Polyphase Filter Bank implementation in GWB

Version 2, 03/08/2021

S. Harshavardhan Reddy

1. Polyphase Filter Bank

Discrete Fourier Transform (DFT) suffers from two drawbacks i.e; leakage and scalloping loss. Applying Polyphase Filter Bank (PFB) supresses the drawbacks of DFT. In signal processing terms, PFB is a linear filter applied on frequency channels representing the DFT frequency bins after DFT. Another important use of PFB is that it serves as a band pass filter where desired frequency range is kept and the rest is made to zero.

2. Implementation

Instead of taking a N-point FFT directly, a block of data of size N x P = M (where P represents number of taps) is read and multiplied point-by-point with a window function. The window function is basically a sinc function which is tranform of rectangular function as the aim is to have single bin frequency response to be rectangular. This window function can further be multiplied by any smoothing window function like hamming or hanning window for better response. After multiplication, the block of data is split into P subsets of length N each, and added point-by-point. Then an N-point FFT is performed on this array of data. Pictorial representation is given below.

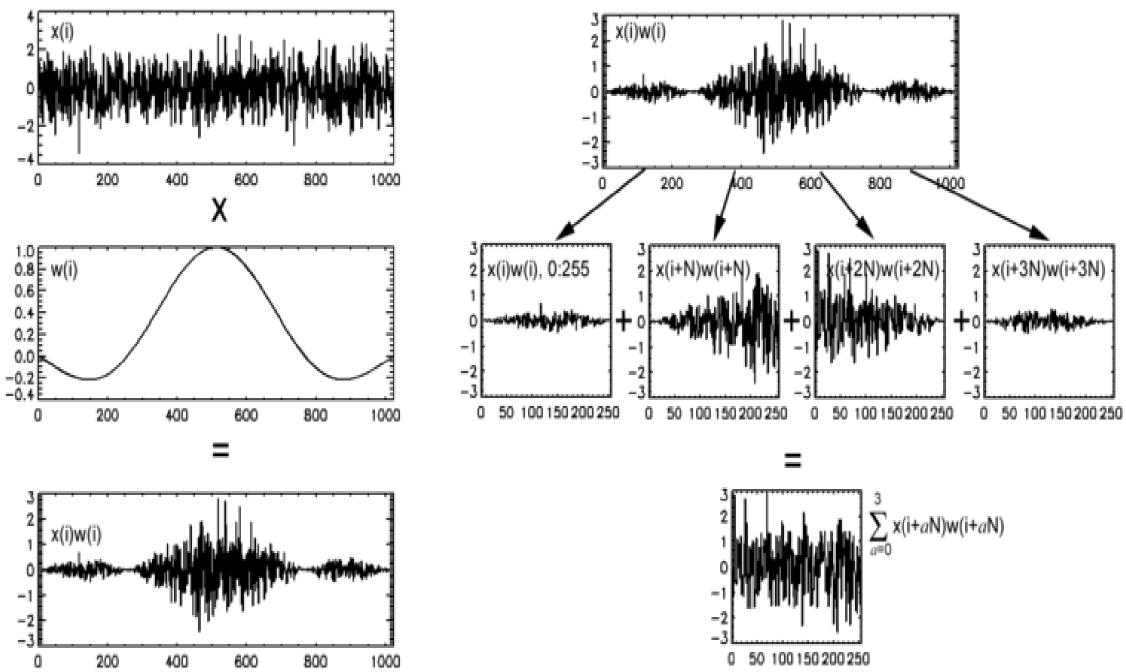

Fig. 1 : Graphical depiction of polyphase filtering. x(i) is a time series of length M = 1024 samples, multiplied point-by-point with the window function w(i) (a sinc function), also of the same length. The product is split into P = 4 blocks of length N = 256 samples each, and summed. This summed array of length N = 256 samples, shown at the bottom, on the right, is then input to a routine that takes a 256-point Fourier Transform. (Pic courtesy : "The Polyphase Filter Bank Technique" by Jayanth Chennamangalam)

3. Implementation in the GWB

The GWB is implemented using time-slicing design i.e; the data stream from all antennas is sliced in time and each compute node with GPUs processes a slice of data stream from all antennas. The steps of processing in GPUs are converting 8-bit or 4-bit data into floating point followed by FFT using CUFFT library and multiplication and accumulation (MAC).

