Addition of New features in offline processing Chain for Packetized Beamformer

By

Swapnil Vir Lal

Under the guidance of

AJITHKUMAR B
KAUSHAL BUCH

GIANT METREWAVE RADIO TELESCOPE
NATIONAL CENTRE FOR RADIO ASTROPHYSICS
TATA INSTITUTE OF FUNDAMENTAL RESEARCH
KHODAD, DIST. PUNE - 410 504
MAHARASHTRA, INDIA
December 2013

ABSTRACT

In GMRT, an incoherent Beamformer is implemented on FPGA platform, plays a vital role in observing a pulsar with higher sensitivity to get the pulsar profile. The design uses the packetized correlator design of F-X Engine for 4 Antennae and works at bandwidth of 400 MHz on ROACH Boards. Offline processing of data from the Packetized beamformer design successfully handles only one polarization at an instant. So, this can be considered as a limitation of the previous work. As a part of GMRT backend up gradation system, the goal of this project is to upgrade the offline processing chain. This work briefly describes the logic to process both polarization and to do offline long term integrate up to the n number of cycles which would be provided by the user while compiling. The testing of the scripts is done with noise. Results are also attached with this report to ensure that Beamformer design is functioning properly with both polarization and long term integration is being done on it.

Acknowledgements:

I owe a great many thanks to a great many peoples who helped and supported me during my project work.

My deepest thanks to Prof. Yashwant Gupta, Dean of GMRT for giving me such a great opportunity to work in GMRT during my semester vacations.

I express my immense gratitude to the Group Co-ordinator (Digital Back-End) Mr. Ajithkumar B, for giving me a valuable opportunity by assigning this project tasks. His Guidance, support, encouragement and motivation at every step of the project was the key to the completion of this project tasks.

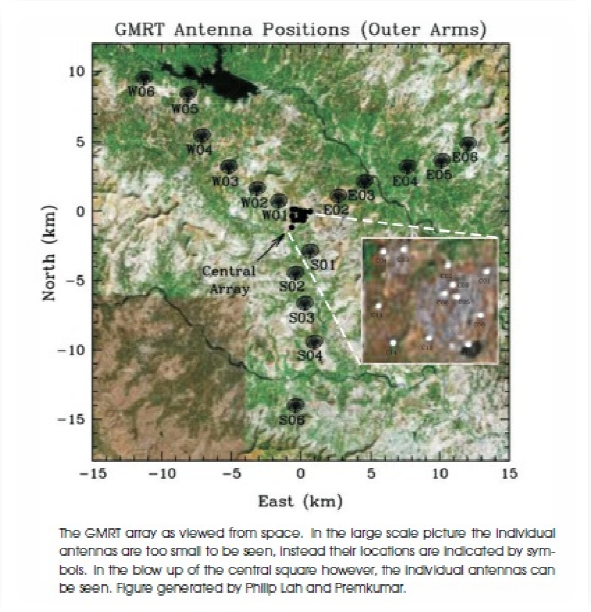
I would like to express my sincere thanks to my guide Kaushal Buch, in guiding me right from the inception till the completion of this project. His constant encouragement, dedication, innovative ideas and moral support helped me in maintaining the progress and complete the project within prescribed time-frame.

I would like to acknowledge Mekhala Muley and Sandeep Choudhari for extending their technical guidance at all stages of the project.

I would also thank to entire Digital Back-end Team for their operation and for spending their quality time which helped me in during the project.

January 2014 **Swapnil Vir Lal**

Introduction: GMRT is the world's largest array of Radio Telescopes at meter wavelength and it is operated by National Centre of Radio Astrophysics, a part of Tata Institute of Fundamental Research. Currently, GMRT consists of 30fully steerable telescopes based on '**SMART'** (Stretch Mesh Attached to Rope Trusses) Concept, each of 45meters in diameter spread up to 25km. Each Antenna contains 4Feeds and operates on 5 Frequency Bands, centred as 153, 233, 327, 610 and 1420MHz.All these provides Dual Polarizations output. The positions of Antenna are as shown in figure below: GMRT can act as an interferometer which uses a technique known as aperture synthesis to make images of radio sources. To provide seamless coverage from 100 MHz to 1600 MHz, upgrade in Digital backend Electronics must require with an upgrade in mechanical and servo control systems. Two possible solutions to the backend upgrade are currently being developed – one based on multiple FPGA boards, second on GPU cluster.

