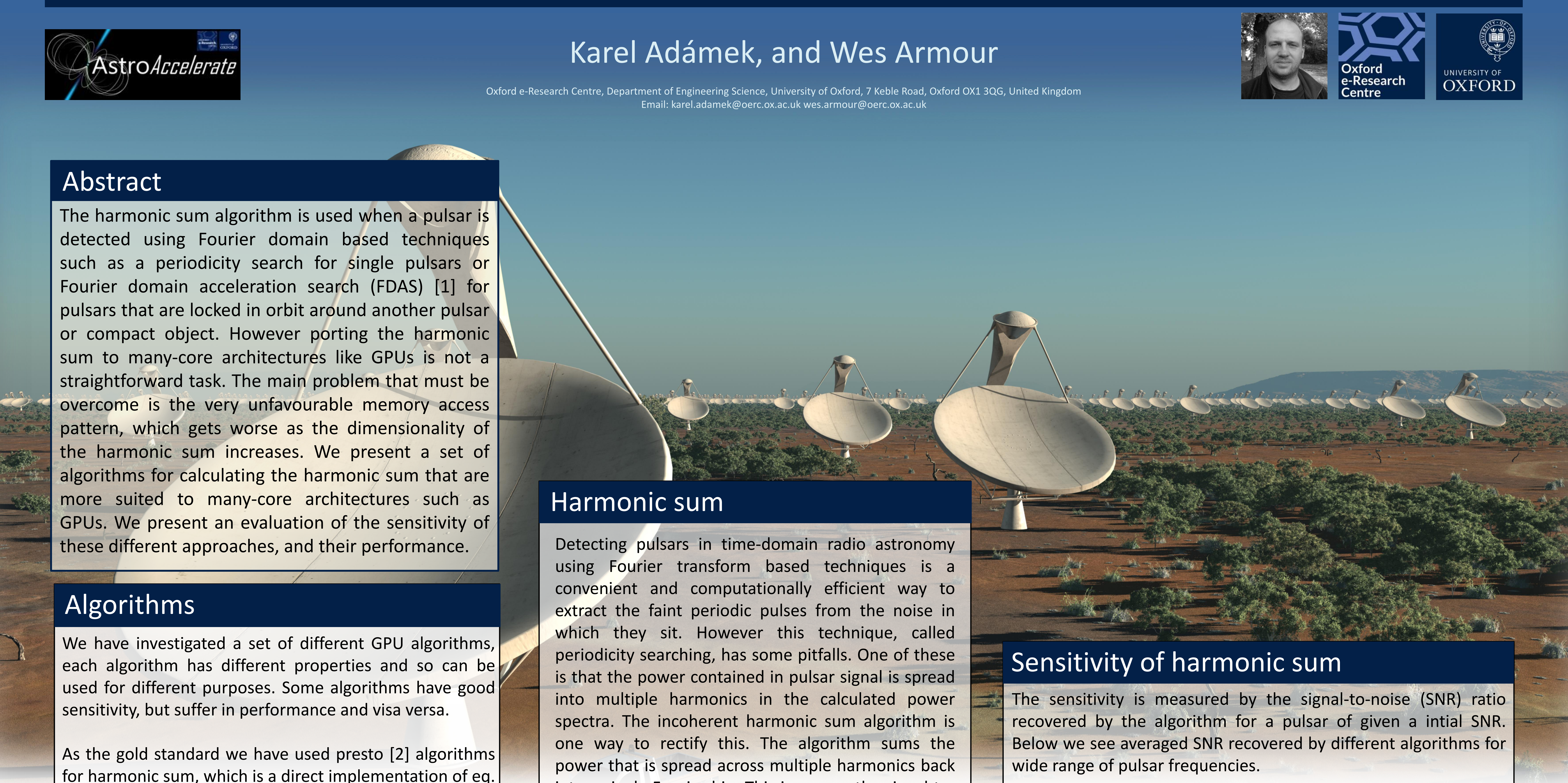
A GPU implementation of the harmonic sum algorithm



Karel Adámek, and Wes Armour

Oxford e-Research Centre, Department of Engineering Science, University of Oxford, 7 Keble Road, Oxford OX1 3QG, United Kingdom
Email: karel.adamek@oerc.ox.ac.uk wes.armour@oerc.ox.ac.uk

