Development of MAD based spectral domain RFI filtering techniques

FINAL SEMESTER PROJECT REPORT

by

Nishit Baburaj

SC09B131

Department of Avionics

Indian Institute of Space Science and Technology

Thiruvananthapuram

APRIL 2013

ACKNOWLEDGEMENTS

I take this opportunity to thank **Dr. Thomas Kurian, HOD, Avionics, IIST** for giving me the opportunity to work for the GMRT.

I would like to thank my guide K**aushal Buch** of the digital backend group at GMRT for guiding, monitoring and encouraging me throughout the project. His dedication, annotations and thought provoking discussions were the key behind the successful completion of this project.

I would also like to thank **Ajith Kumar, Group Coordinator, Digital Backend Group, GMRT** whose guidance and support helped in the completion of the project.

I would also like to thank **Prof. Yashwant Gupta, Dean, GMRT Observatory,** for sharing his expertise in RFI mitigation and allied fields. Without his guidance the project would have remained unfulfilled.

I also take this opportunity to express my thanks and appreciation to the GMRT Digital Back-end team for their cooperation and help. I sincerely thank **Harshavardhan, Mekhala, Sandeep, Sanjay, Irappa, Shelton** and **Bhonde** for their help during my project work. I would like to thank the analog backend group for providing the hardware and necessary support for carrying out the narrow band radiation tests for this project

I would also like to thank the staff members of GMRT who were always supportive and friendly.

April 2013 **Nishit Baburaj**

ABSTRACT

Radio Frequency Interference (RFI) is a growing concern for most of the ground-based radio telescopes. The upgraded wideband GMRT digital backend aims at implementing of real-time RFI mitigation techniques for achieving a better sensitivity in presence of RFI. This project aims at study, simulation and analysis of narrow band RFI mitigation using a robust statistical estimator- Median Absolute Deviation (MAD). Following a study of narrow band RFI and its simulation in MATLAB, MAD based RFI filtering was carried out on simulated data. The same filtering was subsequently carried out for data acquired from the antenna through radiation of narrow band RFI. MAD estimation was carried out in two ways: 1.Each individual spectral channel over time and 2.Across the spectral channels. The results using both these approaches are reported in detail. Finally MAD implementation techniques on FPGA are reviewed.

CERTIFICATE

This is to certify that the report entitled **“Development of MAD based spectral domain RFI filtering techniques”** submitted by Nishit Baburaj to the Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfilment for the award of the degree of Avionics is a *bona fide* record of research work carried out by him/her under my/our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

B. Ajith kumar
Group Coordinator
Digital Backend Group
GMRT

Kaushal Buch
Guide
Digital Backend group
GMRT

Pune

April, 2013

TABLE OF CONTENTS

Acknowledgements ii

Abstract iii

List of figures vi

1. GMRT AND ITS BACK-END SYSTEM 1

2. RFI AND NEED FOR MITIGATION 3

3. SPECTRAL DOMAIN RFI REMOVAL USING MAD 8

4. MAD FILTERING ON SIMULATED AND ACTUAL NARROW BAND RFI DATA 11

5. OVERVIEW OF MEDIAN COMPUTATION TECHNIQUES FOR FPGA IMPLEMENTATION 17

6. CONCLUSION 21

7. FUTURE WORK 22

APPENDIX 23

I. INR 23

LIST OF FIGURES

Figure 1 : Narrow band RFI in signal received by one of the GMRT antennas. RFI peak is about 28 dB higher in power level than the astronomical signal 4
Figure 2: Time-frequency representation for understanding MAD filtering methods 9
Figure 3 : Flow chart of MATLAB simulation 10
Figure 4 : Over time MAD filtering simulated in MATLAB 11
Figure 5: Across channel MAD filtering 12
Figure 6 : GSB recorded data before and after MAD filtering 13
Figure 7 : Channel 73 with 100% RFI 13
Figure 8 : Radiation test plots before and after MAD filtering 14
Figure 9 : Plot of channel no. 148 before and after MAD filtering 15
Figure 10: Histogram of channel 148 with RFI in it 16
Figure 11 : Histogram of channel 148 without RFI 16
Figure 12: Histogram based median filter architecture 20
Figure 13 : INR for real part before and after MAD filtering 23
Figure 14: INR for real Imaginary before and after MAD filtering 24

1. GMRT AND ITS BACK-END SYSTEM

1.1 About GMRT

The Giant Metre wave Radio Telescope (GMRT) consists of thirty 45 m diameter antennas spread over a 25 km region. Half the antennas are in a compact, quasi randomly distributed array with a diameter of about 1 km. The remaining antennas are on 3 arms of length of about 14 km (North West, North East and South) with 5 or 6 antennas on each arm. The longest baseline is about 25 km and the shortest is about 100 m without foreshortening [1].

1.2 GMRT Upgrade

A major upgrade of the GMRT is currently underway. The upgrades are aimed to provide

- seamless frequency coverage from 50 to 1500 MHz
- improved sensitivity with better quality receivers
- a maximum instantaneous usable bandwidth of 400 MHz
- a revamped and modern servo system
- a new generation monitor and control system
- improvements in the antenna mechanical structure
- improvements in infrastructure and computational facilities.

The upgrade will result in significant changes to almost all aspects of the GMRT receiver chain and other systems [1].

1.3 GMRT Digital Backend

The main back-end for the GMRT is the GMRT software back-end (GSB). The GSB handles the full 32 MHz baseband signals from each of two polarisations for all 30 antennas, which are digitised and sent to a networked cluster of PCs that performs all the operations needed to realise a correlator and a pulsar receiver, in real time. The standard processing features include gain equalisation, integer and fractional delay correction and fringe stopping for the signals from each antenna. The GSB implements a FX type correlator, with user selectable number of spectral channels across the band.

The GSB has an offline mode, where the raw voltage data from all the antennas can be recorded on an array of SATA disks attached to the GSB cluster, for offline processing.

The entire operation of the GSB is controlled by a set of user friendly functions implemented in a graphical user interface (GUI). The GSB is interfaced to the main control and monitor software of the GMRT. The output data of the GSB are compatible with the existing data formats at the observatory, for the interferometry and beam modes [1].

1.4 GMRT Digital Backend: Upgrade Activities

GMRT digital back end upgrade activities include design and implementation of high speed digital signal processing algorithms for wideband interferometer and pulsar observations. The implementation of digital backend is being carried out on two different computing platforms- FPGA and GPUs. Apart from the development of pocket and packetized correlator/beam former and GPU correlator, the group is also involved in development of auxiliary processing blocks like digital noise and source, digital down converter and RFI mitigation [2].

2. RFI AND NEED FOR MITIGATION

2.1 Radio Frequency Interference

Radio Frequency Interference (RFI) is an unwanted but detectable portion of a desired observation that has the potential to either degrade or inhibit the successful conduct of the observation. Some interference is not easily detectable but can still degrade the observations.

Until recently, the standard observing modes and signal processing techniques used in the course of making observations provided an inherent degree of interference mitigation that proved adequate to provide useful astronomical data in the presence of some interference. For example, “fringe stopping” in aperture synthesis imaging has the tendency to de-correlate the

RFI received at widely-separated antennas, which tends to suppress the RFI in the associated correlation products [3].

RFI has challenged astronomers for decades. But in the recent years it has reached a point where mitigation strategies have to be implemented. Earlier astronomers had specific bandwidth in which they used to observe and hence could avoid radio frequency as that bands were specifically allocated to them. But now astronomers have become more ambitious and want to observe in all possible frequencies. Moreover with increase in population and development of technology, RFI has been a major cause of concern as it degrades the incoming radio signals [4].

In essence, astronomers can choose either to discard contaminated material, including target data along with RFI, or they can excise the RFI without damage to other data. By blanking they can identify the data compromised by RFI and flag the data as bad, but it is obviously preferable to characterise the RFI and remove it (and it alone) from the data. Over the past decade a number of mitigation schemes have been trialled, and shown to work. At the present time however, blanking is, to a very large degree, the only strategy in use [4].

2.2 Types of RFI

RFI can be broadly classified into two main categories:
1) Narrow band RFI emanates from intended transmissions such as radio or TV stations, cell phones.
2) Broad band RFI emanates from unintentional radiation from sources such as electric power transmission lines, inductive load switching.

In the Fourier domain, narrow band RFI is visible as a sudden peak in the spectrum whereas broadband RFI spread over the whole spectrum. Following the above property, it is easy to identify and remove narrow band RFI as it can be seen as a sudden peak whereas broad band RFI is difficult to distinguish from the rest of the spectrum. The figure below gives the types of interference and their typical sources [5].

Types of Interference	Typical sources
Broad band RFI	Power line RFI due to sparking/Corona discharge, Due to automobile sparking, Switching of an inductive load
Narrow band RFI	AM & FM stations transmitter's harmonics; Near –band unmodulated transmitter's carriers; Cell phone transmissions

2.3 RFI at GMRT

Since most of the RFI sources are man-made sources, the best solution for avoiding RFI would be to have observatories located away from interference sources or to schedule accordingly so as to reduce the RFI encountered.

