLAB function uigetfile() is called. It pops a dialog box displaying the files and the user can select whichever file he wants.

Each of the filtered files is provided with a *Replacement option, Filtering technique,Threshold* and a *Counter* as these parameters may be different for different files. As the unfiltered beam file and the raw file wont be having a replacement option, filtering technique, threshold or RFI count the inputs are provided only for the filtered data files.

The *Calculate Flagging %* pushbutton calculates the flagging percentage for each of the filtered files. Flagging percentage is calculated as:

$$Flagging\% = \frac{RFIcount}{Totalcount} \times 100$$

where the RFI count is the number of samples for which RFI was present in the signal and the total count is the total number of samples in the data. The beam data files are obtained by extracting a single channel from a raw data file containing multiple channels. In order to observe multiple channels for multiple timestamps we provide an option called as the *Extract file*, The callback of this pushbutton enables the select pushbutton for the raw data file. The user then can specify the channels and timestamps he wishes to extract in the inputs for channel and time stamp extraction panel. The multiple channel with multiple time stamps are obtained in *2D plot*. This allows us to observe if there was RFI present in a particular channel or time stamp.

If the *Use the extracted file* pushbutton is pressed, the inputs for Extract file option are disabled and the *Plot* pushbutton is enabled in its callback. In this case, if the plot pushbutton is pressed, it gives the plot for beam files in subplot as per the number of beams selected. Each subplot is an overlay of an unfiltered and a filtered file and one subplot is for the improvement. Therefore, for two beam, there will be two subplots, one is an overlay of filtered and unfiltered files and the other is of improvement plot.

The X-axis for each of the plots is the time IST (Indian Standard Time) at which the observation was made. This time is obtained by performing calculations on the starting time and the integration time. All the plots, overlay and the improvement factor can also be obtained with grid if the Grid radio button is selected.

The *Movie* pushbutton will generate a movie for the selected checkbox. As the movie option is only available for time series plot, it will give an error if any of the phase checkboxes is selected. The frame of the movie is decided by the window size. The *Movie* option will help notice the time series more clearly.

The plots, .m file and the fig files can be mailed to a person by typing his/her mail address in the static textbox present in the report panel and pressing the *Send the results* pushbutton. The callback of this pushbutton uses the sendmail function which is inbuilt in MATLAB. The sendmail provides with a provision of ending multiple files with a message in the mail. However the size of this message must be less than 75 characters.