Abstract

The harmonic sum algorithm is used when a pulsar is detected using Fourier domain based techniques such as a periodicity search for single pulsars or Fourier domain acceleration search (FDAS) [1] for pulsars that are locked in orbit around another pulsar or compact object. However porting the harmonic sum to many-core architectures like GPUs is not a straightforward task. The main problem that must be overcome is the very unfavourable memory access pattern, which gets worse as the dimensionality of the harmonic sum increases. We present a set of algorithms for calculating the harmonic sum that are more suited to many-core architectures such as GPUs. We present an evaluation of the sensitivity of these different approaches, and their performance.

Algorithms

We have investigated a set of different GPU algorithms, each algorithm has different properties and so can be used for different purposes. Some algorithms have good sensitivity, but suffer in performance and visa versa.

As the gold standard we have used presto [2] algorithms for harmonic sum, which is a direct implementation of eq. (1)

Our Max HRMS algorithm works by looking for the maximum at possible locations of the harmonic and adds this to the partial sum of the harmonic sum.

$$h(n)_H = \sum_{i=1}^{H} \max_{1\le j\le i}\big(x(in+j)\big)$$

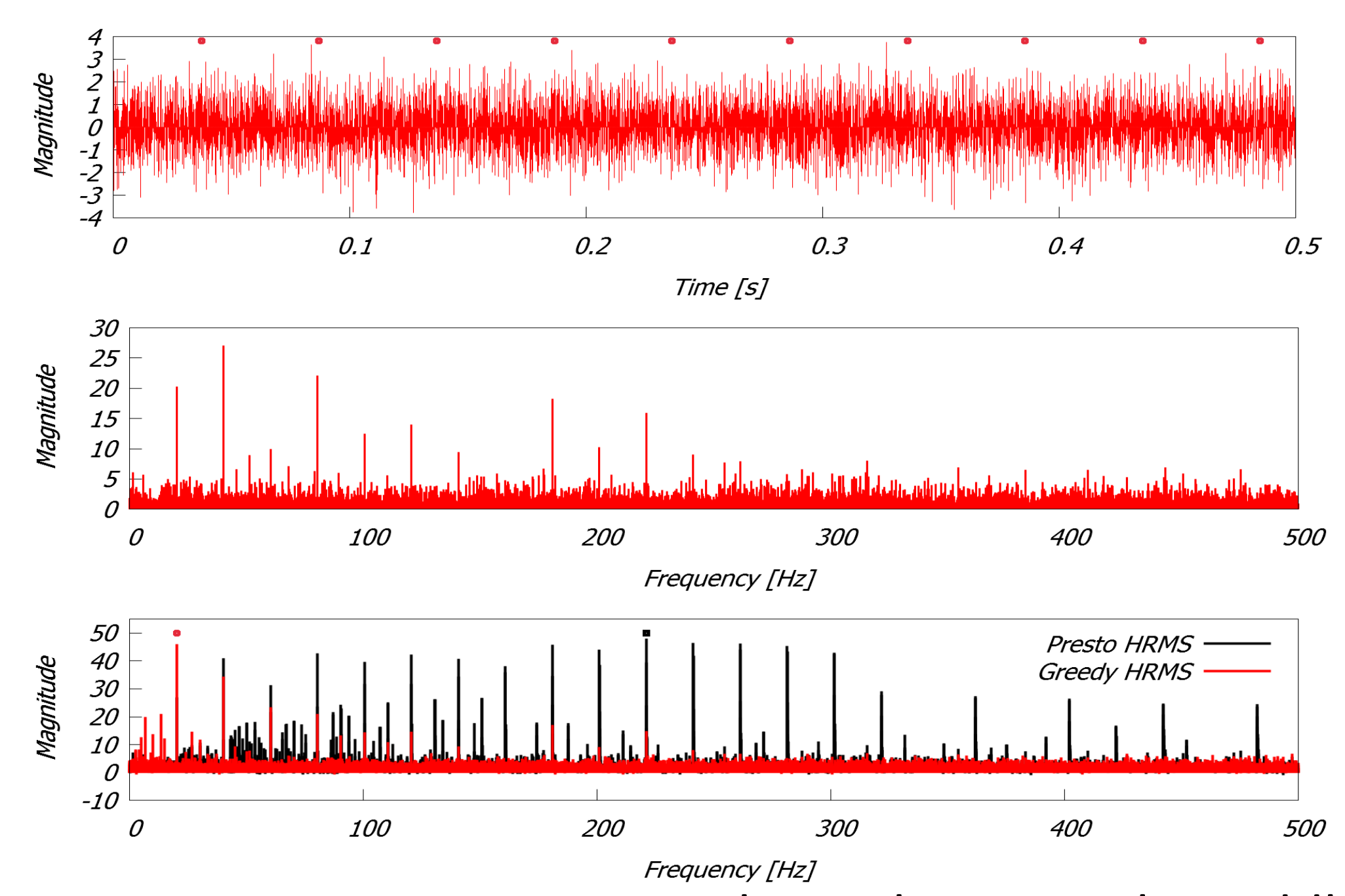
Our freq. bin HRMS only adds values for bins which are integer multiples of fundamental frequency that is