But for GMRT, the problem encountered is more prominent mainly due to two reasons. The first one being, GMRT is spread over a large area and covering nearby towns. This opens up to numerous man made RFI which cannot be kept in control all the time. The second one being, GMRT is used to observe at low frequencies also, hence increasing the probability of contribution of RFI. The power line interference 50/100 Hz is very strong one and very difficult to avoid because of its frequency modulation with the radio signals [6].

High tension (H.T.) power-lines in the vicinity of the GMRT antennas produce harmful pulsed radio frequency interference (RFI) to radio astronomy observations, particularly in the 150 MHz band. Measurements show occurrence of groups of pulses occurring every 10ms (half cycle of 50Hz), with each pulse of approximately tens to few hundred micro-second duration.

It is known that severe radio noise (RFI) is generally caused by corona discharges on the HT lines of voltage > 65 kV. For lines of lower voltage, of around 11 kV and 33 kV, gap discharges at the insulators located at the electrical poles, poor grounding of the support arms and loose contacts in the joints gives rise to severe RFI. Similarly, RFI is also observed near the transformers of the irrigation pumps located near the antennas of the GMRT [7].

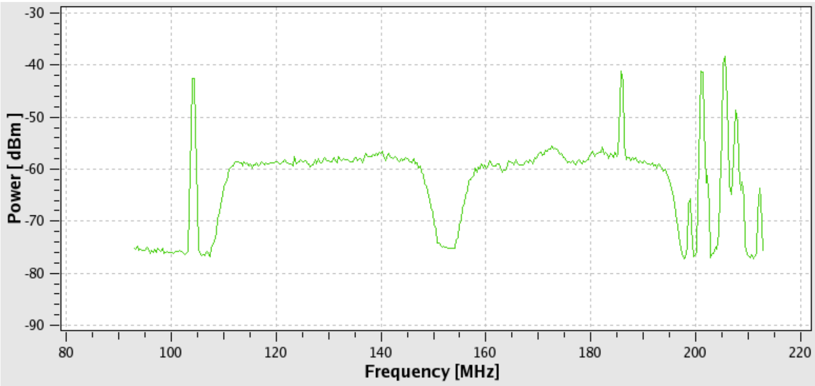


Figure 1 : Narrow band RFI in signal received by one of the GMRT antennas. RFI peak is about 28 dB higher in power level than the astronomical signal

2.4 RFI Mitigation Techniques

As there is no generic method to mitigate each class of interference, all methods depend on the ability to detect the interference i.e. Interference-to-Noise Ratio, INR. INR is defined as the ratio of the power of the interference present in the signal to the power of the noise of the signal. INR ratio is given by:

$$INR = \frac{\sigma_{RFI}^2}{\sigma_{Noise}^2}$$

The techniques used in such approaches cover a wide range of regulatory, technical, analog and digital means to avoid RFI and to later remove it from astronomical data. Therefore, RFI mitigation covers a very wide range of subjects and methods [8]:

- **a)** Measurements of the spectrum environment to identify and characterize the RFI at the telescope and identify ways to eliminate sources of interference,
- **b)** Establishment of quiet zones for existing and for new generation radio telescopes,
- **c)** RFI mitigation methods and at which location in the detection systems they may be optimally used,
- **d)** Digital filtering and sub-space filtering using peculiar characteristics of the RFI,
- **e)** Multiple methods of pre-correlation thresholding of RFI in the time and frequency domains using wide-band spectrometers,
- **f)** Adaptive noise cancellation of specific (well-defined) RFI signals,
- **g)** Spatial filtering with array instruments,
- **h)** Statistical methods to identify RFI in the data and to remove these signals,
- **i)** RFI mitigation algorithms built into software correlators,

The ideal RFI mitigation strategy would be to introduce machinery to reduce the impact of RFI damaging the astronomer's data. It should be automatic, reliable and robust. It should introduce artefacts which mimic real results. The cost of applying the machinery should be predictable. The cost should be less than that of doing nothing [3].

Mitigation techniques can be classified into two namely, Pro-active mitigation: avoid the RFI and Re-active mitigation: remove the RFI from data.
Pro-active mitigation includes Regulation of spectrum, Good observatory disciplines and Location of observatories in remote locations.

Re-active mitigation includes: **Excision**, **Cancellation** and **Anticoincidence**. Each of them is briefly described below:

1. **Excision:** It involves discarding data which are affected by RFI. RFI consisting of brief pulses might be mitigated by blanking the data when the pulse is present; this is known as temporal excision. Alternately, persistent RFI might be mitigated using array beam-forming techniques to orient pattern nulls in the directions from which the RFI is incident; this is known as spatial excision. Excision includes loss of data which may cause distortion of the remaining data and hence increases the observing time required to reach the required sensitivity or accuracy.

2. **Cancellation:** It is to identify and remove the RFI only while leaving the rest of the signal untouched. It is far more superior than excision as it doesn't include loss of data but only removal of the unwanted signal. It is much more difficult to implement because of the precision required and the varying characteristics of RFI.

3. **Anticoincidence:** It is based on the fact that widely separated antennas would perceive signals identically but RFI differently. The RFI contributes to the background noise level at each antenna rather than to correlated signals. This degrades the correlated signal received, which may require an increase in the observing time to achieve the signal to noise ratio needed.

The re-active mitigation schemes work generally works on short set of data. This means that low level RFI could be a problem: too weak to trigger the mitigation machinery, but able to show up in the final product. This implies that the RFI received should have significant value of INR (Interference to Noise ratio).

Signals from astronomical sources follow gaussian probability distribution function. Most of RFI that the GMRT sees is non-gaussian in nature. The RFI filtering algorithms are flagging algorithms that indicate the presence of non-gaussian RFI. These algorithms have a common workflow. First, a statistical quantity is calculated from the data. This quantity is compared with the corresponding quantity for the Gaussian distribution. If there is a large difference in these two quantities, the block of data considered is probably affected by non-gaussian RFI, and thus can be flagged. Also note that a considerable value of INR should be present for these algorithms to function properly.
The probability distribution function of a gaussian noise is by:

$$P(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}}$$

2.5 RFI Filtering Using Robust Statistical Techniques

There are various robust filtering algorithms but some of them which have been tested are given below.

1. **Kurtosis based filtering**:

Kurtosis is a measure of the "peakedness" of the probability distribution of a real-valued random variable. Kurtosis is a descriptor of the shape of a probability distribution. In Kurtosis based filtering, the kurtosis is defined as

$$K = \frac{m_4}{(m_2\,)^2}$$

Where m_4 and m_2 are its forth central moment and second central moment respectively.

From the above formulas, if the signal is corrupted by RFI, its PDF may deviate from that of a Gaussian. Consequently, the Kurtosis may deviate from 3. In the Kurtosis based approach of filtering, the detection of RFI boils down to calculating the Kurtosis estimator and deciding whether this corresponds to that of a Gaussian random variable. Kurtosis operates on a block of data. Entire block of data needs to be removed if the kurtosis value is other than 3 (± the estimation error).

2. **MAD based filtering**:

MAD (Median Absolute deviation) is a measure of statistical dispersion of the variability of a quantitative data set. For a univariate data set X_1, X_2,…….. X_n, the MAD is median of absolute deviations from the data's median.

$$\mathbf{MAD = median_i \left(\left| X_i - median_j (X_j) \right| \right)}$$

Hence the value varying from the MAD value significantly is considered as RFI and removed. MAD based filtering can be applied to each sample individually. MAD being a robust statistical estimator is more resilient to variation in data samples and has 50% resilience towards data change.

3. **Spectral Kurtosis**

The spectral kurtosis (SK) is a statistical tool which can indicate the presence of series of transients and their locations in the frequency domain. As such, it helpfully supplements the classical power spectral density, which as is well known, completely eradicates non-stationary information.

In this document, MAD estimator was considered for RFI mitigation algorithm as it was shown in Prassanna's Report [6] that MAD has huge advantages compared to rest of the algorithms.
The next chapters deal with MAD based filtering in the spectral domain for removal of narrowband RFI.

3. SPECTRAL DOMAIN RFI REMOVAL USING MAD

3.1 Introduction to MAD filtering

A brief introduction of MAD was given in the preceding chapter. In statistics, the median absolute deviation (MAD) is a robust measure of the variability of a univariate sample of quantitative data. For a data set X_1, X_2 …………………… X_N , the median of absolute deviations is given by

$$\mathbf{MAD = median_i\,(|\,X_i - median_j\,(X_j)\,|)}$$

The median absolute deviation is a measure of statistical dispersion. Moreover, the MAD is a robust statistic, being more resilient to outliers in a data set than the standard deviation. Median is fifty per cent resilient to outliers in data. In the standard deviation, the distances from the mean are squared, so large deviations are weighted more heavily, and thus outliers can heavily influence it.