In the GWB, PFB is implemented after the stage where data is converted from 8-bit or 4-bit to floating point. For a slice of data, the last N (FFT size N) samples requires (P – 1) x N samples of next slice for performing the PFB operation. Hence, at the stage of time slicing and data sharing, extra data required for last N samples' PFB operation is also shared.

In the narrowband mode, PFB is applied on the decimated data. In the GWB, narrowband mode is implemented using digital downconversion (DDC) technique. In this method, In-phase and Quadrature timeseries are low pass filtered and decimated by a factor more than desired amount of decimation before passing them on to a Complex FFT stage as real and imaginary parts respectively. For example, from 100 MHz Baseband signal if we want to extract 25 MHz narrowband, the In-phase and Quadrature timeseries are decimated by a factor of 8 instead of decimation by a factor of 4. In this mode, PFB is implemented on both In-phase and Quadrature timeseries after the low pass filter and decimation stage (*Fig. 2*). Window function length here is P x N samples where P is No. of taps and N is FFT size. It has to be noted that in the Narrowband mode as the input to FFT stage is a Complex time series, number of spectral channels is equal to the FFT size.

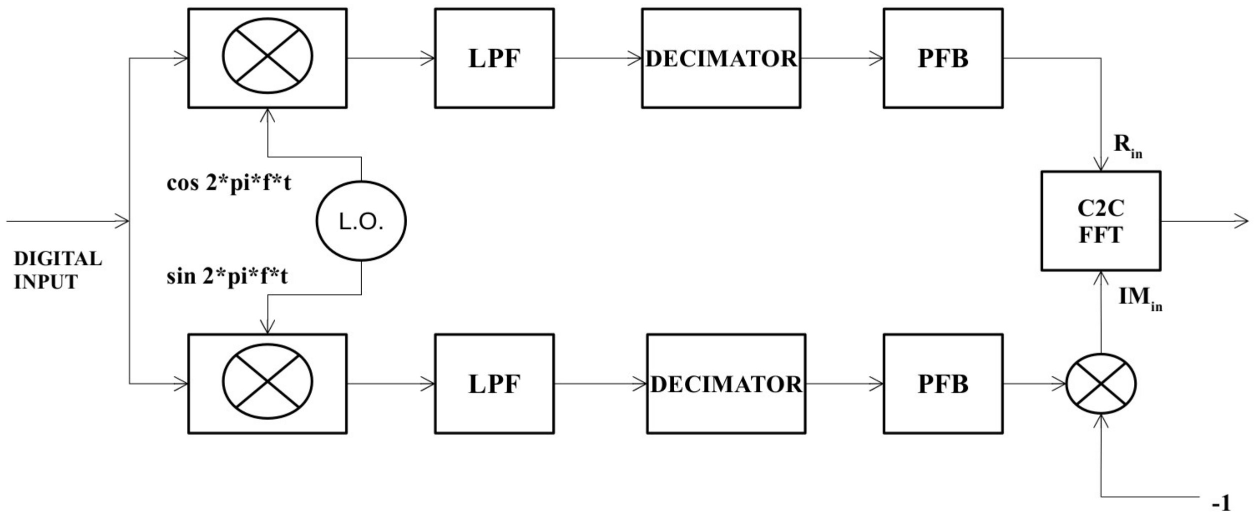

Fig. 2 : Block diagram showing implementation of PFB in Narrowband mode in the GWB

Coming to the actual implementation in the GPU kernel, the window function is pre-calculated depending on the number of spectral channels and number of taps and stored in the global memory of GPU. Here the window function is sinc function multiplied with hanning window function for smoothing. In the kernel, each thread perform PFB (filtering) operation for a single sample. The number of threads in a block is kept fixed at 256 (Block Size). The block dimensions are then calculated such that, in the x-dimension the number of blocks = (Total number of samples per time slice per antenna / Block Size) and y-dimension the number of blocks = number of antennas * number of pols.