$$h(n)_H = \sum_{i=1}^{H} x(in)$$

Lastly in Greedy HRMS we recursively add to the partial sum value in an appropriate bin or its neighbor depending what is bigger.

$$h(n)_{H+1} = h(n)_H + \max\big(x(Hn+j), x(Hn+j+1)\big)$$

where *j* is increased by one if *x(Hn+j+1)* is selected.



Top: Time-series containing the pulsar signal. Middle: The signal (Top) Fourier transformed, the series of harmonics are visible. Bottom: The results from two of our algorithms for the harmonic sum.

Harmonic sum

Detecting pulsars in time-domain radio astronomy using Fourier transform based techniques is a convenient and computationally efficient way to extract the faint periodic pulses from the noise in which they sit. However this technique, called periodicity searching, has some pitfalls. One of these is that the power contained in pulsar signal is spread into multiple harmonics in the calculated power spectra. The incoherent harmonic sum algorithm is one way to rectify this. The algorithm sums the power that is spread across multiple harmonics back into a single Fourier bin. This increases the signal-to-noise ratio of detected pulsars and allows us to detect weaker pulsars as a result. Harmonic summing also forms part of the search techniques used for detecting accelerated pulsars where a two-dimensional harmonic sum is required. The harmonic sum is given by equation

$$h(n)_H = \frac{1}{\sqrt{H}} \sum_{i=1}^{H} x\left(\frac{ni}{H}\right) \qquad (1)$$

Where *H* is the number of harmonics summed. The number of harmonics we need to sum is governed by the duty-cycle of the pulsar we are looking for.

To simulate a pulsar's signal we are using the von Mises distribution with white noise.

Where it fits in

Simplified pulsar search pipeline
DEDISPERSE ↓ FFT ↓ HARMONIC SUM ↓ THRESHOLDING ↓ CANDIDATE LIST

In pulsar searches the observed time-series are first de-dispersed and then transformed using an FFT into frequency space. Then the harmonic sum is applied, which aims to sum the signal present and average out the noise. This is done by calculating partial sums of an increasing number of harmonics. The flow diagram of the pulsar search is on the left.