Because the MAD is a more robust estimator of scale than the sample variance or standard deviation, it works better with distributions without a mean or variance, such as the Gaussian distribution and Rayleigh distribution.

The standard deviation of the sample is estimated by the formulae for Gaussian distribution:

$$\sigma = 1.4826 * MAD$$

3.2 Simulation of Spectral Domain MAD filtering

Before implementing MAD filtering on actual telescope data, its analysis was carried out on simulated data. The simulation was carried out in MATLAB. First of all, since astronomical signals follow Gaussian distribution, random gaussian noise was generated. The mean and the standard deviation were user defined. Next, RFI was generated in form of sine waves with user defined input parameters. The duty cycle of RFI insertion was also user defined so that the effect of MAD filtering due to changing amount of outliers could be studied. Then the sine waves according to their duty cycle were added randomly to the Gaussian noise and a signal similar to that received as astronomical signals with narrow band RFI was generated. Then the Fast Fourier Transform (FFT) of the signal was computed and the result were stored in a matrix.

Real and imaginary parts of the generated signal were separated and MAD filtering was performed on these values separately. Thereafter, the first phase of MAD filtering was applied that is calculation of MAD for the signal. There were two methods by which MAD could be calculated: Across the Channel and Over Time. Both these methods are described in detail in the later part of this chapter. After calculation of MAD, the robust standard deviation using MAD was calculated and threshold level was determined. The boundary level decided the interference to noise ratio and rejected accordingly to the value calculated. The threshold level is given by the equation:

$$\mathbf{Threshold = median \pm N * \sigma}$$

N is the user configurable aggressiveness parameter and σ is the standard deviation calculated using MAD.

3.3 Across the channel and over time analysis

As discussed in the previous section, MAD filtering can be carried out using two ways: Across the spectral channels and on each spectral channel over time. The details of both these methods are provided in this section. Consider the matrix in figure below where the rows represent the time sample T_0, T_1… T_k and the columns represent the value of spectral channel 1 to N. Each block represents the value of a particular spectral channel at a given time sample.

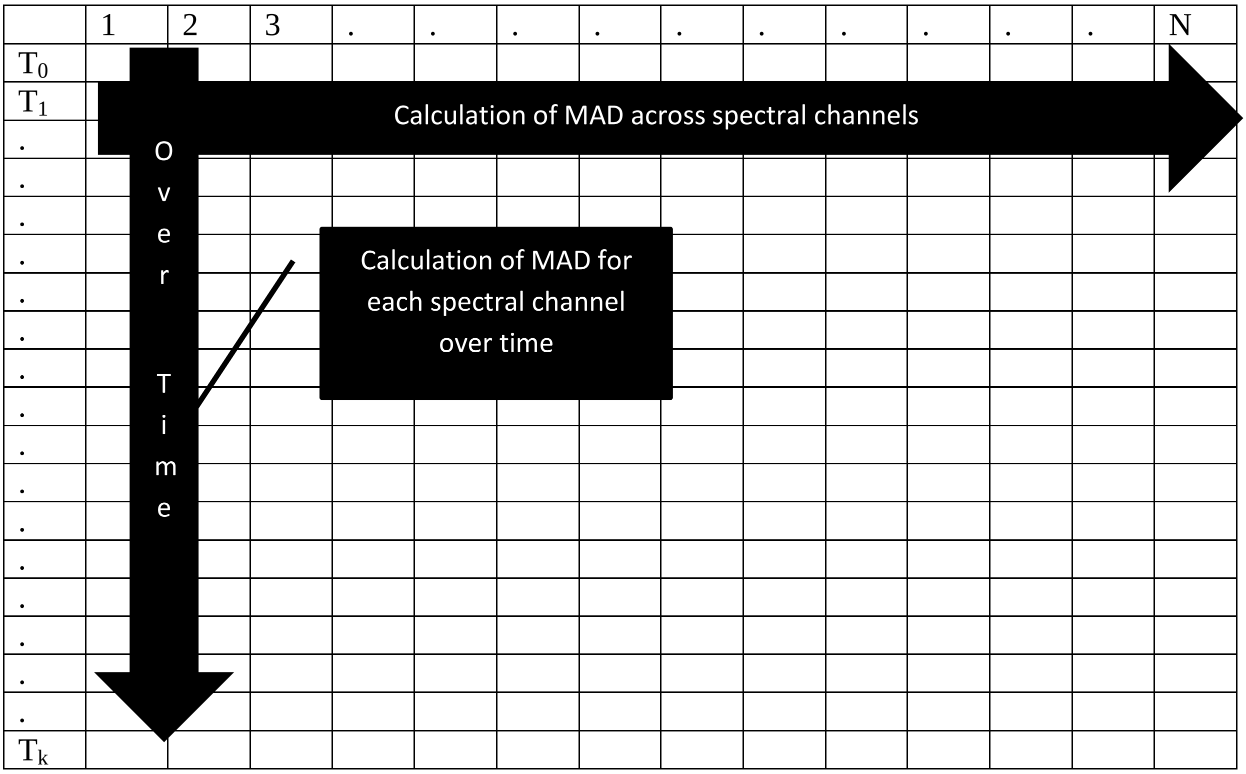


Figure 2: Time-frequency representation for understanding MAD filtering methods

As shown in the above matrix, for a particular time, the MAD is calculated for all the channels and hence the method is called Across Channel MAD calculation. In case of MAD calculation for spectral channel over time, computation is carried out for a particular spectral channel over definite time samples.
The computational complexity is more in the second (channel over time) method.
Also, in case of persistent RFI (duty cycle > 50%), the across channel method performs better and will be shown in the subsequent chapter.

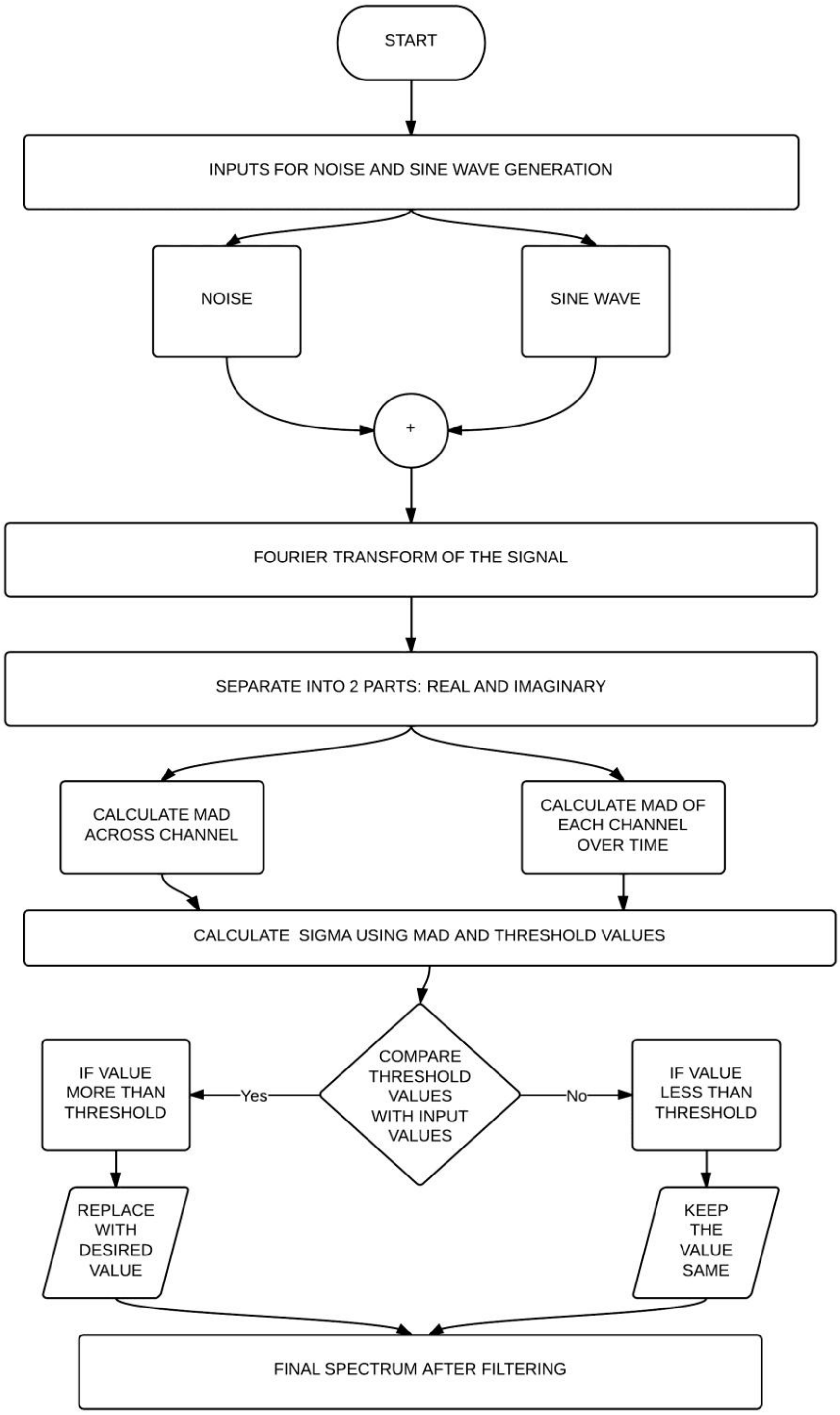


Figure 3 : Flow chart of MATLAB simulation

4. MAD FILTERING ON SIMULATED AND ACTUAL NARROW BAND RFI DATA

4.1 Simulation of Gaussian noise and MAD filtering

In the previous chapter, the method of the simulation of narrowband RFI and filtering using MAD in MATLAB are shown. In this chapter, the results of simulation are shown along with their plots. The results for both the MAD band spectral domain RFI filtering methods are described. Note that no windowing was applied in the simulation tests which might be the reason for spectral leaking in some of the results. Also, as 512 FFT points was performed, only 256 channels were plotted as the rest would be mirror images.