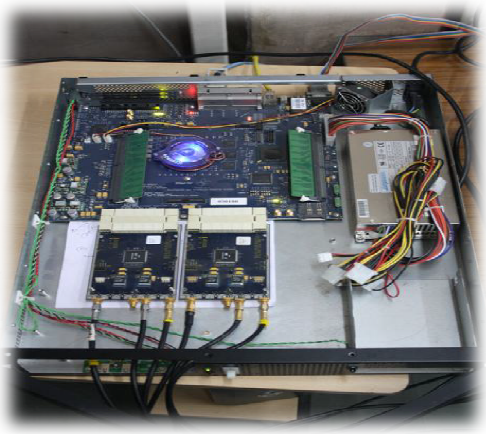


The GMRT array as viewed from space. In the large scale picture the individual antennas are too small to be seen, instead their locations are indicated by symbols. In the blow up of the central square however, the individual antennas can be seen. Figure generated by Philip Lah and Premkumar.

Currently, the GMRT is undergoing an upgrade. As a part of upgrade, the GMRT plans to increase the bandwidth of GMRT from present value of 32 MHz to about 400 MHz and also plans to upgrade the digital backend from GSB (GMRT Software Backend) to FPGA and GPU based backend.

Description about Previous Work: Digital backend mainly deals with signal processing of data received from Telescopes, used in interferometer and beamforming modes. The digital signal is processed through FX Correlator (FX: FFT followed by Multiplier) to generate cross amplitude and phase information between each pair (baseline) among the 30 antennas to give the visibility information. This data is used in imaging, continuum and many other astronomical observations.

As a part of Digital backend up gradation process of GMRT, An incoherent Packetized Beamformer was designed during June to Nov 2013.In Radio astronomy, Beamforming is a technique which is used to get the pulsar profile. It can be of two types such as, Incoherent Beamforming Mode and Coherent Beamforming Mode. The Incoherent Beamformer adds voltage signals from different antennae and computes the basic self term of voltage signals of the two polarizations. This Incoherent Beamformer for 4 antennae and 2 orthogonal polarizations is implemented on a multiple ROACH-boards (FPGA platform) and tested with proper pulsar source.



For the Design, CASPER based FPGA is used as shown in fig:

The Centre for Astronomy Signal Processing and Electronics Research (CASPER) is a global collaboration dedicated to streamlining and simplifying the design flow of radio astronomy instrumentation by promoting design reuse through the development of platform-independent, open-source hardware and software. The CASPER tool flow is better known as the MSSGE (Matlab/Simulink/System Generator/EDK/ISE) or Bee XPS tool flow.

Packetized Beamformer Specifications:

- Number of antennas: 4
- Polarization: Both polarization

- Number of spectral channels: 512
- Number of F engines: 4
- Number of X-engines: 8
- Number of spectral channels per X-engine:64
- Networks used: 1Gbps, XAUI link and 10Gb Ethernet
- Clock Frequency: 800 MHz
- Bandwidth : 400 MHz
- Base integration time: 0.163 milliseconds
- Data rate from 1 X-engine: 27.19 Mbps
- Data rate from 8 X-engines: 223.2 Mbps

4 Antenna Packetized Beamformer Design: 4 antenna packetized Beamformer uses four F-engines and 8 X-engines. Below Figure shows the function performed by an F-engine and also shows after which stage the signal for beamforming is taped.