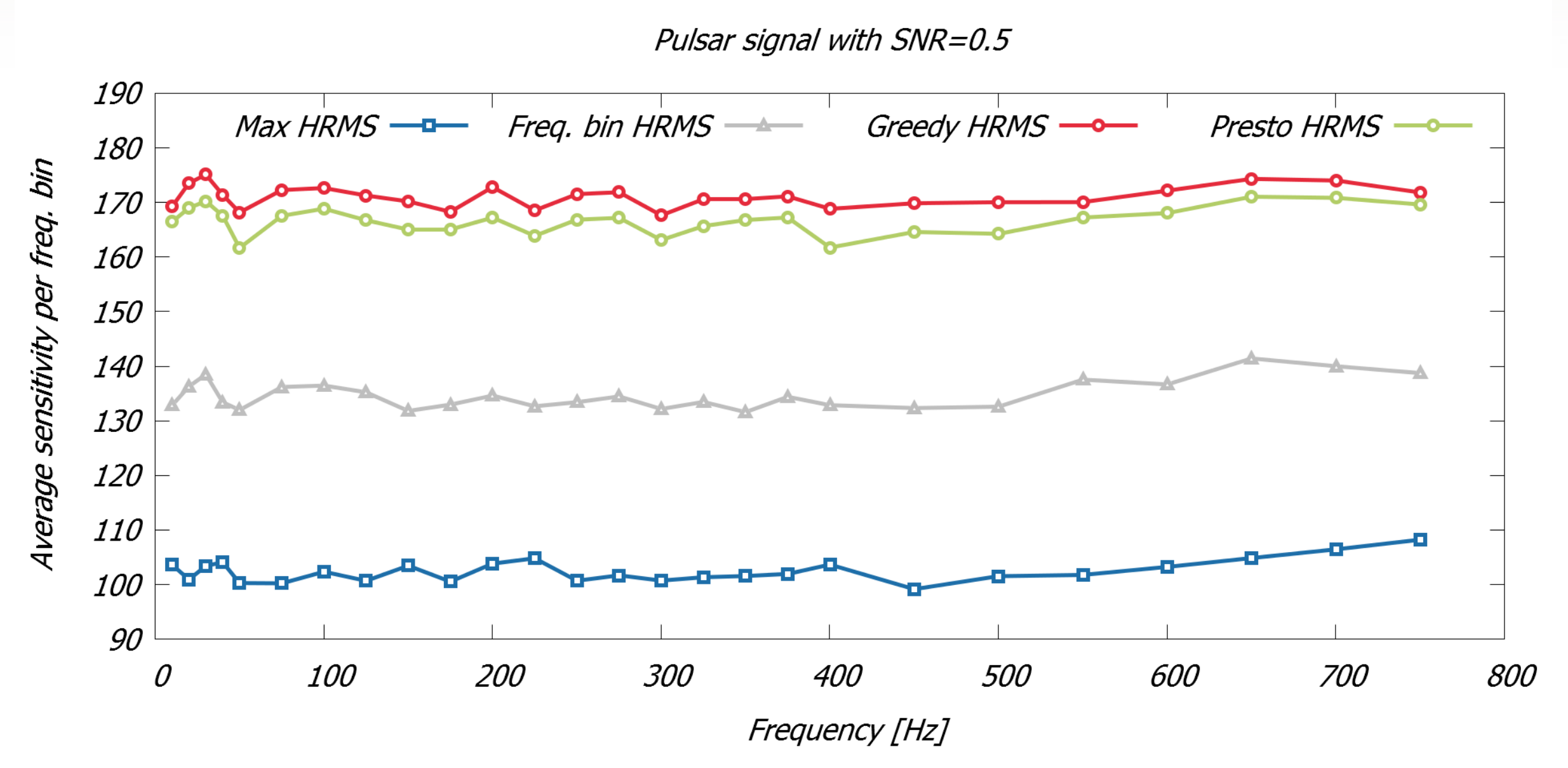
AstroAccelerate

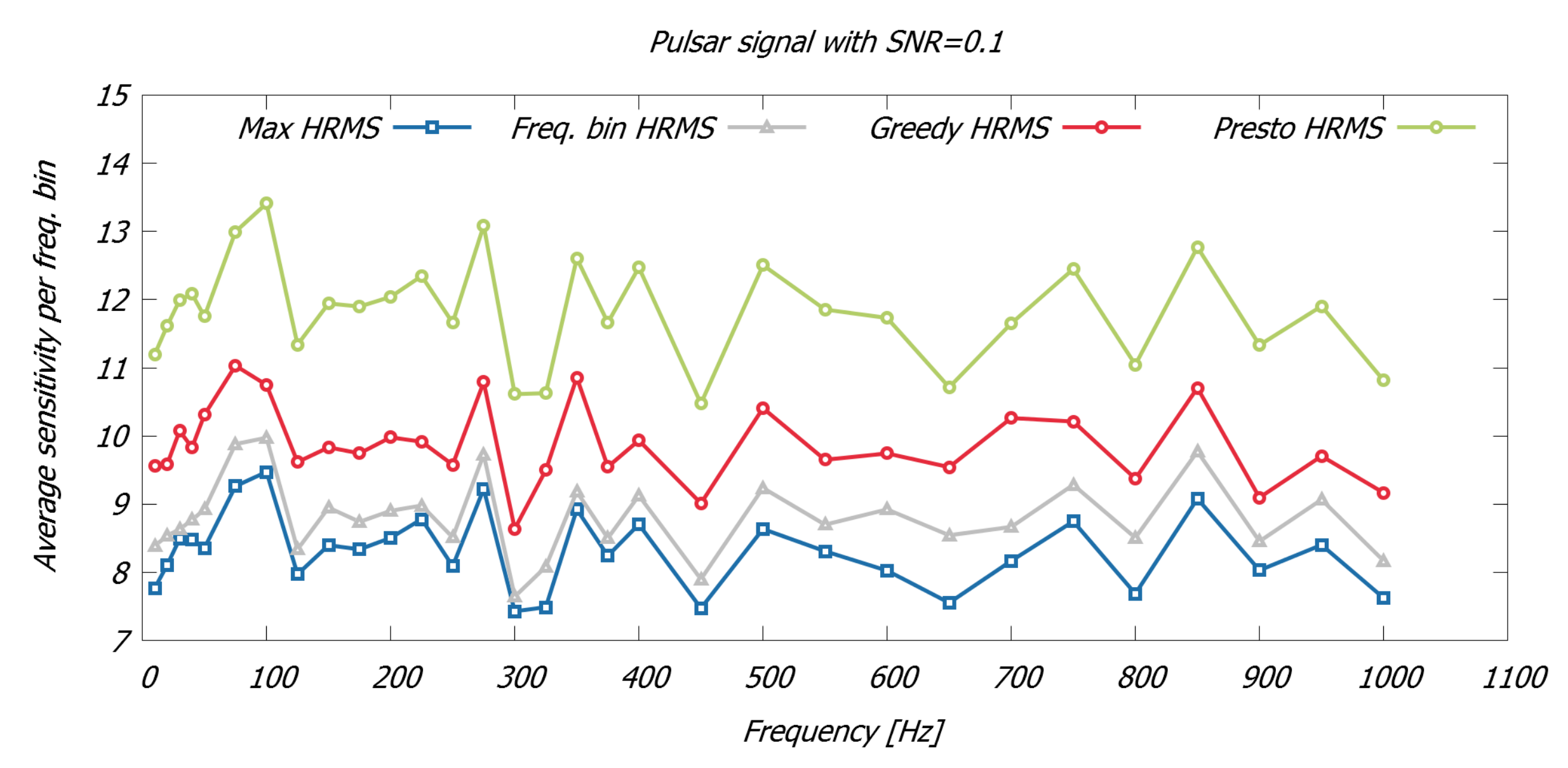
AstroAccelerate is a many core accelerated software package for processing time domain radio-astronomy data.

More information and source codes here: **https://github.com/AstroAccelerateOrg/astro-accelerate**