4.1.1 MAD filtering over time

White gaussian noise was generated for 1024 FFT cycles (for a 512 FFT point, 512 time samples are taken for each FFT cycle) and 256 channels. 3 sine waves were added randomly to the signal according to the user inputted duty cycle.
The amplitude and frequencies of the sine waves introduced are as follows:
1.) 12 and 600Hz respectively
2.) 15 and 800Hz respectively
3.) 18 and 2000Hz respectively

The sampling frequency was 8000Hz and the duty cycle was 30% i.e. 308 cycles had sine wave added to them. The FFT channels containing sine wave peak are 38, 51 and 128 channel number respectively. The value of N (aggressiveness parameter) was given as 3. Figure 4 shows the plot before and after the MAD filtering.

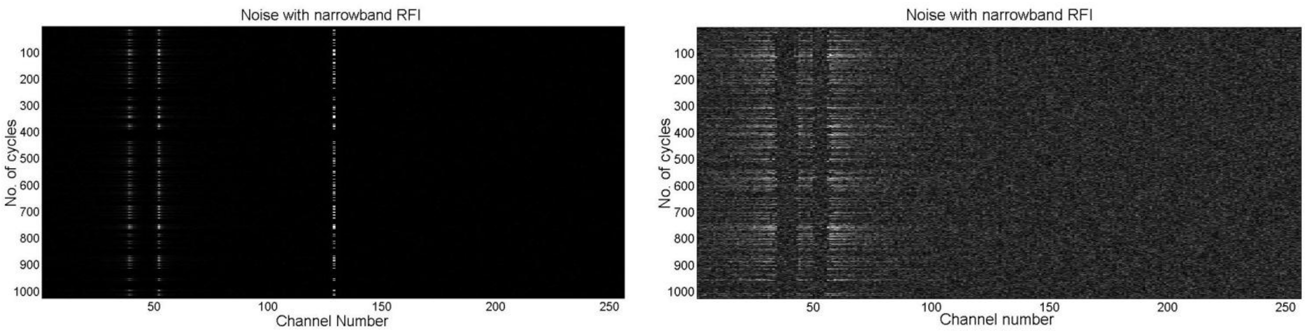


Figure 4 : Over time MAD filtering simulated in MATLAB

As seen in the above figures, the narrow band RFI is removed for all the three channels. In case of channel 38 and 51, the RFI is removed from the channel. However, the adjacent channels which have affected been affected by RFI still show some presence of RFI. The input sine wave frequency is not an integer multiple of the frequency resolution and hence the leakage into adjacent channels. (This is not seen in channel 128 as the input sine wave frequency is an integer multiple of frequency resolution of this channel.)

4.1.2 MAD filtering Across Channel

White Gaussian noise was generated for 1024 FFT cycles (for a 512 FFT point, 512 time samples are taken for each FFT cycle) and 256 channels. 3 sine waves were added randomly to the signal according to the user configurable duty cycle. The amplitude and frequencies of the sine waves introduced are as follows:

1.) 12 and 500 Hz respectively

2.) 14 and 700 Hz respectively

3.) 16 and 3000 Hz respectively

The sampling frequency was given as 8000Hz and the duty cycle was 30% i.e. 308 cycles had sine wave added to them. The FFT channels containing sine wave peak are 32, 44 and 192 channel number respectively. The value of N (boundary limit) was given as 3. Figure 3 shows the plots before and after MAD filtering across the channel.

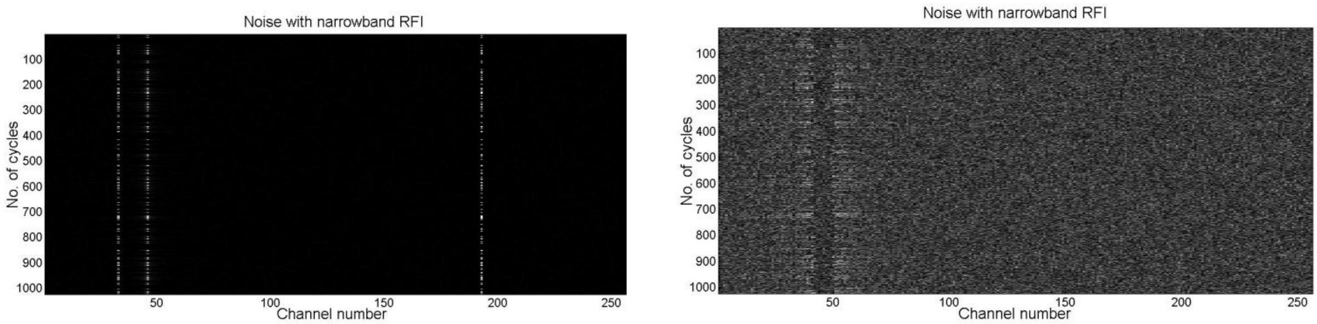


Figure 5: Across channel MAD filtering

As it can be seen from the above plots, the channels 32 and 192 have been deprived of narrow band RFI without spilling in the near channels. Though channel 44 has spread over its nearby channels, the dispersion is not as significant as it was noticed in overtime method.

It was also observed that for duty cycle of more than 50%, MAD filtering overtime ceased to remove the narrow band RFI whereas the across channel MAD filtering could identify the narrow band RFI and replace it with the desired value. This was also one of the consideration for concluding that across channel MAD filtering was more efficient and effective than that compared to overtime approach in case of higher duty cycle of RFI.

4.2 GSB data simulated and corrected with MAD across channel

The raw voltage data was recorded using GMRT Software Backend (GSB) at 235 MHz RF. The data was corrupted by narrow band RFI and both overtime analysis and across channel analysis was carried out in the spectral domain using MATLAB.

Across channel analysis was applied with boundary limit 3. Across channel also failed to identify the narrow band RFI and couldn't correct the signal. When the limit was set to 2.5, across channel approach filtered the RFI and gave the desired result. Figure 6 shows the recorded signal and the corrected signal when MAD across channel was applied with boundary limit as 2.5.

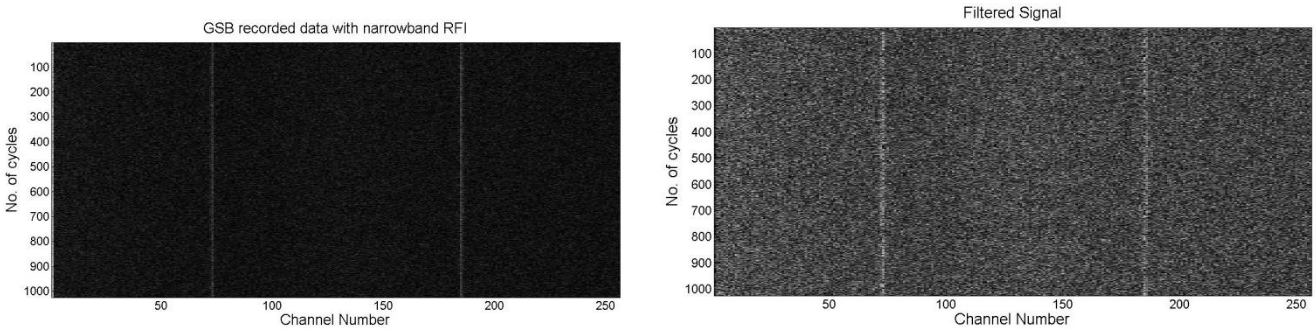


Figure 6 : GSB recorded data before and after MAD filtering

The overtime method could not perform on this data because the narrow band RFI was almost continuous i.e. 100% duty cycle. Figure 7 shows channel 73 across time. Red indicates the signal after MAD filtering whereas blue indicates the original signal which was affected by narrowband RFI. Since, MAD has breakdown point of 50%, this method could not correctly estimate the standard deviation of noise in presence of narrow band RFI.