4. Computation requirements and performance :

On each time sample, the computation performed is equal to number of taps multiplication plus number of taps additions. If each multiplication takes one floating point operation and addition takes one floating point operation then for each sample the computation performed is (number of taps x 2) floating point operations. The data rate per polarisation per second is 400 MSamples/second. So, total computation required is (Bandwidth x 2 x number of taps x number of antennas x number of pols x 2). For bandwidth of 200 MHz and 8 taps, the total computation required for PFB is 409.6 GFlops. The performance achieved in K40 GPUs was around 75 GFlops. The relatively low performance compared to the peak performance of K40 GPU is because of non-coalesced memory access of global memory.

5. Test results

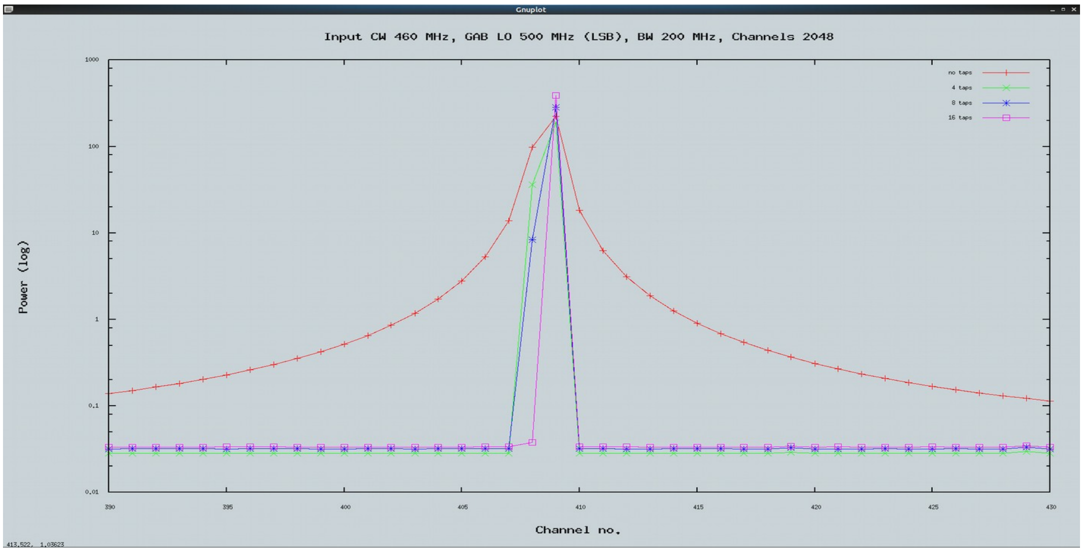

Fig. 3 : Plot showing effect of PFB at various tap lengths. BW : 200 MHz, CW Signal : 460 MHz, LO : 500 MHz, No. of channels : 2048