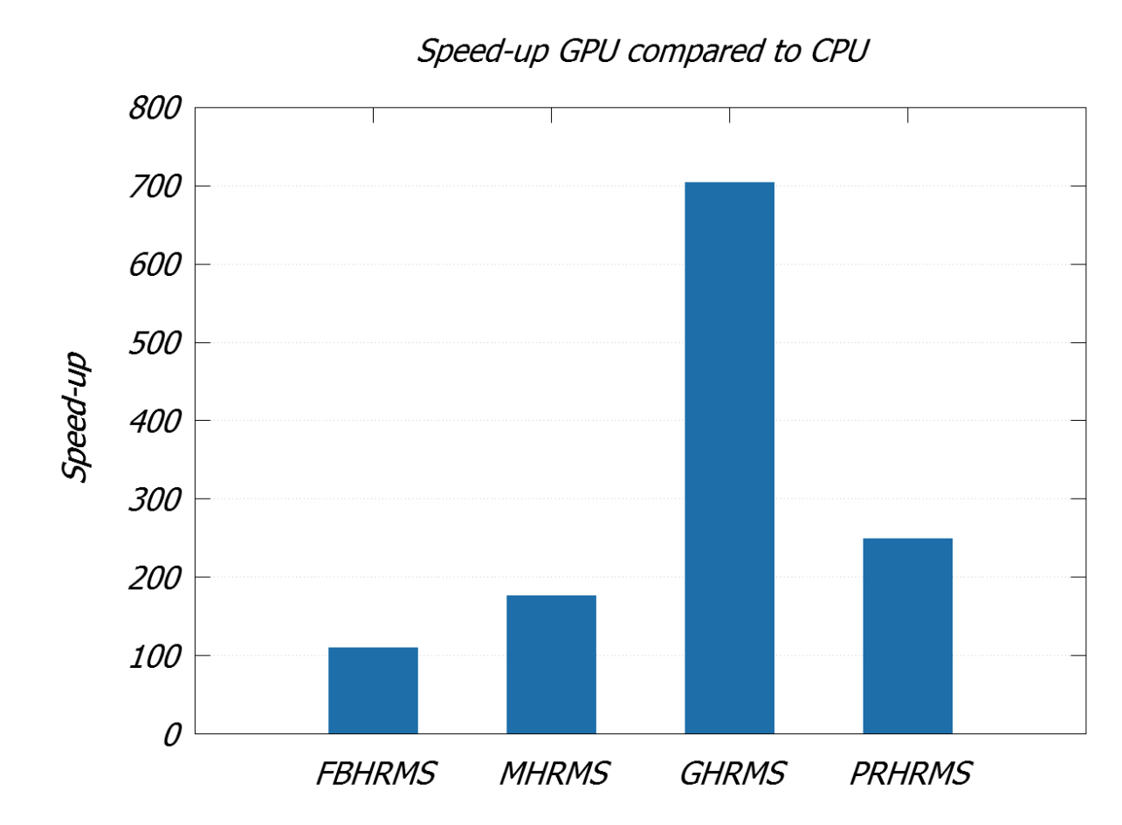
Sensitivity of harmonic sum

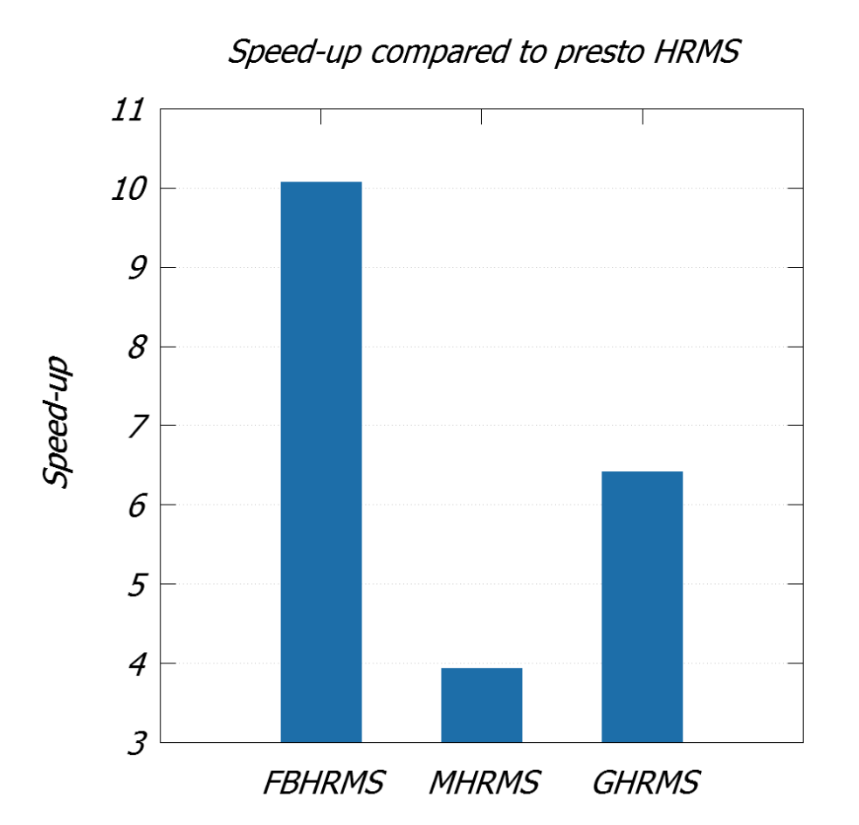
The sensitivity is measured by the signal-to-noise (SNR) ratio recovered by the algorithm for a pulsar of given a intial SNR. Below we see averaged SNR recovered by different algorithms for wide range of pulsar frequencies.