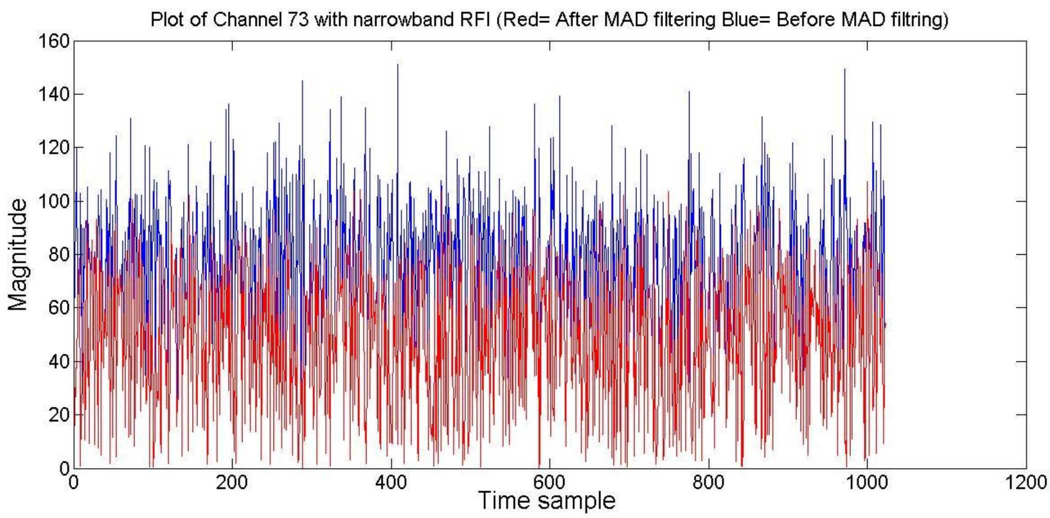


Figure 7 : Channel 73 with 100% RFI

4.3 Radiation Test

A test was carried out were a continuous signal was transmitted from Central Electronics Building of GMRT and the signal was received by one of the central square antennas. The setup consisted of a continuous wave generator and a noise generator programmable as per user requirement.

Initially, a 610 MHz signal was generated and transmitted. A sudden pulse of power -10 dBm was introduced and the signal was recorded using the GSB in the phase array mode. The starting frequency for the antenna was 603MHz. The total on time of the pulse was 4secs and the sine wave was inserted for duration of 32 msec.

The recorded data was in 16-bit signed binary format. Hence, it had to be converted first to ASCII before MAD filtering in MATLAB. Figure 5 shows the recorded signal and MAD filtered signal.

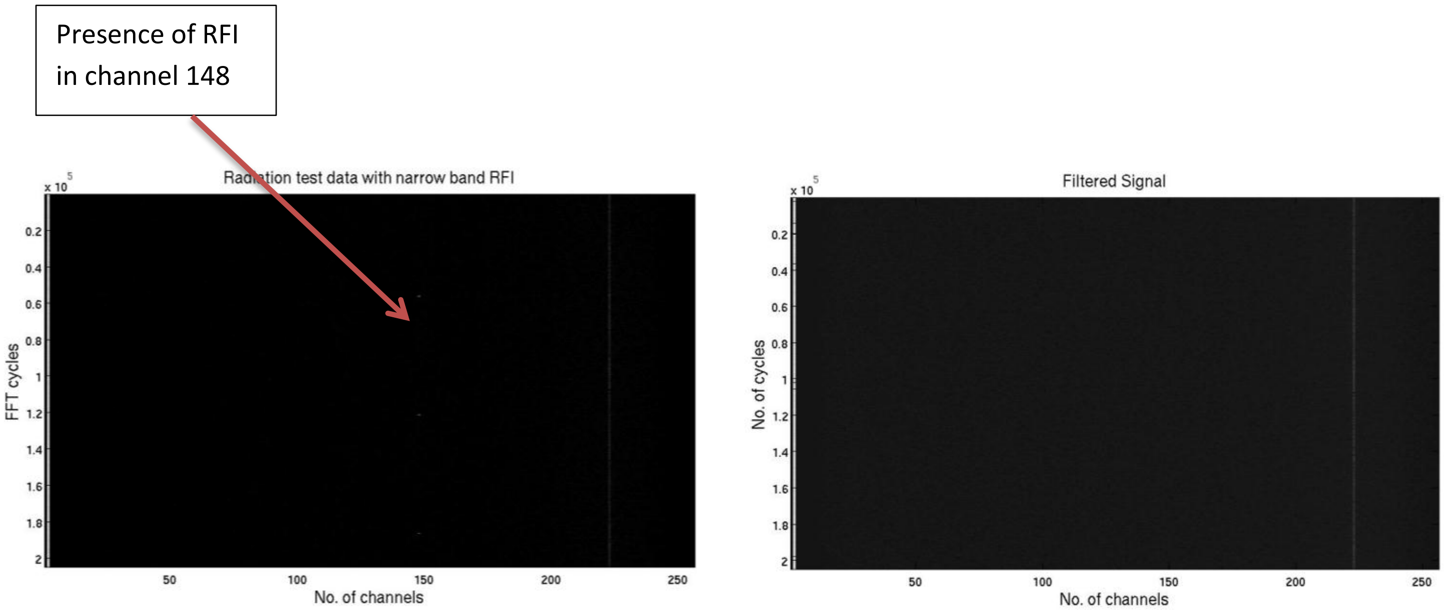
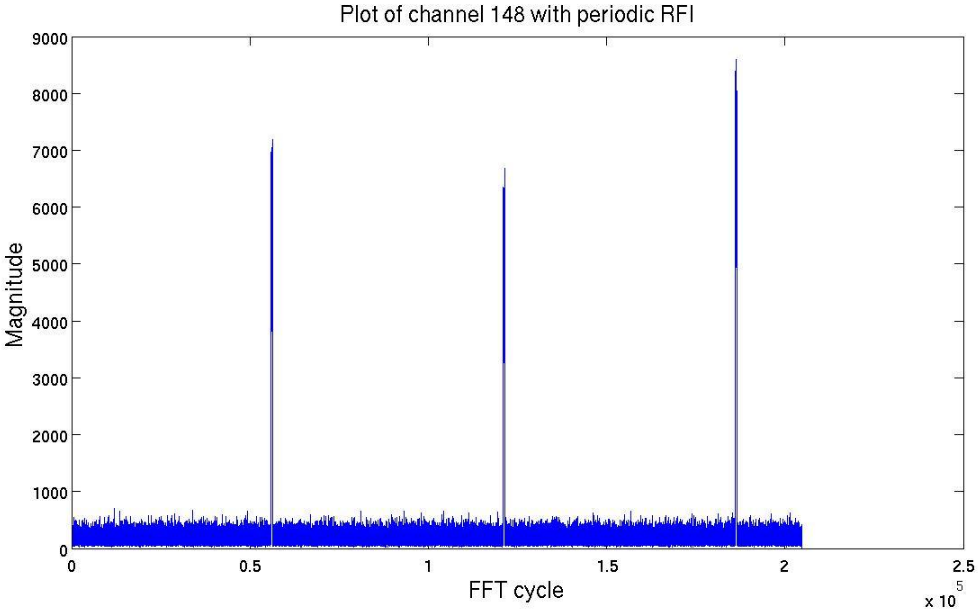


Figure 8 : Radiation test plots before and after MAD filtering

Spectral channel number 148 consists of narrowband RFI. Both MAD across the channel filtering and MAD over time filtering give the same result. A plot of channel number 148 is given in figure 6. The original signal plot is shown along with the MAD corrected plot. The peaks which is narrow band RFI has been removed and replaced by a white gaussian noise having statistics similar to the underlying noise.



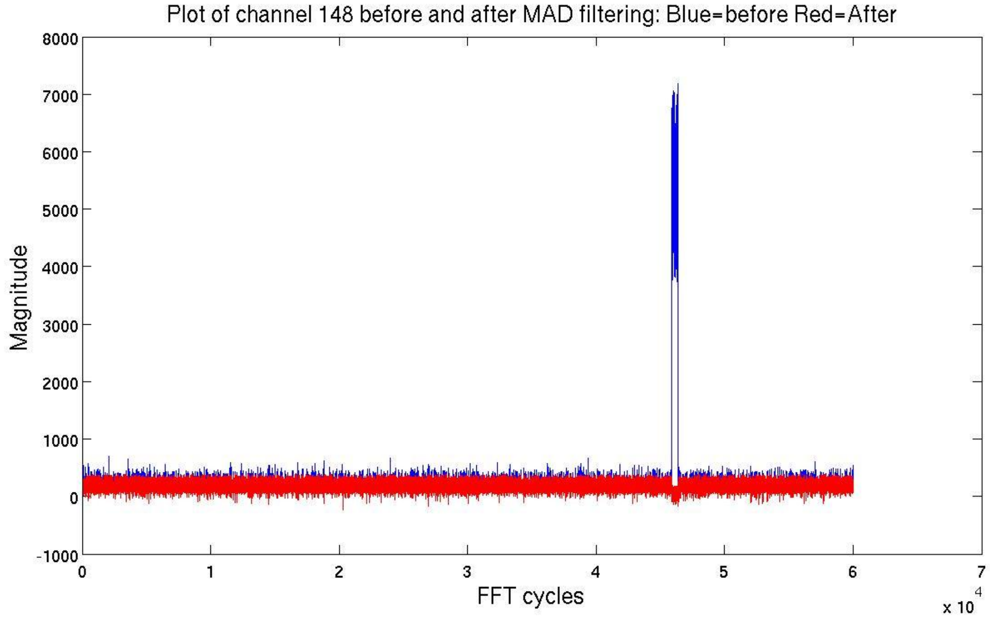


Figure 9 : Plot of channel no. 148 before and after MAD filtering

Variation in the statistics of data was observed for the radiation test. Spectral channel 148's histogram was plotted before and after MAD filtering. The histograms are shown in figure 10 and 11 respectively. The variation in mean was also observed. The mean of the channel before MAD filtering was 515.3844 and after MAD filtering was 174.7623. This shows the robustness of MAD estimator.

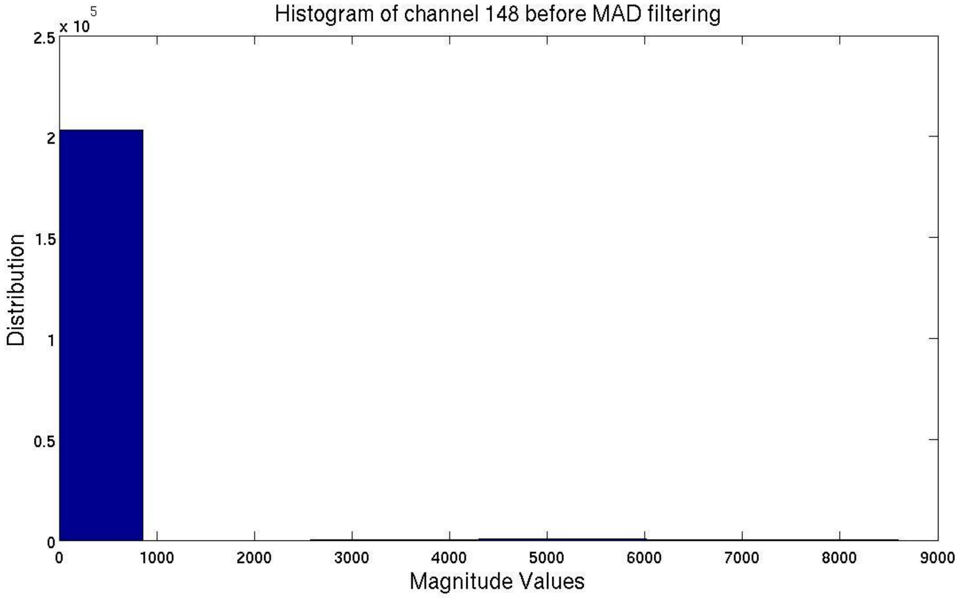


Figure 10: Histogram of channel 148 with RFI in it

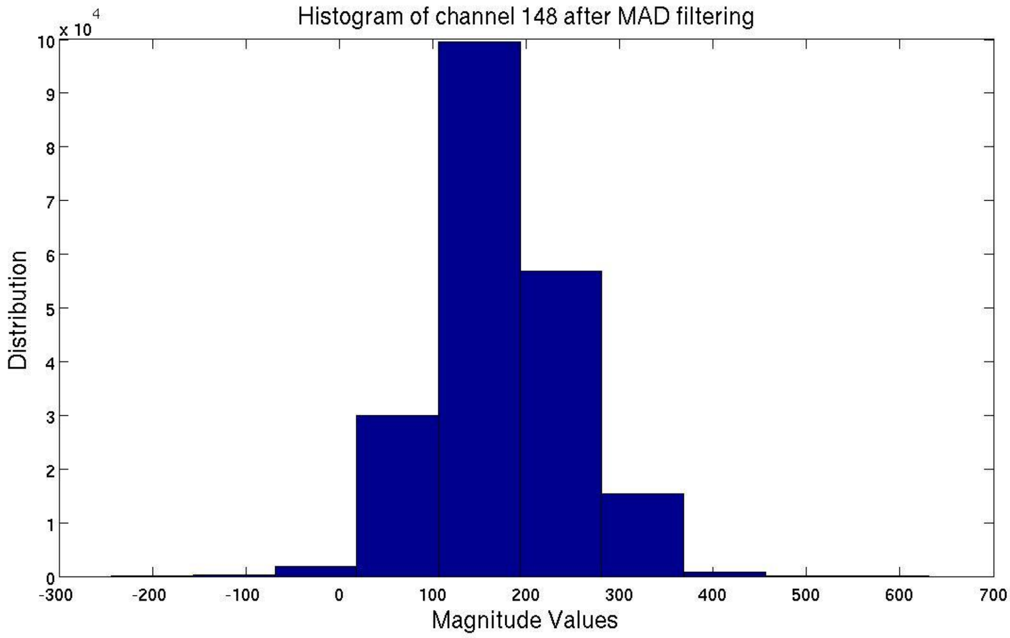


Figure 11 : Histogram of channel 148 without RFI

Hence, from the above simulation and radiation tests, we can conclude that MAD being a robust estimator, is effective in removing narrow band RFI in data having significant Interference to Noise ratio. Moreover, across channel MAD filtering is an effective way for real time mitigation of narrow band RFI.

Implementation of MAD in real time is computationally challenging. It needs to be implemented on either GPU or FPGA based digital backend system which is being developed as a part of GMRT upgrade. The next chapter gives an insight on the method of median implementation on FPGA and the approach suitable for real-time median computation.

5. OVERVIEW OF MEDIAN COMPUTATION TECHNIQUES FOR FPGA IMPLEMENTATION

5.1 Introduction

Programmable Logic Devices (PLDs) are semiconductor devices that can be configured by the end user. In other words, their functionality is not defined during manufacturing, and can be programmed to perform a large number of different functions [9].

Field Programmable Gate Array (FPGA) is a PLD that uses logic cells, and consists of basic gates. FPGAs can implement any logic function as long as the resources required by the function are available. It consists of flip-flops, multiplexers, arithmetic and carry logic, and function generators. Although logic cells are used to implement the logic, they can also be used as distributed memory.

Real-time signal processing applications employ filtering to process and to manipulate the signals, or to remove noise from data. Median filter is a non-linear filter used for removing impulsive noise from data. In this chapter, various median computation techniques for FPGA are mentioned. MAD is basically recursive median and hence it is important to find a suitable method of implementing median computation for FPGA which can be later upgraded to MAD computation[10].

5.2 Median Filters

There are two types of filters: linear filters and non-linear filters. The median filter is a non-linear filter; it is a special case of rank order filters whose rank is half the length of the sequence. The median filter is a highly versatile non-linear filter that has been used extensively in a variety of domains. Its strength lies in its ability to filter out noise while minimally affecting the properties of the underlying signal. The median filter replaces a sample with the middle ranked value among all the samples within the sample window, centred on the sample. It filters out samples that differ greatly from the expected signals; i.e. outliers. There are various methods of implementing median filtering on FPGA. There are two main methods. The first is to maintain the input sample list in its original order and then sort it through a sorting logic (the sorting network architectures). The median value is then extracted from the mid-position in the sorted list. The second method involves sorting the samples as they enter the system (the histogram method).

5.2.1 Sorting Based Median filtering

Sorting network architectures can be implemented by bubble sort, quick sort or insertion sort logic. These sorting logics use compare and delay units to implement the median filter. The incoming data is passed through a cluster of comparators and swapping units - the comparator units compare two to three incoming data at once and then the swapping unit sorts them accordingly. The median value is the middle value of the sorted data.

To implement a 3x3 median filter using bubble sort, compare-and-swap units are required. As the size of the window increases, the number of compare-and-swap units required for implementation will also increase. The resources required to implement the sorting network architecture on a FPGA device increases with the size of the filtering window. An important aspect of the sorting based algorithms is that it is independent of the size of the signal and depends only on the size of the window [10].

Bubble sort is a simple sorting algorithm that works by repeatedly going through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in wrong order. Insertion sort is a simple sorting algorithm that builds the final sorted array (or list) one item at a time. It is much less efficient on large lists. Insertion sort iterates, taking one input element at each repetition, and building a sorted output list. It removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remain. Quicksort is a divide and conquer algorithm. Quicksort first divides a large list into two smaller sub-lists: the low elements and the high elements. It then recursively sorts the sub-lists.