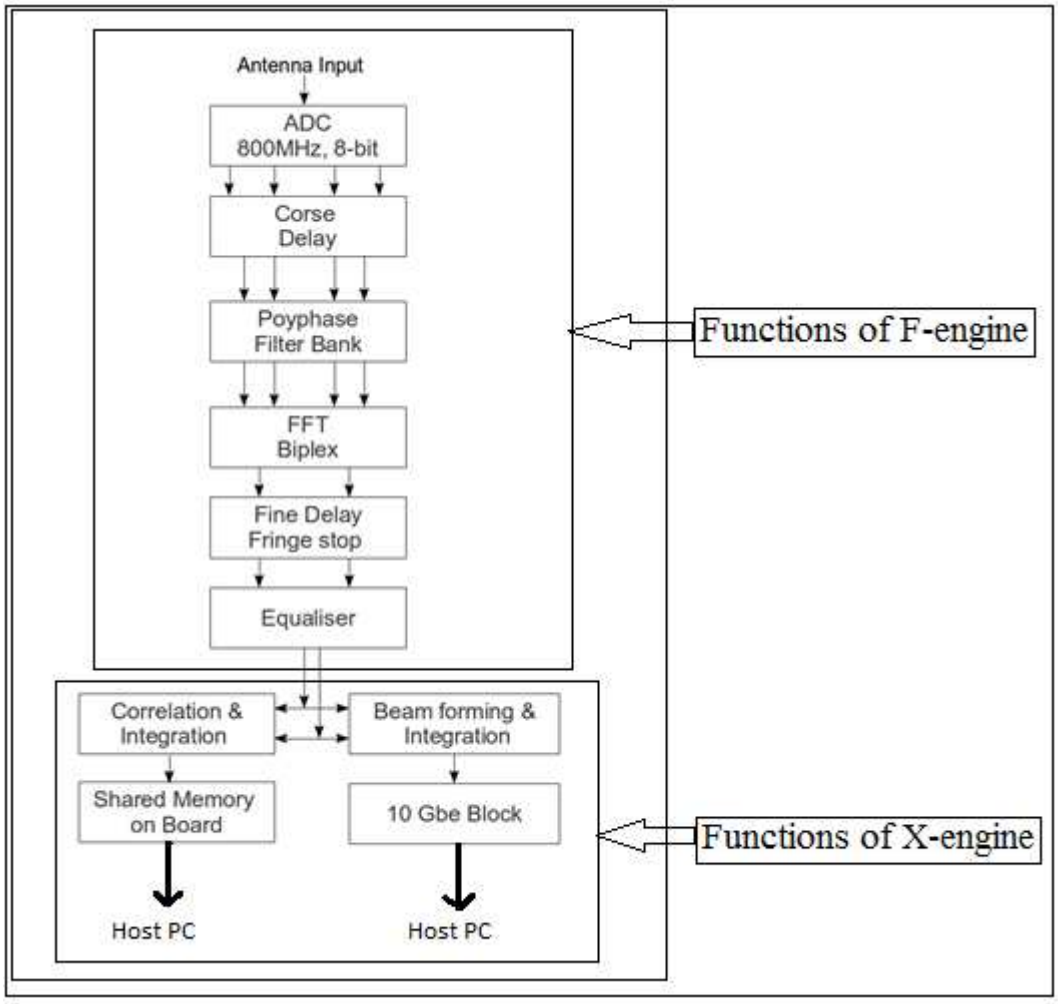


Figure: Functions performed by an F-engine

Note: For detailed working of the above design, Refer to the report developed by Pranjali Chumbhale and Shreya Shetty.