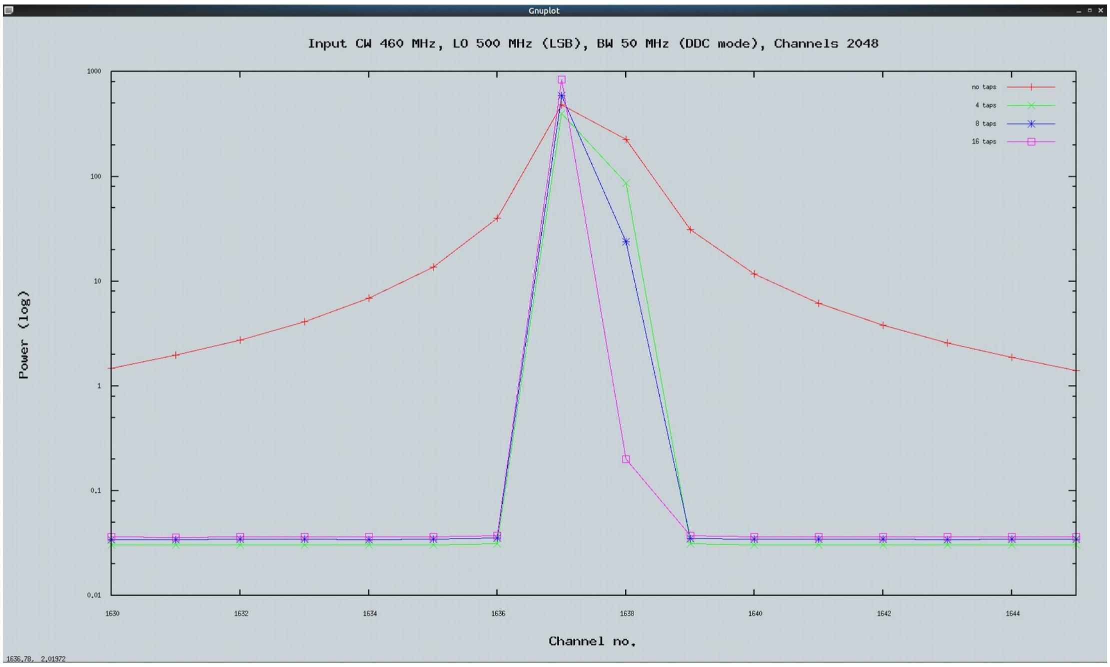

Fig. 4 : Plot showing effect of PFB at various tap lengths. BW : 50 MHz, CW Signal : 460 MHz, LO : 500 MHz, No. of channels : 2048

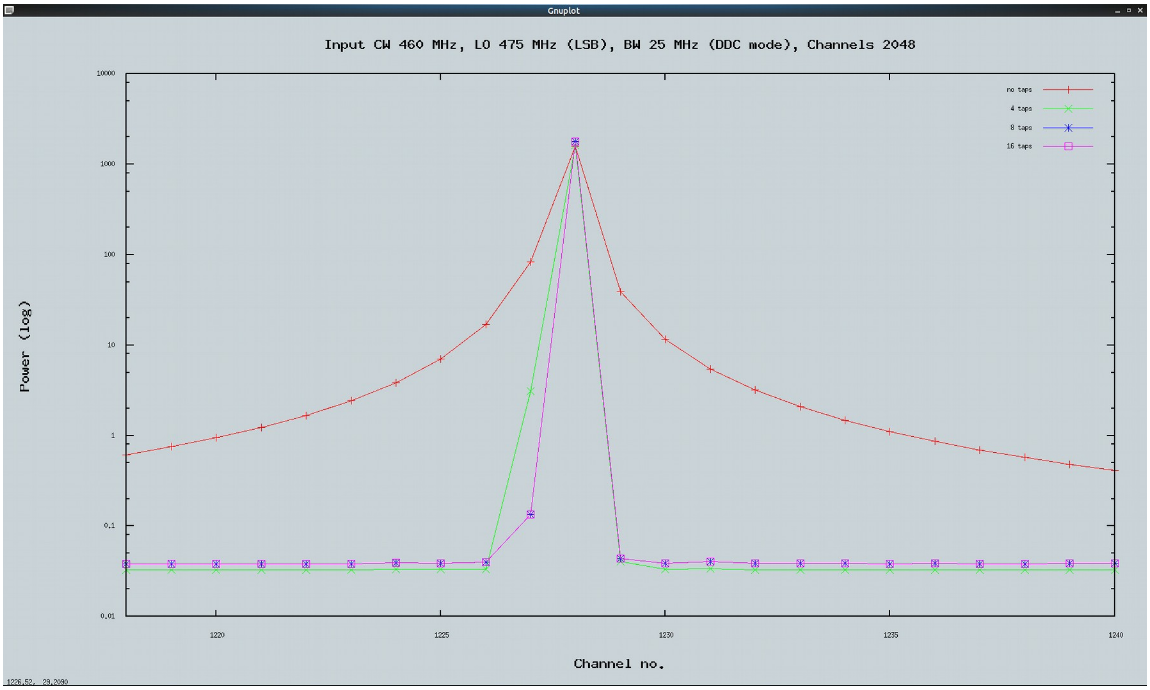

Fig. 5 : Plot showing effect of PFB at various tap lengths. BW : 25 MHz, CW Signal : 460 MHz, LO : 475 MHz, No. of channels : 2048

6. Possible modes in GWB with PFB :

As PFB comes at the cost of extra computation some modes of GWB where the computation time is near to real-time will not work as the total computation time including PFB will overflow the real-time. With the computation performance achieved on K40 GPUs, possible modes are listed below :

Bandwidth	**Interferometry**	**Beamformer**
200 MHz/ 100 MHz	Maximum taps = 16	All modes are possible including PA full polar mode*
400 MHz	Maximum taps = 4	**Above 8192 channels no beams are possible with PFB mode ON**
		Up to 8192 channels all modes are possible incluing PA full polar mode*
Narrowband mode	Decimation <= 4, Maximum taps = 16	All modes are possible including PA full polar mode*
	Decimation = 8, Maximum taps = 8	
	Decimation = 16, Full Stokes mode, Maximum taps = 4 Total Intensity, Maximum taps = 8	
	Decimation > 16, GWB does not support PFB mode in this configuration	**Decimation > 16, GWB does not support PFB mode in this configuration**

7. Timestamp offset in PFB mode :

In the PFB mode (for P taps), timeseries for an FFT block of FFT size N requires data from the next P-1 FFT blocks to perform Polyphase filtering as mentioned in the "Implementation" section. In the GWB, for any block timestamp assigned is the time at the beginning of the block. But in the Polyphase filtering process, (P x N) samples' timeseries is multiplied by a window function of same length which has significant weights around the centre of the (P x N) samples' block and hence any signal amplitude variations from around (P x N / 2) samples appear in the current FFT block. Hence, it has been decided to adjust the timestamp by (P x N x Sampling Period / 2) samples for the PFB mode. In the narrowband mode as the PFB is implemented on In-phase and Quadrature timeseries that are decimated by twice the decimation factor desired and as the FFT stage input is Complex timeseries, the timestamp is adjusted by (P x N x Sampling Period). This has been experimentally verified by using a PPS modulated CW signal as input, GWB running with PFB mode ON and grabbing beam data at minimum possible integration and plotting the timeseries of the channel corresponding to the CW signal frequency within the band.

Hence, to correct the timestamp the following equation has to be applied

timestamp actual = timestamp recorded + Δt^{pfb} where Δt^{pfb} = No. of taps in PFB * No. of Spectral Channels * Sampling Period[1]

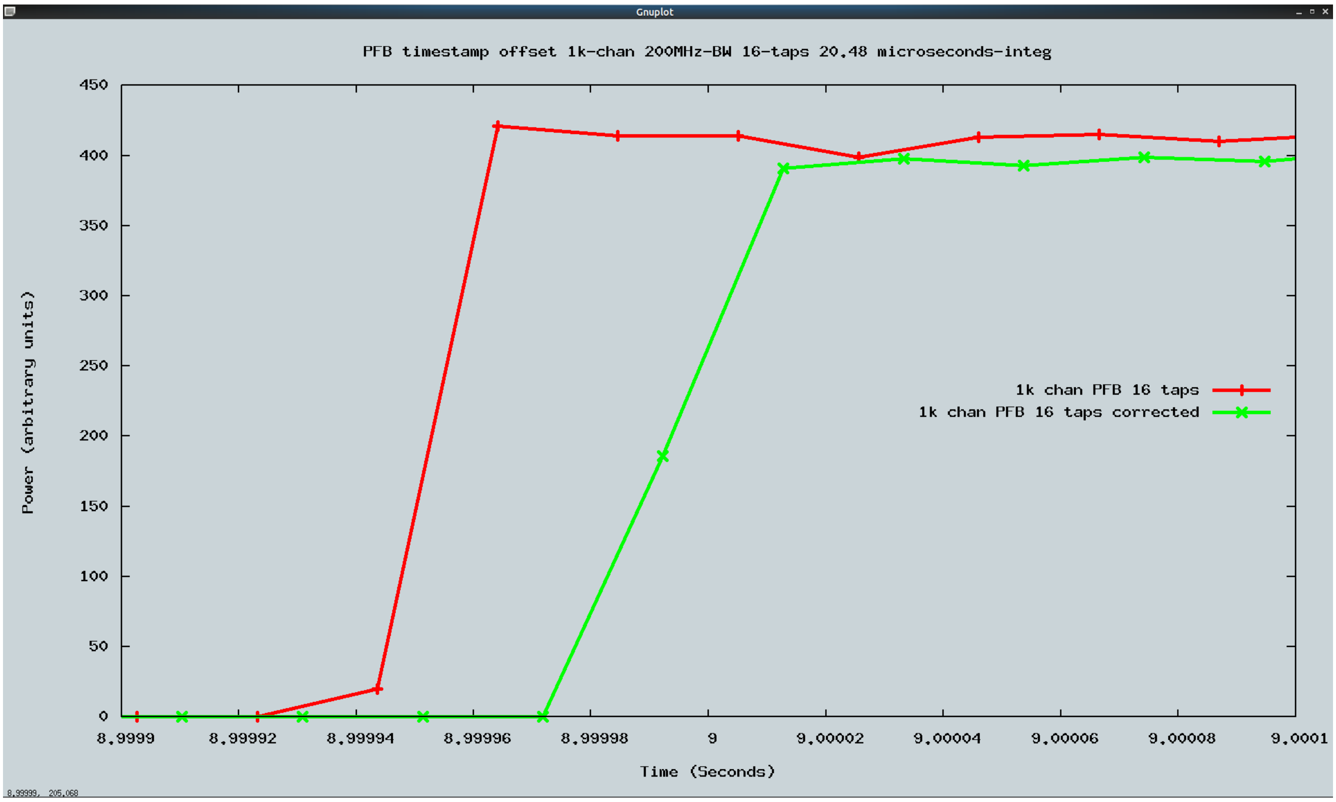

Fig. 6 : Plot showing timestamp offset in PFB mode. BW : 200 MHz, Channels : 1024, GAB LO : 550 MHz (550 – 750 MHz band), CW signal : 686.71875 MHz (700th channel), PA beam, Sampling Period : 20.48 microseconds, PFB taps : 16. Red plot : Before timestamp offset correction Green Plot : After timestamp offset correction.

1 Sampling Period in Narrowband mode is to be considered with respect to final bandwidth. For 25 MHz Bandwidth Narrowband mode, Sampling Period is 20 nanoseconds.

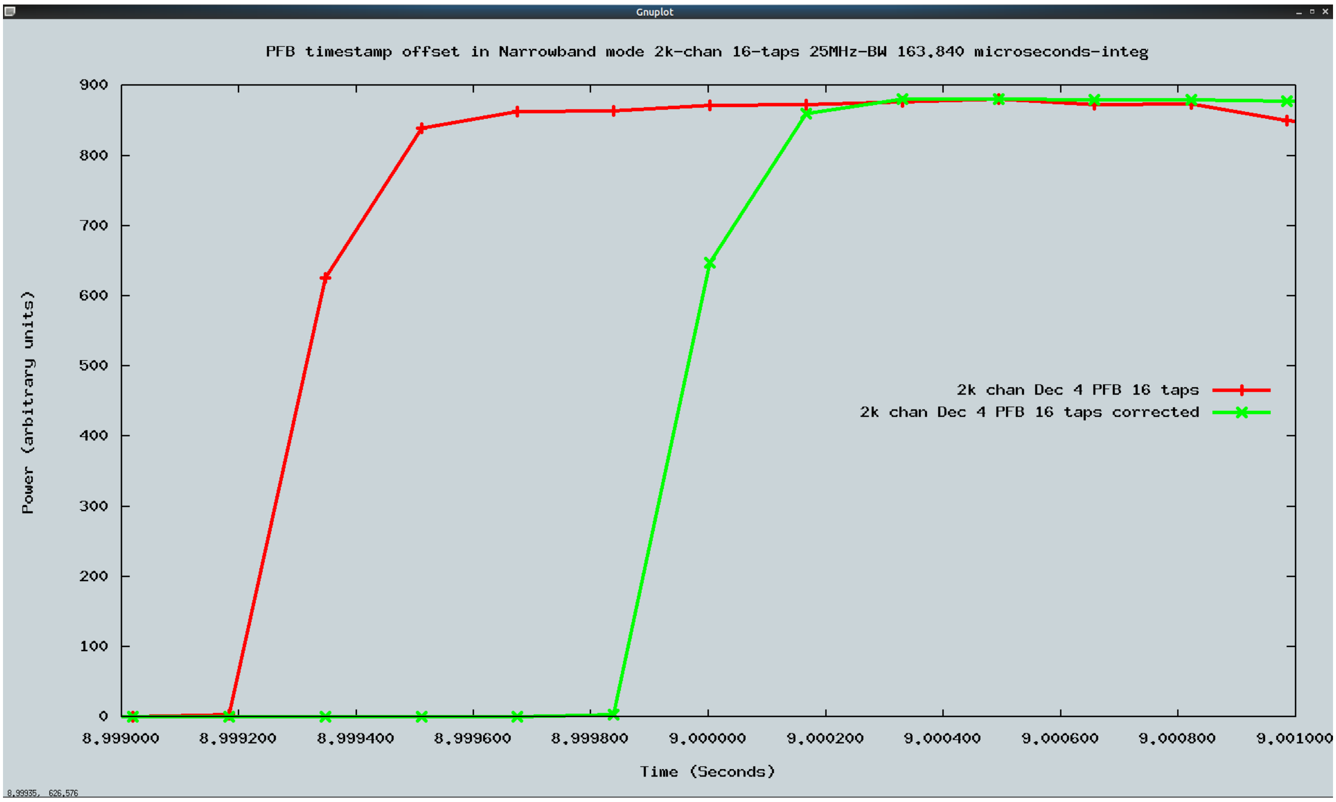

Fig. 7 : Plot showing timestamp offset in PFB mode in Narrowband mode. BW : 25 MHz, Channels : 2048, GAB LO : 550 MHz DDC LO : 10 MHz (560 – 585 MHz band), CW signal : 569.765625MHz (800th channel), PA beam, Sampling Period : 163.840 microseconds, PFB taps : 16. Red plot : Before timestamp offset correction Green Plot : After timestamp offset correction.

References :

1. Chennamangalam J., [2011] The Polyphase Filter Bank Technique, http://casper.berkeley.edu/wiki/The Polyphase Filter Bank Technique.

2. Reddy, S. H. et al. 2017, "A Wideband Digital Back-End for the Upgraded GMRT", JAI, 6, 164101