5.2.2 Histogram Based Median filtering

Histogram is a representation of the distribution of the intensities. A histogram is generated by incrementing the value of the bin representing the corresponding intensity level whenever that particular intensity is encountered. The implementation of the histogram requires as many counters as the number of intensity levels. Each counter is associated to an intensity level and the values of these counters are incremented according to the received intensities.

Another method for computation of median using histogram is calculating the cumulative histogram of the incoming samples. The cumulative histogram is calculated by increasing the values of all the bins greater than or equal to the incoming intensity. Then, after building the cumulative histogram, the first bin which has a value greater than or equal to the median index is taken as the median value. Histogram based median filtering depends on the size of the window.

5.3 Comparison of Sorting Based and Histogram Based Median Filtering

Histogram based median filtering can be implemented in large one-dimensional windows whereas a sorting based median filtering would require atleast a two- dimensional window to be implemented. Now, as the size of the data increases, the sorting based algorithm would require large space and time to compute median. The number of registers and input sorters required to implement the sorting algorithms will significantly increase as the window size increases. Therefore, the time cycles required would increase as the registers and input sorters increases. In case of histogram based filtering, the only aspect to be taken into account is the proper increment and decrement of the bins as data flows in and out of the system. The order of each of the sorting and histogram algorithms is given in the table below:

Type of Median Filter	Order of the filter
Bubble sort Median filter	$O(N^2)$
Insertion sort Median filter	$O(N^2)$
Quick sort Median filter	$O(N \log N)$; worst case: $O(N^2)$
Histogram based Median filter	$O(\log N)$

From the above table we conclude that, as the value of N increases, the sorting filter's complexity increases exponentially. Hence, we conclude that histogram based filter is the best suitable architecture for GMRT where real time filtering is required and a large window size is required.

5.3 Histogram Based Median Filter

5.3.1 An Overview

Histogram based median filter is constructed by creating a cumulative histogram of the input data. Every possible input value has a bin associated to it in the histogram. The filter accepts a new data at very clock cycle and so the histogram would keep getting updated after every clock cycle. The count values for each bin should be compared with the median index parallel to all other processes occurring on the FPGA system. The first bin whose count value exceeds the median index would give the median value. This method has no sequential processing mode and hence would be able to give median values in real time as new data keeps arriving.

5.3.2 Implementation

Since, many concurrent processes occur in a single clock cycle, a rank of parallel bin nodes is created. A separate register is also required to keep a count for each of the possible input values.

Now, as all the bin nodes should be updated in parallel, bin nodes are equipped with its own incrementer. The incrementer increases by a value of 1 when the bin accepts a data. Each bin also consists of an enable input (enable), median index register (med_reg) and an output node (o/p) which gives single binary value i.e. 0 or 1 as output. The enable input determines whether the bin value should be incremented in the on-going clock cycle and the median index keeps the count of the number of values stored in the bin. The output node gives the output as 1 when the value of the median index input equals or exceeds the median index and 0 otherwise.

A circuit composed of such histogram bin nodes are the input signals of the cumulative histogram. The first bin value to have an output of 1 from its output would give the median value for that particular clock cycle.

The output of each of the bin nodes go to a priority register whose main function is to determine which bin node has greater value than that of median index and determine the median value for that clock cycle subsequently.

FPGA boards consist of block RAMs which can be used as ROMS to store bin patterns of the cumulative histograms. This would help to send the incoming data to its associated bin. Also the ROM ensures that correct bins are enabled for any input sample. Figure 12 shows a simple diagram of the histogram with its parts.

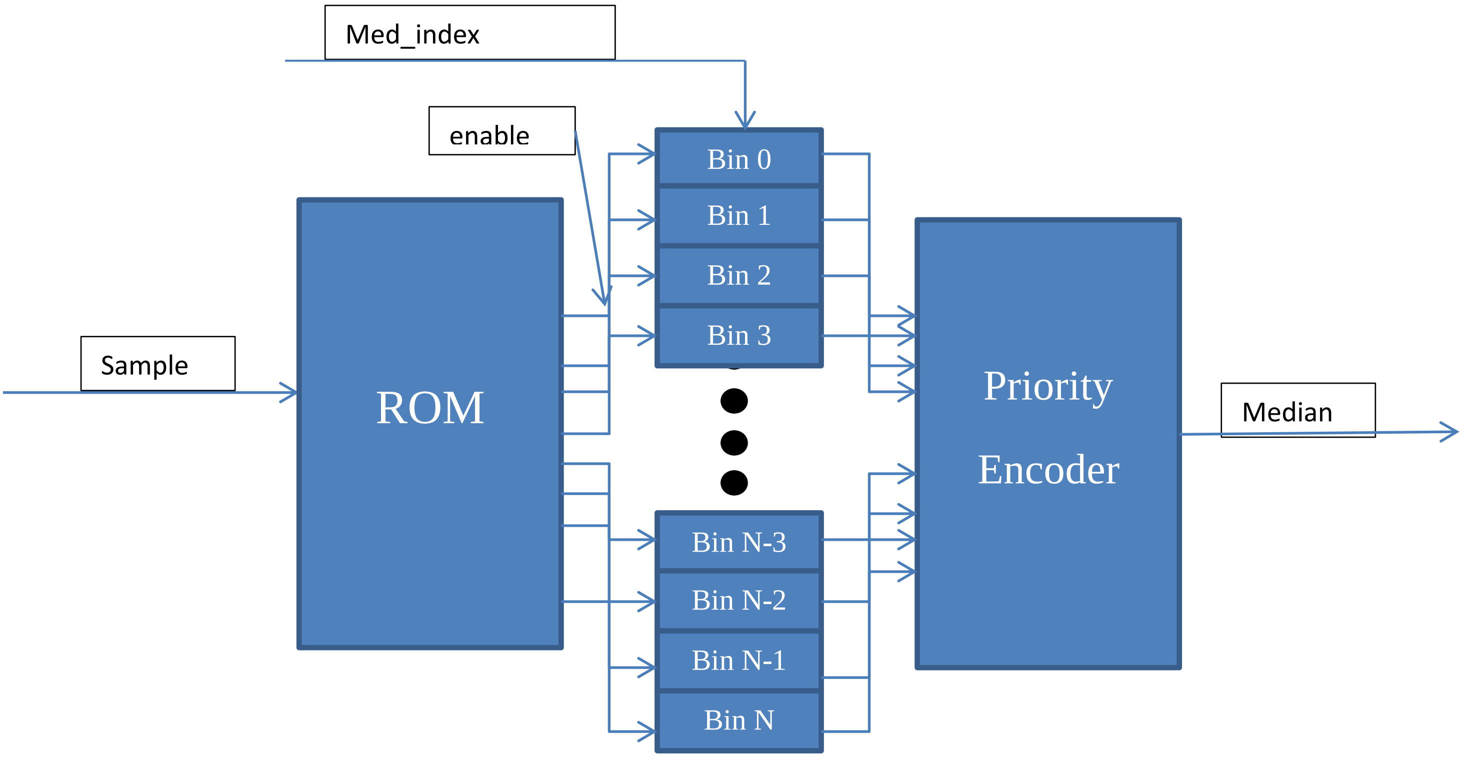


Figure 12: Histogram based median filter architecture

5.4 Sliding Window Implementation

In the above implementation method, the system takes a sequence of sample and returns the median for a fixed window after its filled. The median must be computed for a sliding window which means for each cycle the window moves one sample down the sequence, discarding the oldest sample and adding a new sample to the window. Also, it should happen only after the window is full for the first time and then keep updating as new samples arrive.

An implementation of this would require a FIFO (First In First Out) buffer to store the samples for the window over which the median must be found. When the new sample arrives, update of histogram should include all bins corresponding to the access pattern for the oldest sample to be decremented and the corresponding bin of the new input sample to be incremented. This should be done in one cycle. It can be done by updating the histogram bins in a fashion where only the bins associated with the outgoing and incoming sample be decremented and incremented respectively. This will ensure that the histogram is up-to-date in every clock cycle and there would be no need for a pause. In this implementation, the bin should consists of an incrementer as well as a decrementer so that when data is received, the median index should be incremented and when the data is discarded , the median index should be decremented.

The RAMs should be dual ported as it would enable signals for both the new and old sample access the pattern in ROM paralelly.

.

6. CONCLUSION

MAD based spectral domain RFI filtering technique for removal of narrow band RFI was discussed. The technique was applied to simulated and actual telescope data and the results were discussed. Two variants in the technique of spectral domain MAD calculation (channel over time and across the channel) were described and analysed. It was found that across the channel method is superior in case the duty cycle of RFI is more than 50%. Also, in terms of computational complexity, across channel seems to be a better option. The initial tests were carried out in simulation using MATLAB. Spectral MAD filtering using actual data was carried out on GSB raw voltage data and radiation tests. The quantitative efficacy of the algorithm has been verified by calculation the Interference to Noise ratio for some of the initial tests on simulated data. Finally, various techniques for real time calculation of median (and MAD) on FPGA were studied. It was concluded that the histogram based method is optimal for an FPGA based real time MAD calculation.