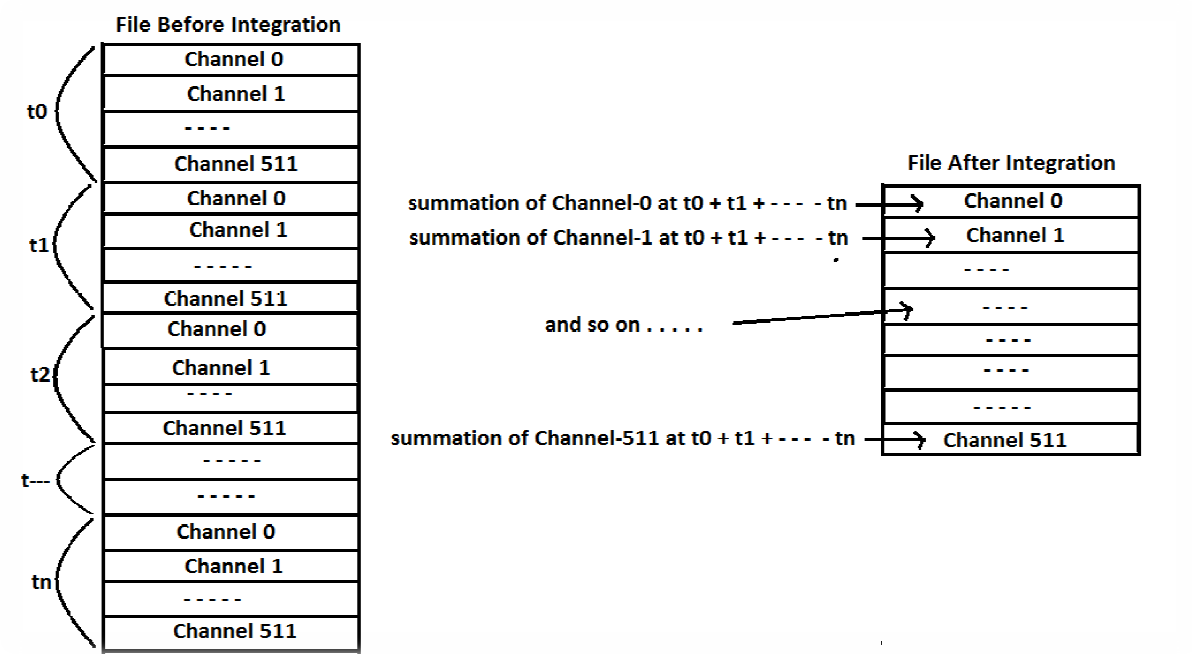
Adding different features to the offline processing chain: My motive of the project is divided into two main parts as follows:
1. To provide the integration
2. To process both polarizations simultaneously.

Part 1: Adding Long Term Integration

In previous work, the script does not process multiple time stamps. So, to get more clear profile of PULSAR, we need to add the multiple time stamps. Hence, integration is necessary. We can do integration at any one stage from the following two stages (refer SOP):

1. Before interleaving of the packets, i.e. after separation of 8 packets which are captured in gulp. But in this case, channels are not arranged in a sequential manner because the packets are arriving from different X-engines. So, we need to do integration of all 8 packets and then interleaving should be done.
2. We can also do the integration after the interleaving of 8 files of different x engine. But at this stage, we need to integrate only one file.

In this work, integration is done after interleaving of all the files. Before integration, the file will contain all the channels at different time stamp as shown in fig below:



So, to add all channel 0, channel 1 till channel 511 of different time stamps, logic is developed in the scripts which are attached in the end of this report. Two File pointers are used, one is used to read the file and 2nd one is to write the addition in file. All the channels are stored in one array by reading from the input file i.e. the interleaved file. And then they are added and stored sequentially in different array and printed in an output file. This output file contains the integrated channels which can be plotted using GNU plot. The program is designed such that it will ask the user to enter the number of integration required for his observation. The program is well tested on noise and outputs are also attached with this report.

Part 2: Processing Pol0

Also, previous works processes only one polarization i.e. pole 1. It is not processing the 2nd polarization after capturing into the gulp. So 4 scripts are developed to process both polarizations. One script is to print the both polarization in the 16 new files. And the other two scripts are to interleave 16 files into 2 different files i.e. one file contains pole 0 and 2nd file contains pole 1. And 4th script is to add both the files which contain polarization 1 and polarization 2.

Syntax of the commands used for the compilation of scripts:

Till acquiring the data from Antenna/Noise source is same as in the previous project report.

1. **Converting data from binary to ASCII**

File name: for_pmon_pol0_pol1.c: this script processes both the polarizations and generates 16 text files received from different X-Engines.

Syntax to run the script:

./pole12 <input file name.dat> <packet size> <scaling factor>

Example: ./pole12 example 554 4

2. **Interleaving of both polarizations:**

Interleaving of Polarization-Q: This scripts generates a single text file coming from different X-engines.

File name:pol1.c

Syntax to run the the script:

./interpol1 > <outputfilename.txt>

Example: ./interpol1 > pole1.txt

File name:pol0.c

Syntax to run the the script:

./interpol0 > <outputfilename.txt>

Example: ./interpol0 > pole0.txt

3. **Integration of Cycles:**

File name: intgeration.c: This scripts asks a number of cycles to be integrated from the user.