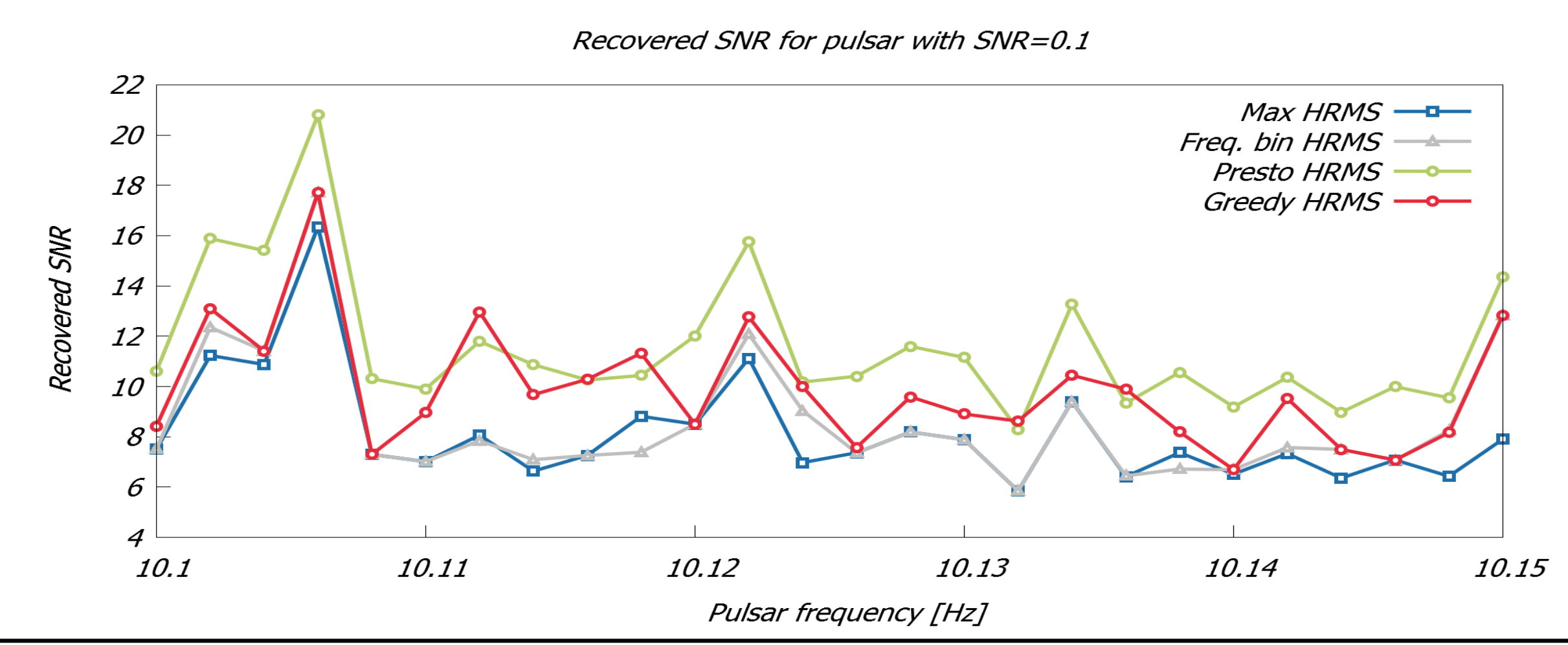


The speed-up of the GPU implementations is on the left and speed-up of examined algorithms vs Presto HRMS is to the right.





Recovered SNR for pulsar with SNR=0.1 within one frequency bin is shown below.



Conclusions

CPU vs GPU

The GPU ported harmonic sum algorithms outperform our parallel implementations of the harmonic sum on a CPU. Whilst our current harmonic sum codes suffer from poor memory access patterns, we still achieve (for some of our GPU algorithms) very good speedups.

References

[1] Ransom, S. M., Eikenberry, S. S., & Middleditch, J. 2002, The Astronomical Journal, 124, 1788
[2] Presto: https://github.com/scottransom/presto
Picture: SKA Website https://www.skatelescope.org
GPU used: NVIDIA TITAN V (CUDA 9.2)
CPU used: Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz (gcc version 6.4.1 20170727)