7. FUTURE WORK

In order to implement the mentioned technique within a digital backend system, following work needs to be carried out:

1. Equalization of band shape (in the corresponding real and imaginary parts) during MAD based filtering (especially in the across channel method).

2. Understanding the leakage of RFI and its removal in the adjacent spectral channels.

3. Architecture and implementation of real time MAD calculation on FPGA using the histogram based technique.

4. Calculation of accuracy of across the channel MAD estimation as a function of the number of spectral channels.

5. Analyse the performance MAD filtering on 8-bit spectral data (real and imaginary) - for FPGA implementation.

APPENDIX

I. INTERFERENCE TO NOISE RATIO

The mitigation schemes of RFI work on short sets of data. It implies that a narrowband RFI can be either excised or cancelled only when its Interference to Noise ratio (INR) is high or above a particular value. In MAD based filtering, the INR is an important factor due to the aggressiveness parameter (as discussed in the above chapters). Hence, knowing the INR ratio is of primary importance. Here, the INR ratio was calculated using the formulae:

$$INR = \frac{\sigma_{RFI}^2}{\sigma_{Noise}^2}$$

Using MATLAB simulations, first the standard deviation of the noise was calculated for all the FFT cycles and an average of its value was taken. It formed the σ_{noise} for our calculation. Then the standard deviations for each channel with sine wave corrupted data were calculated. The real and imaginary parts were considered separately and a plot of the INR ratios before and after MAD filtering was plotted. The plots are given below.

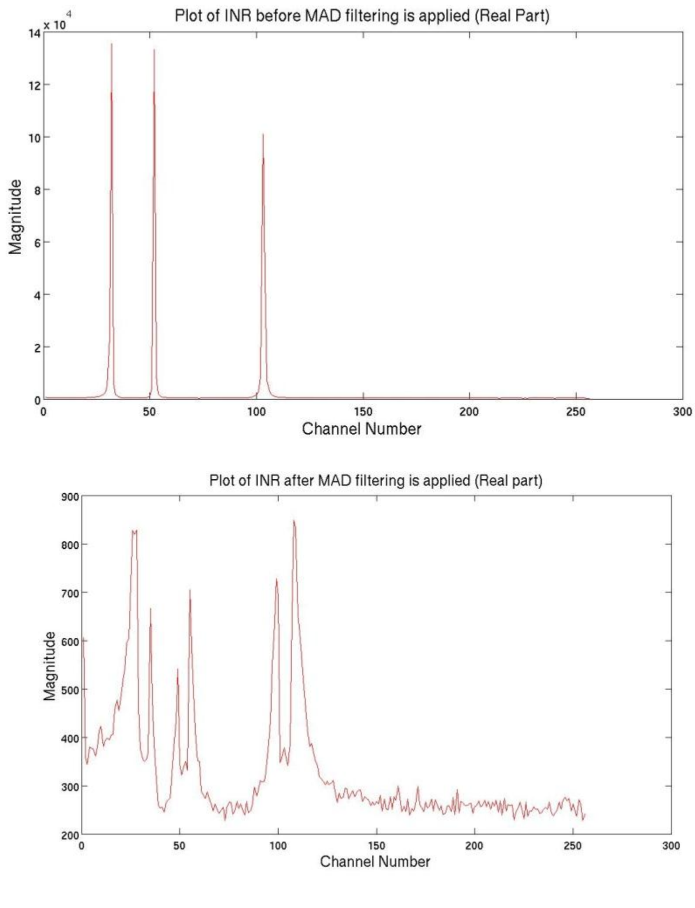


Figure 13 : INR for real part before and after MAD filtering

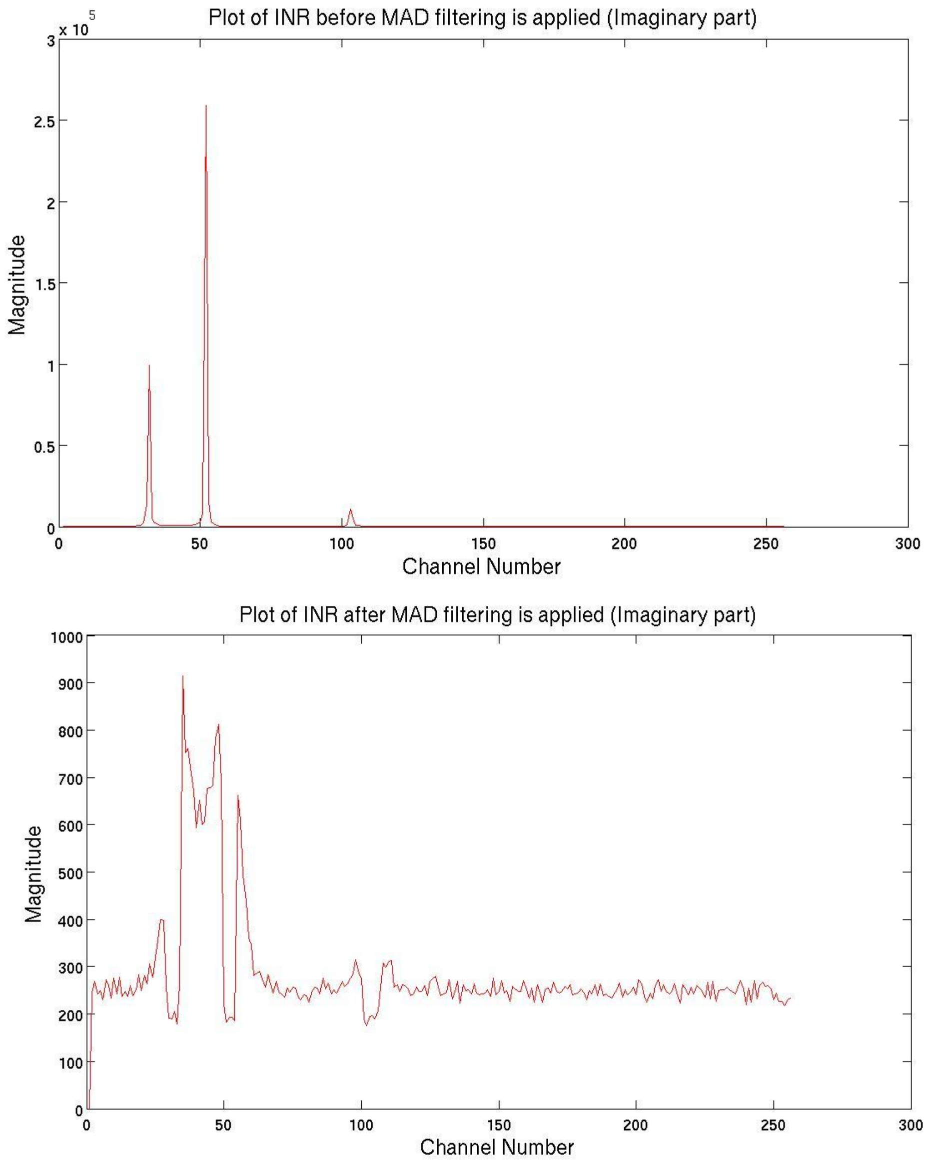


Figure 14: INR for Imaginary part before and after MAD filtering

REFERENCES

[1] B. Ajithkumar, S.C. Choudhari, K.D. Buch, M.V. Muley, G.J. Shelton, S.H. Reddy, S. Kudale, J.Roy and Y. Gupta, "Next Generation Digital Backends for the GMRT" ,TIFR, 2011

[2] Buch, K.D.; "Development of FPGA based wide-band digital backends for the Upgraded GMRT", TIFR, 2011,

[3] REC. ITU-R RA2126, "Techniques for mitigation of radio frequency interference in radio astronomy", 2007

[4] Kesteven, Micheal, "Current Status of RFI mitigation in Radio Astronomy", 2009

[5] Jiang, Zhi, "Mitigation of Narrow Band Interference on Software Receivers based on Spectrum Analysis",2009

[6] R, Prassana, "Study of RFI Filtering Algorithms for the GSB", GMRT, 2009

[7] Swarup, Govind, "Power-line Radio Frequency Interference at the GMRT", TIFR, 2007

[8] Baan, Willem A., "RFI Mitigation in Radio Astronomy", RFI mitigation workshop - RFI2010, 2010

[9] Gundam, Madhuri, "Implementation of Directional Median Filtering using Field programmable Gate Arrays" 2010 ,University of New Orleans Theses and Dissertations

[10] Fahmy, S.A.; Cheung, P.Y.K.; Luk, W.; "High- throughput one-dimensional median and weighted median filters on FPGA", 2008, IET Computers & Digital Techniques