Syntax to run the the script:

Note: this script is working only after head command.
./intgeration <input file name.txt> <outputfilename.txt>

Example: ./intgeration pole1.txt intgpole1.txt

4. **Addition of both polarizations:**

File name: addp.c: This scripts adds the two files which contains polarization I and polarization Q

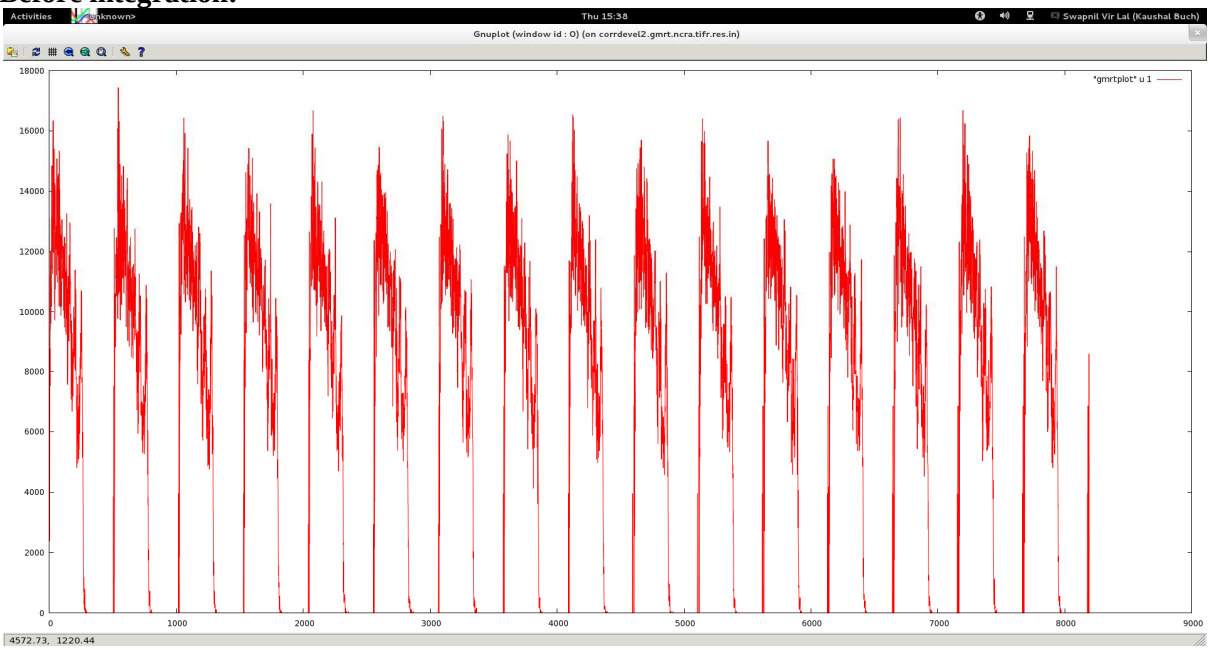
Syntax to run the the script:

./adpolel1-2 <input file name.txt> <input file name.txt> <outputfilename.txt>

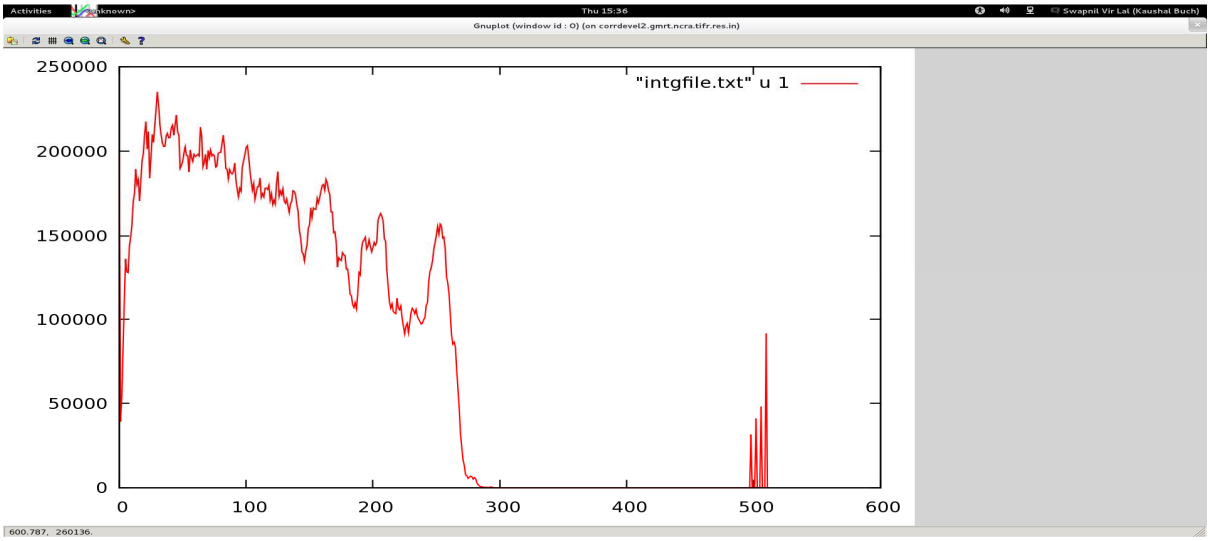
Example: ./adpolel1-2 pole0.txt pole1.txt addpole12.txt