4.6 Correlator GUI

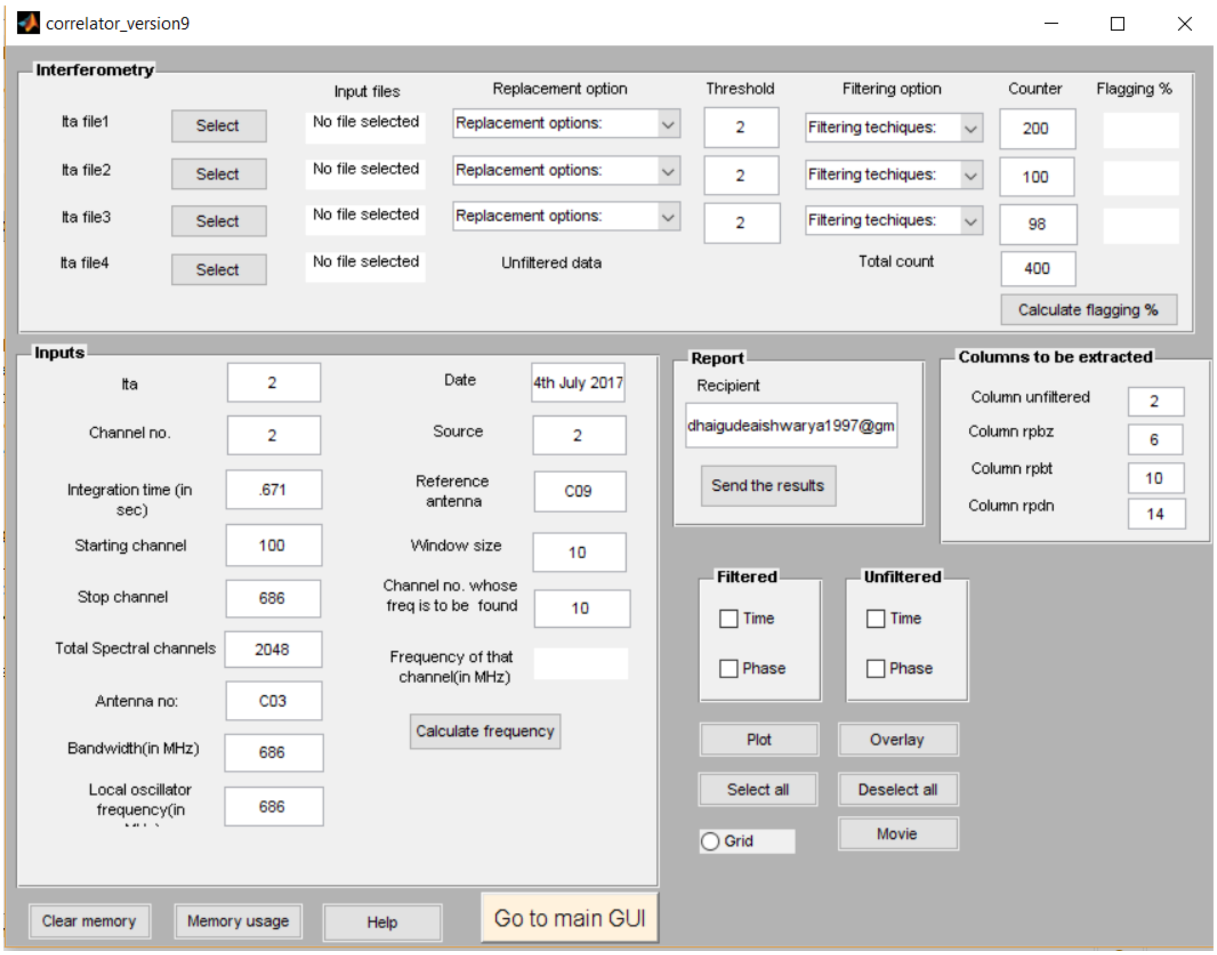

Figure 18: Correlator GUI

The correlator GUI is used for the analysis and observation of interferometry data. Figure 18 shows the screenshot and the figure 9 shows the design flow for *Correlator GUI*. The *Select* pushbutton for each of the files allows the user to browse through his/her PC and select the required file. The callback of each of the select pushbuttons sets the static textbox with the filename of the browsed file and also reads the file into a variable called filename after the neglecting the number of rows specified by the header. After extracting the channel specified by the column number for that file, the file is store in another variable. It is also checked that the column to be extracted from the file does not exceed the total number of channels from the file.

Each of the files is provided with a replacement option, filtering technique, threshold and a counter as these parameters may be different for different files. As the unfiltered beam file will not be having a replacement option, filtering technique, threshold or RFI count the inputs are provided only for the filtered data files. The

Calculate Flagging % pushbutton calculates the flagging percentage for each of the filtered files. Flagging percentage is calculated as:

$$Flagging\% = \frac{RFIcount}{Totalcount} \times 100$$

The 'lta'input is Long Term Accumulation. This is the used in the calculation of time for interferometry files. The inputs such as 'Antenna no.', 'Ref antenna', 'Date'are used in the title of the plots whereas the replacement options, filtering technique and threshold are specified in the legends.

The *Select all* pushbutton selects all the checkboxes and the *Deselect all* pushbutton deselects all the checkboxes. These features are added so that the user need not select or deselect each of the checkboxes. The *Plot* pushbutton gives a plot of the selected checkboxes. If no checkbox is selected, it pops an error dialog box. The callback of the plot pushbutton consists of consideration for each case of checkbox selected. The plot gives a plot of all the selected checkboxes in subplot. The time series plots are obtained as 'Power(in arb. Units)'versus IST and the phase plots are obtained as 'Phase'versus IST.

The *Overlay* pushbutton gives an overlay of selected checkboxes along with the overlay of the improvement plots. However overlay works only when the time-time or phase-phase or all the checkboxes are selected. The legends are provided for the overlay to distinguish the plots.The Improvement factor is a plot of improvement in dB versus IST. The improvement factor is the ratio of the mean by rms of filtered data to the mean by rms of unfiltered data converted in dB. The mean by rms is nothing but Signal to Noise Ratio(SNR). Thus, 'improvement'is the improvement in SNR.

The *Movie* pushbutton will generate a movie for the selected checkbox. As the movie option is only available for time series plot, it will give an error if any of the phase checkboxes is selected. The frame of the movie is decided by the window size. The movie option will help notice the time series more clearly.

All the plots, overlay and the improvement factor can also be obtained with grid if the Grid radio button is selected.The Xaxis for each of the plots is the IST(Indian Standard Time) at which the observation was made. This time is obtained by performing calculations on the starting time and the integration time.

The plots, .m file and the fig files can be mailed to a person by typing his/her mail address in the static textbox present in the report panel and pressing the *Send the results* pushbutton. The callback of this pushbutton uses the sendmail function which is inbuilt in MATLAB. The sendmail provides with a provision of sending multiple files with a message in the mail. However the size of this message must be less than 75 characters.

4.7 Combined GUI

The GUI provides the user with an option to plot and analyze beam as well as interferometry data simultaneously. The GUI provides an option for two, three

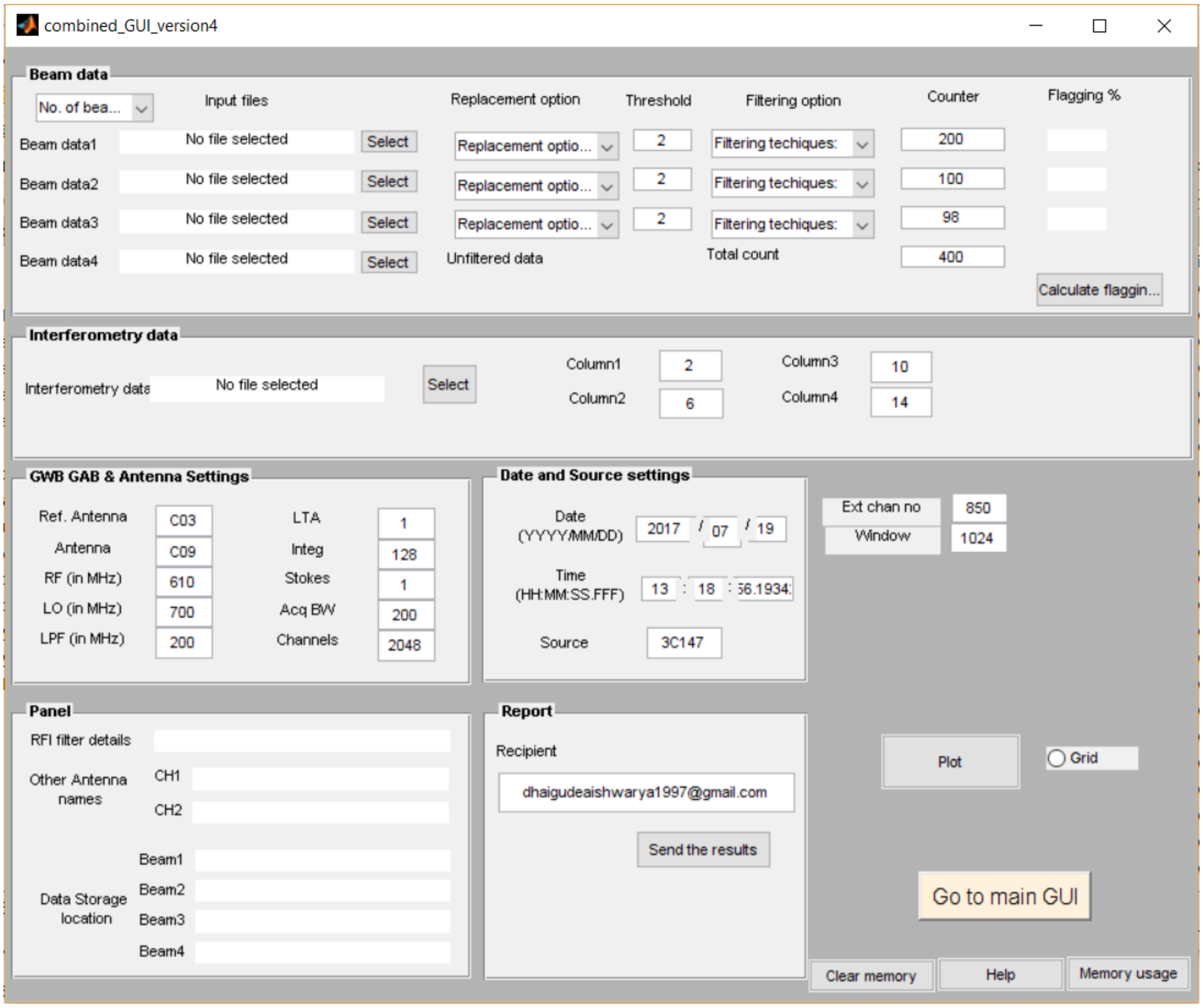

Figure 19: Combined GUI

and four beam data analysis. Figure 19 shows the screenshot and the figure 10 shows the design flow for *Combined GUI*.

Beam data panel:
When the GUI is run, the three of the four *Select* pushbuttons are disabled. The user first needs to select the *No of beams* from the pop up menu. As soon as the *No of beams* are selected, the required number of *Select* pushbuttons will be enabled. The *Select* pushbutton for the unfiltered file is already enabled as it is irrespective of the number of beams.The *Select* pushbutton for each of the files allows the user to browse through his/her PC and select the required file. The callback of each of the select pushbuttons sets the static textbox with the filename of the browsed file and also reads the file into a variable.

Each of the filtered files is provided with a *Replacement option, Filtering technique, Threshold* and a *Counter* as these parameters may be different for different files. As the unfiltered beam file will not be having a replacement option, filtering technique, threshold or RFI count the inputs are provided only for the filtered data files. As soon as the No of header input is entered, the select pushbutton for interferometry is enabled in its callback. The file is read after the neglecting the number of rows specified by the header in the callback of select pushbutton. The header detection is done automatically so that the user need not open the file and then count the number of lines to be neglected as the header. The interferometry file contains the amplitude as well as the phase information, hence we need to select only one interferometry file. The interferometry file has the same replacement and filtering technique as the beam file.

The *Calculate Flagging %* pushbutton calculates the flagging percentage for each of the filtered files. Flagging percentage is calculated as:

$$Flagging\% = \frac{RFIcount}{Totalcount} \times 100$$

Both the RFI count and the total count are taken as inputs from the user.

GWB and GAB antenna settings:
The panel includes the inputs such as antenna name, reference antenna name, RF, LO, LPF frequencies and other antenna parameters.RF is the Radio frequency, LO is the local oscillator frequency and the LPF is the cut-off frequency of the low pass filter. All these frequencies are in MHz.The input channels denotes the total number of channels.The LTA is the Long Term Accumulation. It is used in the calculation of sampling time for interferometry file. Sampling time for interferometry file = lta*0.671. The 'Stokes' input is used to find the polarization of the data.

Date and Source settings:
This panel consists of inputs such as date on which the observation was made, time of the observation and the source in the sky which is under observation. Though all the inputs are not used in the code for any of the calculations, they are taken from the user so as to know the details about the data being observed and record them. It helps the user to later track the saved reports.

The *Plot* pushbutton generates two figures. One figure is for beam data plots and the other figure is for interferometry file. The beam plot contains an overlay of unfiltered and filtered signals in time in subplots. Each subplot is an overlay of an unfiltered and a filtered file and one subplot is for the improvement. The number of subplots may differ as per the number of beams selected. Therefore, for two beam, there will be two subplots, one is an overlay of filtered and unfiltered files and the other is of improvement plot. The Xaxis for each of the plots is the time (Indian Standard Time) at which the observation was made. This time is obtained by performing calculations on the starting time and the integration time.The interferometry file contains three subplots. One subplot gives an overlay of Time series, second gives the overlay of improvement factor and third is the overlay of phase plots.

The plots, .m file and the fig files can be mailed to a person by typing his/her mail address in the static textbox present in the report panel and pressing the *Send the results* pushbutton. The callback of this pushbutton uses the sendmail function which is inbuilt in MATLAB. The sendmail provides with a provision of sending multiple files with a message in the mail. However the size of this message must be less than 75 characters.

5 Results

5.1 Raw voltage GUI

Figure 20 to figure 25 shows the output figures for the input raw files

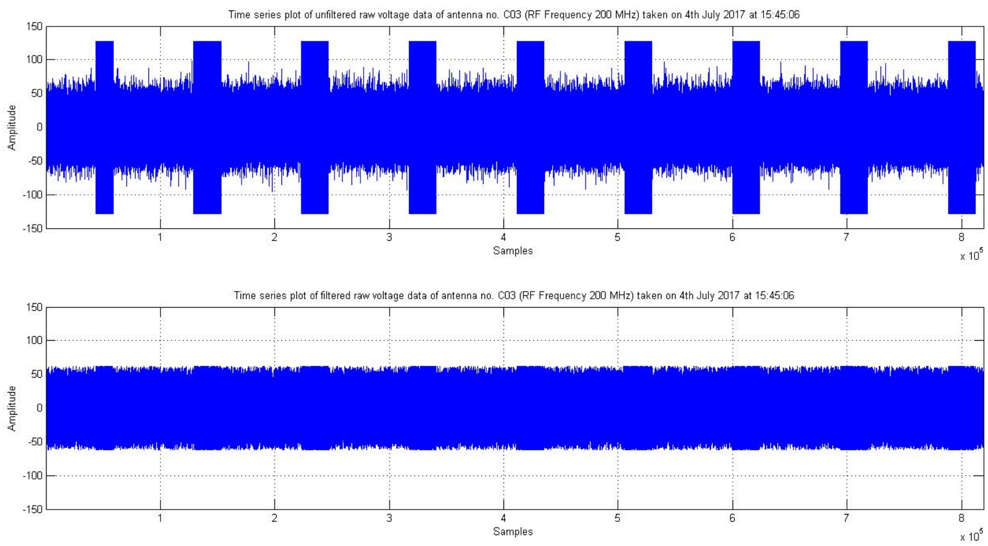

Figure 20: Time series plot of filtered and unfiltered raw voltage data.

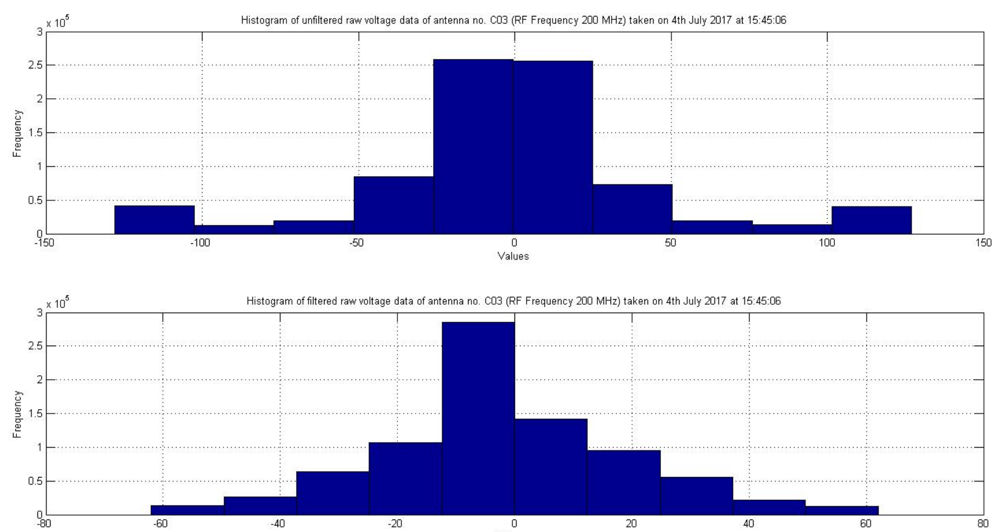

Figure 21: Subplot of histogram of filtered and unfiltered raw voltage data.

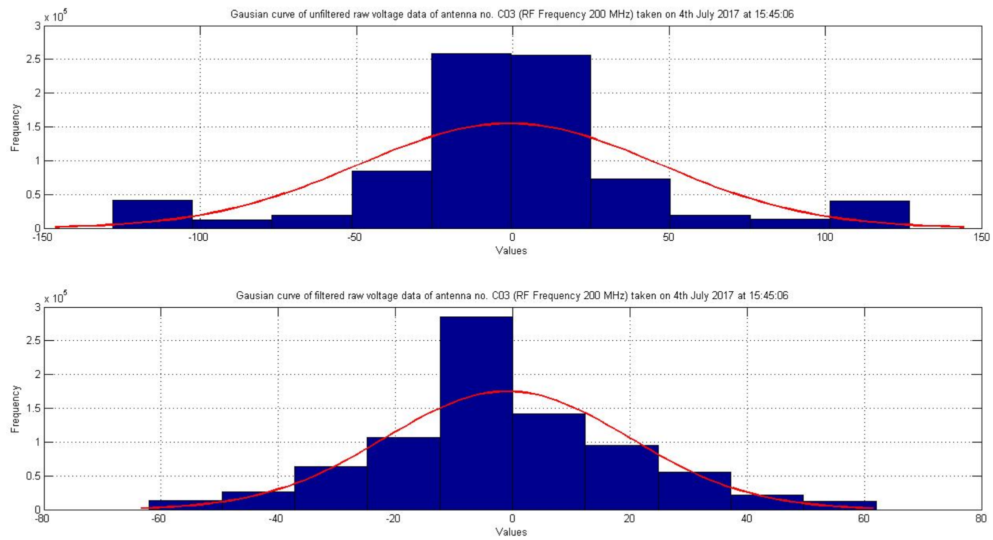

Figure 22: Subplot of gaussian curve fitted in histogram of filtered and unfiltered raw voltage data.

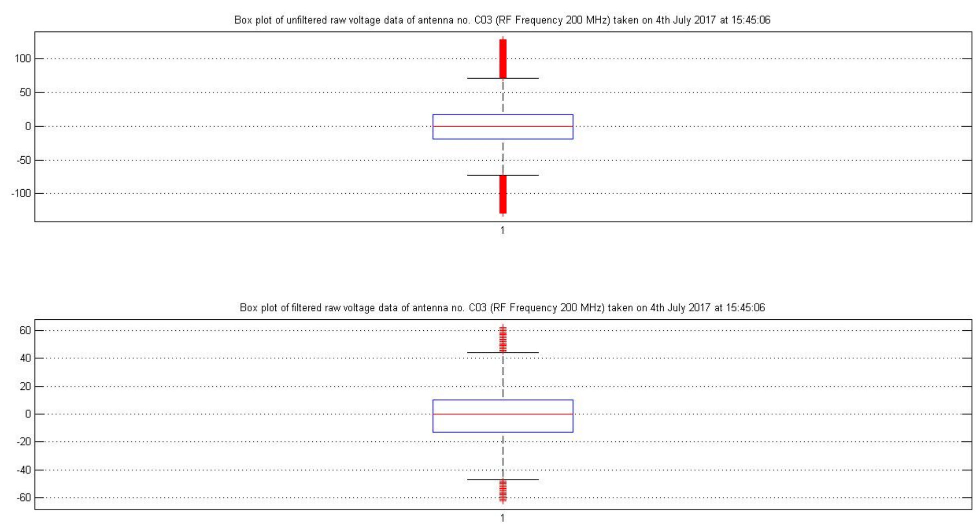

Figure 23: Subplot of boxplot of filtered and unfiltered raw voltage data.

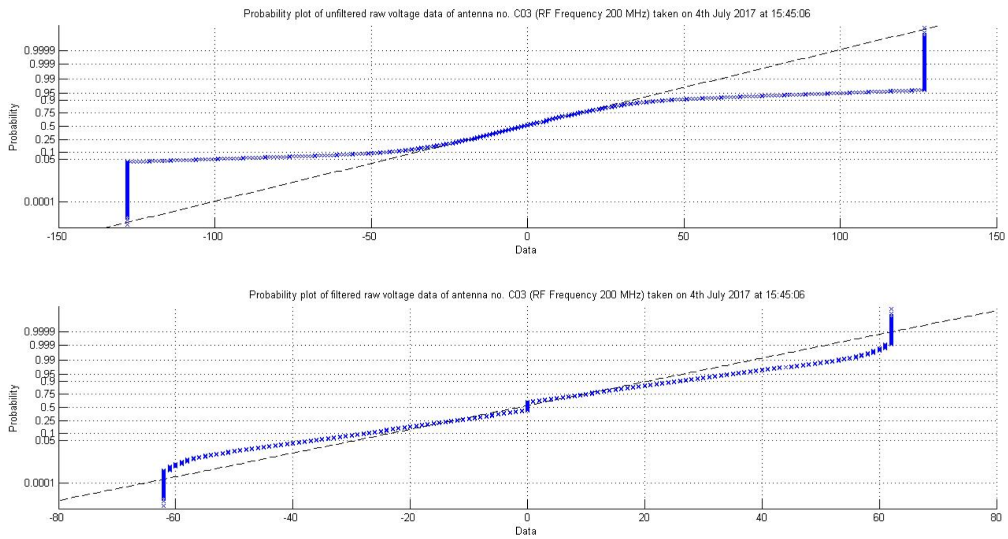

Figure 24: Subplot of probability plot of filtered and unfiltered raw voltage data.

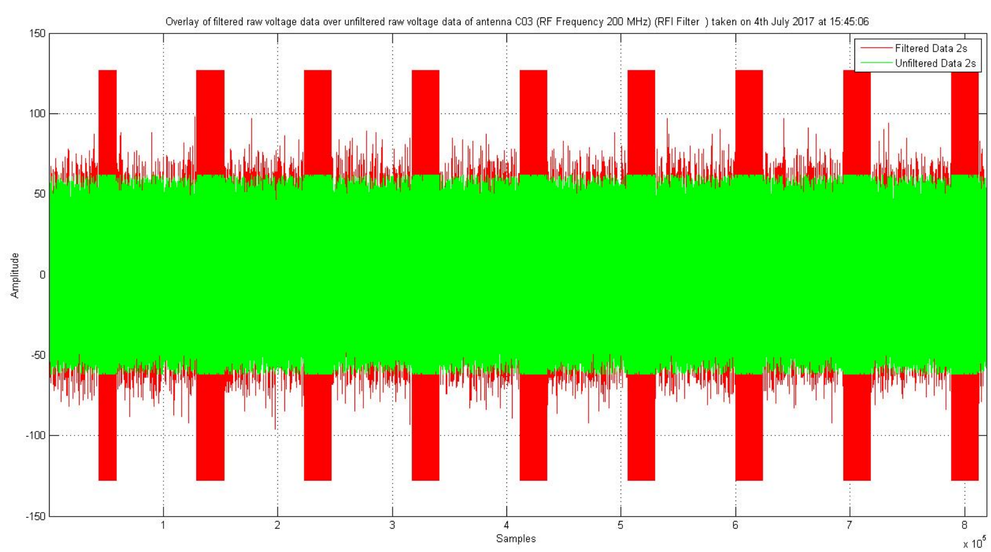

Figure 25: Overlay of filtered over unfiltered raw voltage data.

5.2 Beam GUI

figure 26 shows the output figure for the input files of 19July2017 raw files

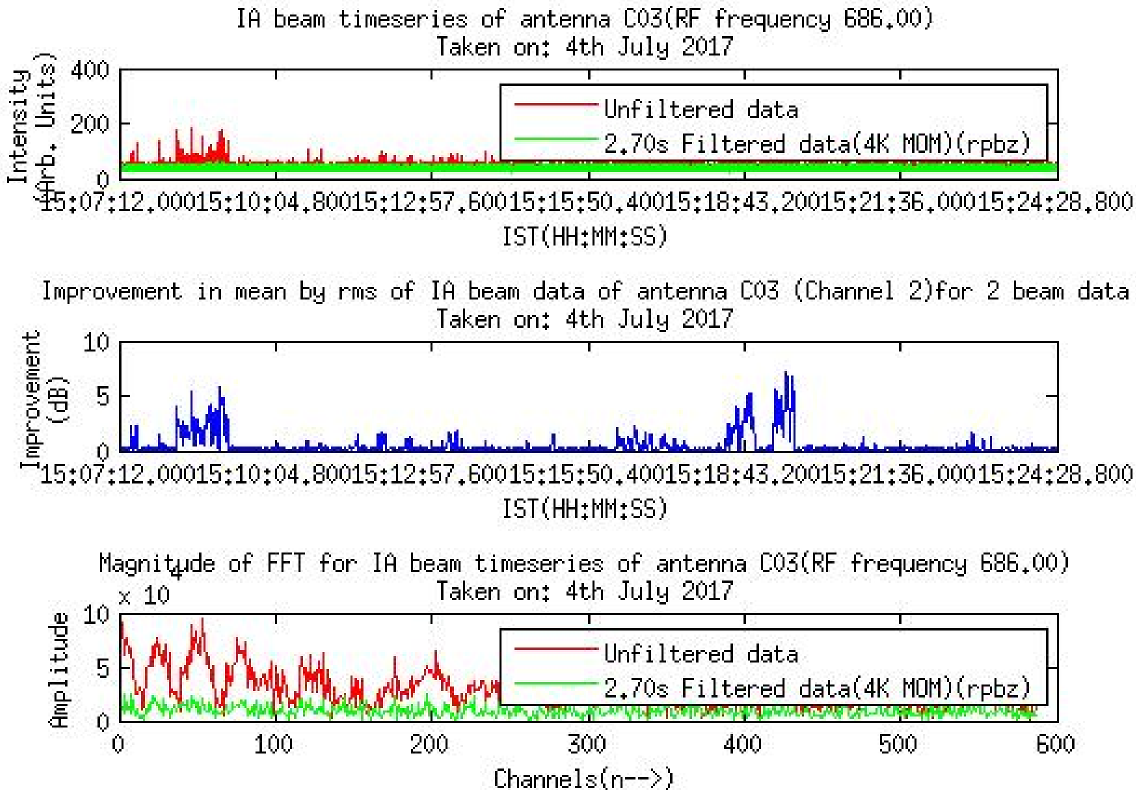

Figure 26: Plot for Beam GUI(three beam)

5.3 Correlator GUI

Figure 27 and Figure 28 show the output figures for the input files of 19July2017(lta file)

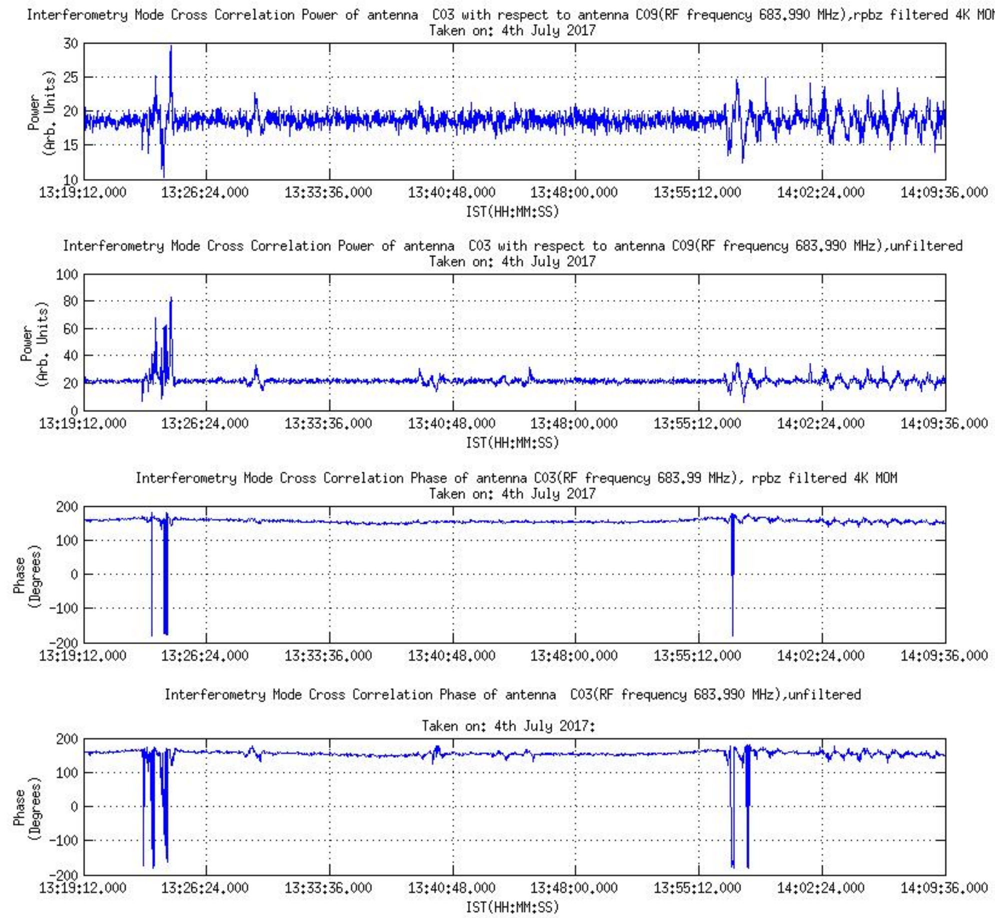

Figure 27: Plot for plot option in correlator GUI

5.4 Combined GUI

figure 29 and figure 30 show the output figure for the input files of 19July2017(raw and lta files)

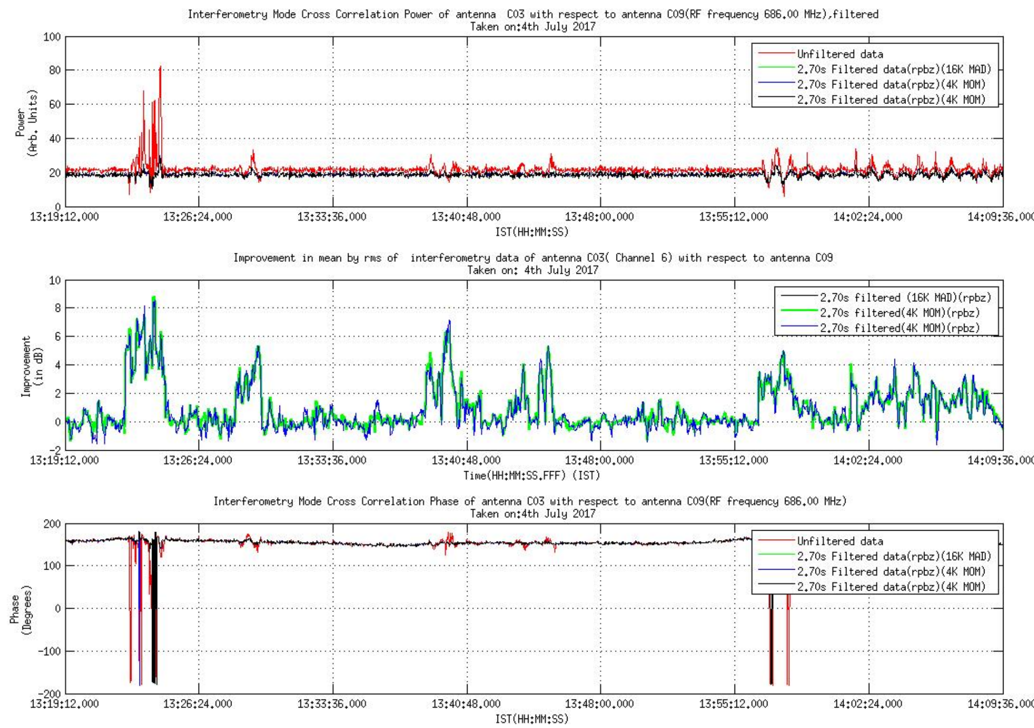

Figure 28: Plot for overlay option in correlator GUI

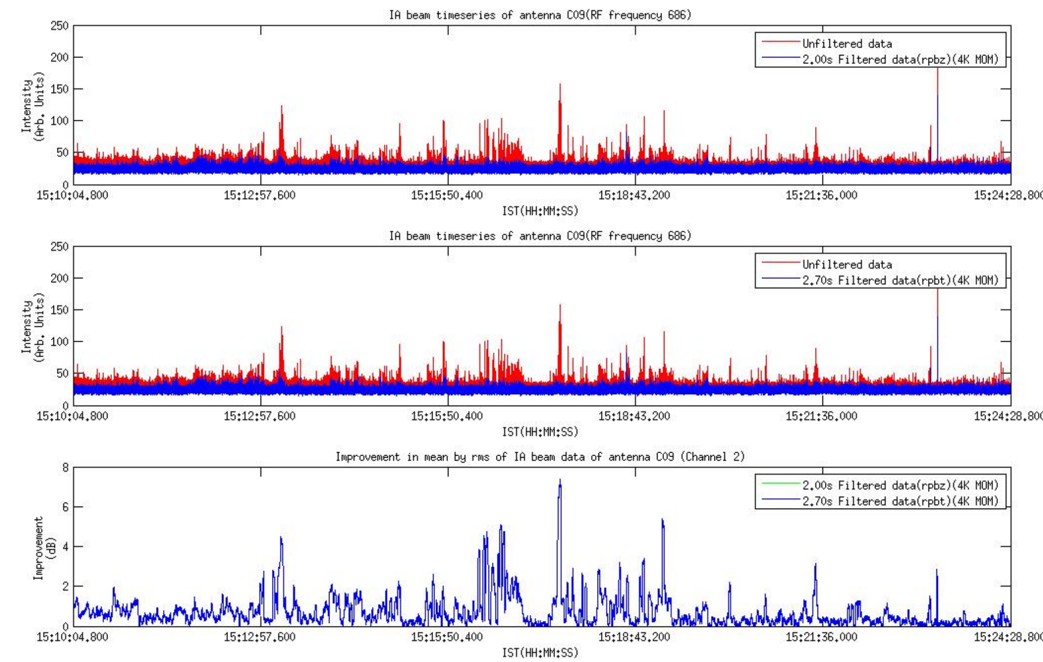

Figure 29: Plot for Combined GUI(three beam)

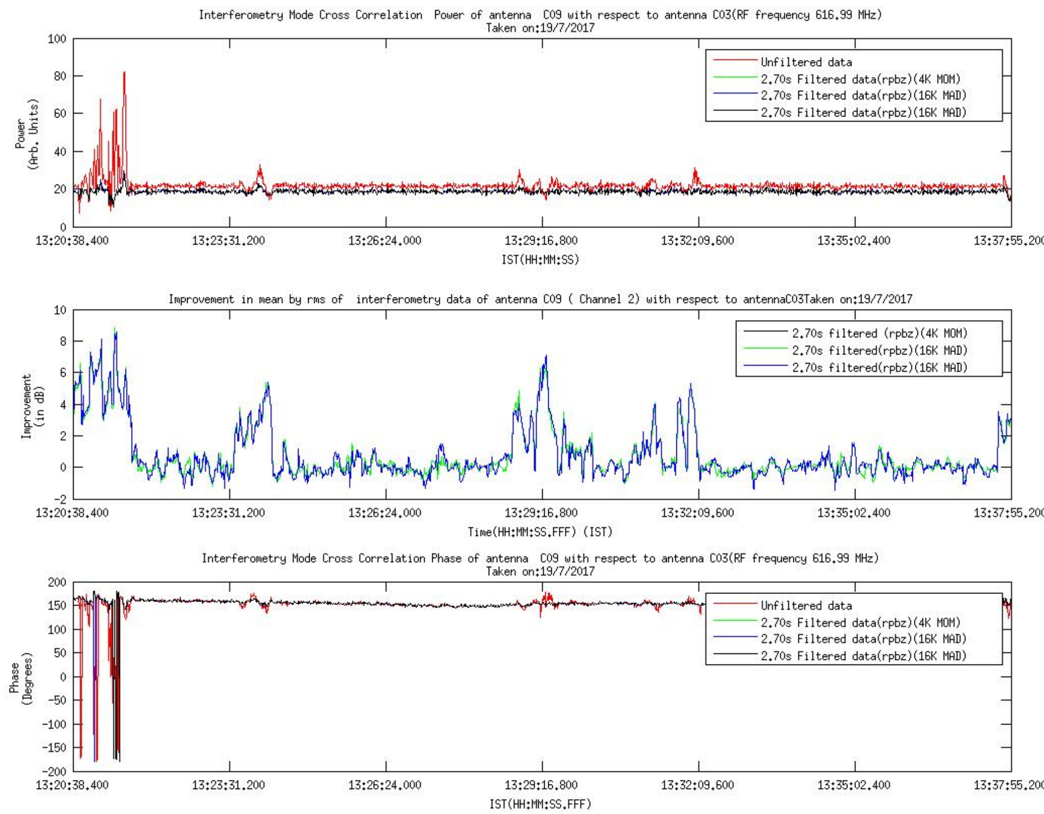

Figure 30: Plot for Combined GUI(three beam)

6 Conclusion and Future scope

6.1 Conclusion

1. A number of MATLAB commands and in-built functions were explored.
2. Exception handling is provided wherever necessary, so that the user gets to know if an error is committed.
3. The resolution of the GUI has been set such that it adjusts itself according to the desktop screen on which it is run.
4. The time required to obtain an improvement plot for raw voltage, beam data and correlator data is too long . In comparison with raw voltage and correlator data, beam data takes much longer time. The time required is long due to the mean by rms calculations. However, the time can be reduced by parallelizing loops in the code for faster computation.

6.2 Future scope

1. An executable file can be generated for the current files.
2. Computation time for the Improvement factor of beam data can be reduced.
3. The send mail option can be modified to send the mail to multiple people at the same time.
4. The statistics table in raw voltage GUI can be saved to a .csv file.
5. A report can be prepared for the combined GUI and can be saved to a .csv file.
6. Modifications and addition of new features in the diagnostic tool.

References

[1] I-WONDER-MAGAZINE-Issue3, pp. 66-67

[2] https: $//en.wikipedia.org/wiki/Electromagnetic_interference$

[3] Ford, John M., and Kaushal D. Buch. "RFI mitigation techniques in radio astronomy." In *Geoscience and Remote Sensing Symposium (IGARSS)*, 2014 IEEE International, pp. 231-234. IEEE, 2014.

[4] Buch, Kaushal D., Shruti Bhatporia, Yashwant Gupta, Swapnil Nalawade, Aditya Chowdhury, Kishor Naik, Kshitij Aggarwal, and B. Ajithkumar. "Towards Real-Time Impulsive RFI Mitigation for Radio Telescopes." *Journal of Astronomical Instrumentation 5, no.* 04 (2016): 1641018.

[5] Buch, Kaushal D., Yashwant Gupta, Shruti Bhatporia, Swapnil Nalawade, Kishor Naik, and B. Ajithkumar. "Real-time RFI excision for the GMRT wideband correlator." In *Radio Frequency Interference (RFI)*, pp. 11-15. IEEE, 2016.