Snapshots of Results:

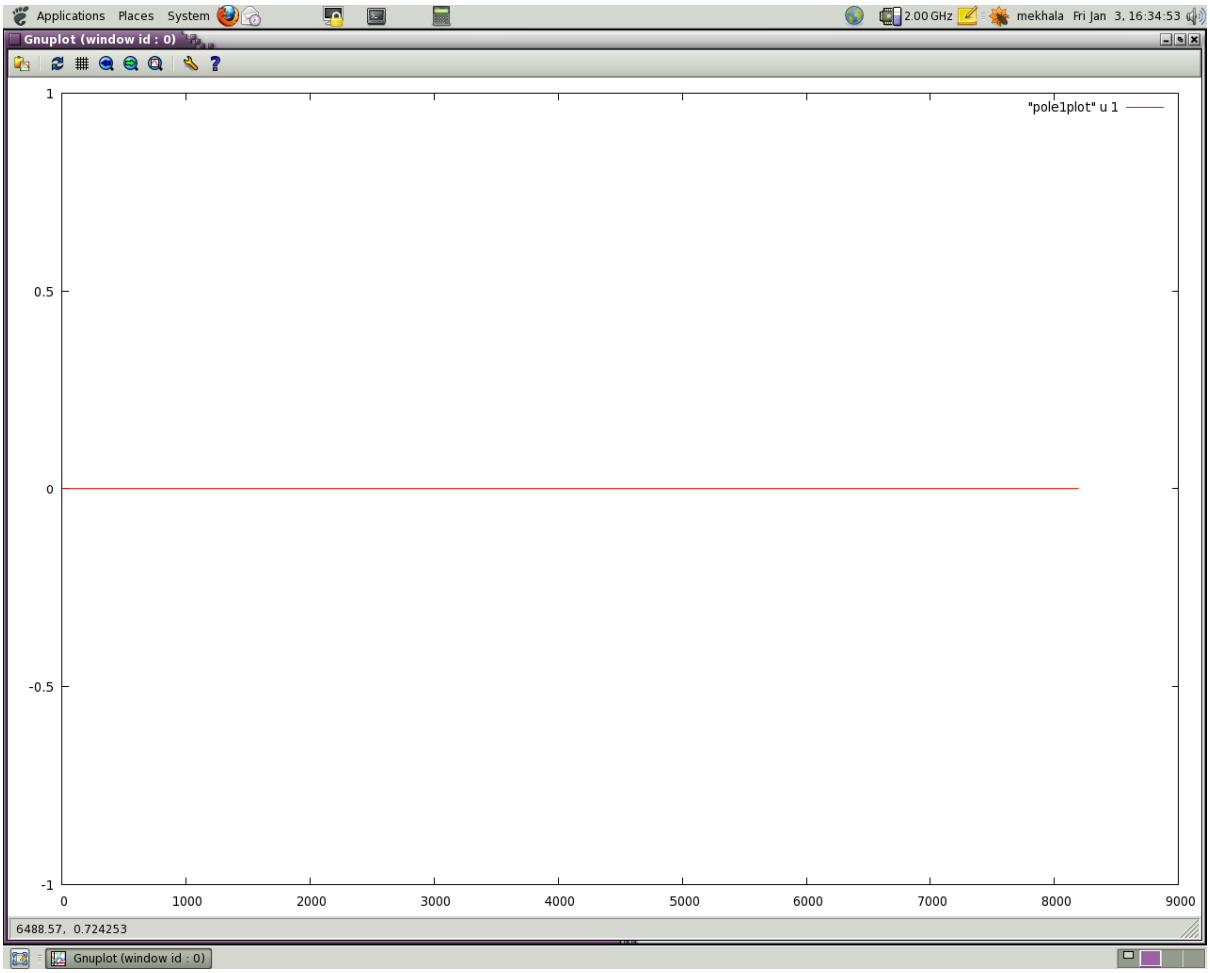
Before integration:



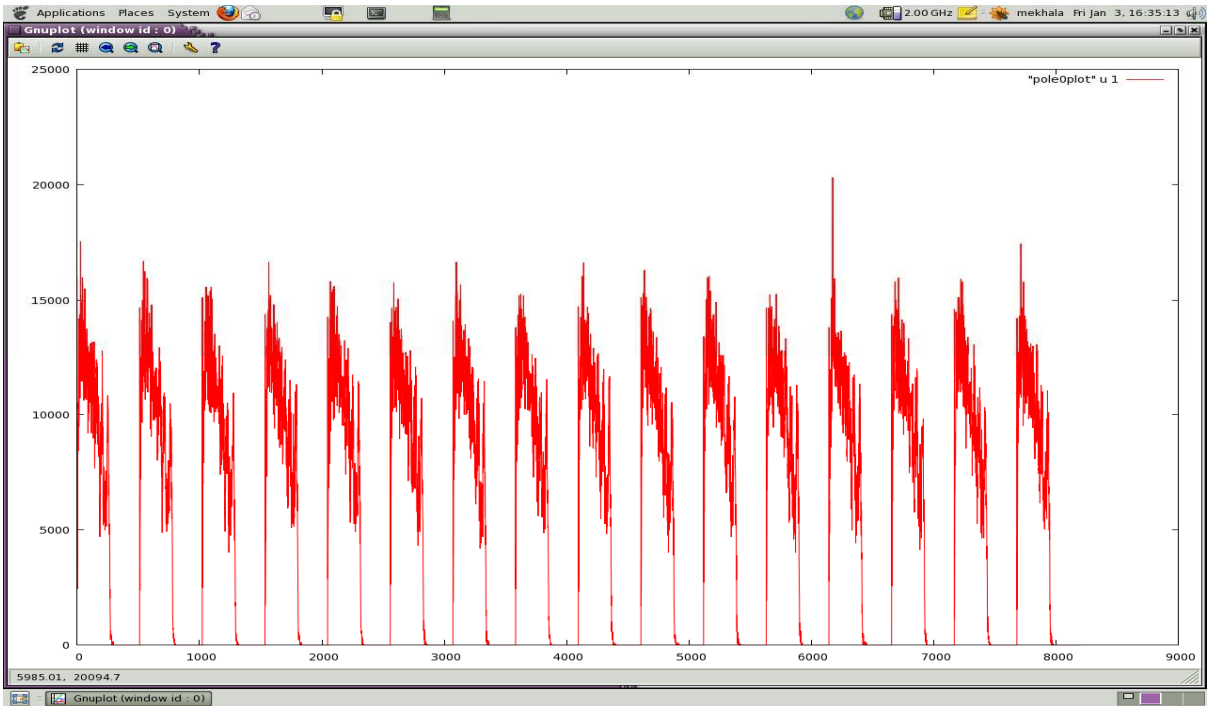
After Integration:



Pole1:


Applications Places System
2.00 GHz mekhala Fri Jan 3, 16:34:53
Gnuplot (window id : 0)
"pole1plot" u 1
1
0.5
0
-0.5
-1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
6488.57, 0.724253
Gnuplot (window id : 0)

Pole0:


Applications Places System
2.00 GHz mekhala Fri Jan 3, 16:35:13
Gnuplot (window id : 0)
"pole0plot" u 1
25000
20000
15000
10000
5000
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
5985.01, 20094.7
Gnuplot (window id : 0)

Addition of Pole1 and Pole0: