# Searching for pulsars in extreme orbits - GPU acceleration of the Fourier domain 'jerk' search

AstroAccelerate

Karel Adámek, Jan Novotný, Sofia Dimoudi, 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

Oxford e-Research Centre

UNIVERSITY OF OXFORD

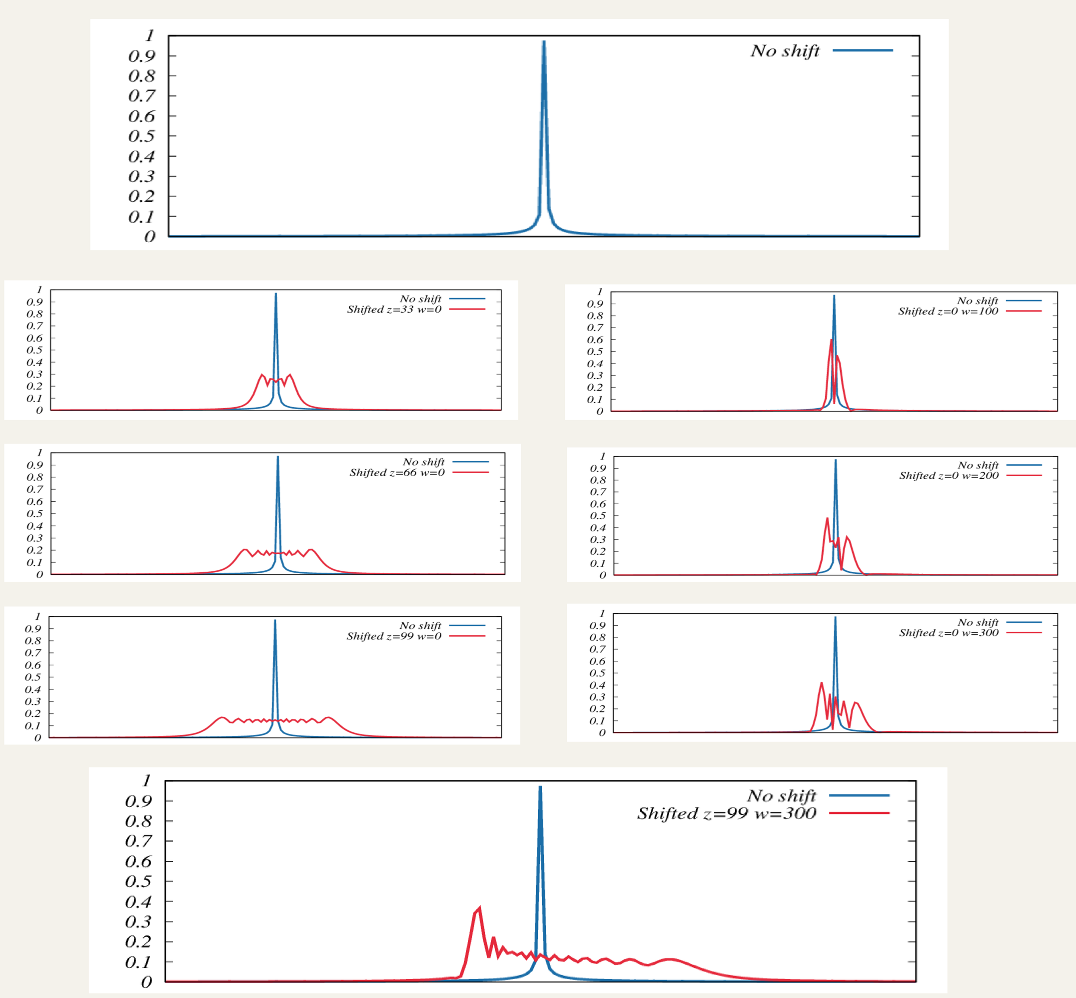
## Abstract

Binary pulsars represent examples of some of the most extreme physics in our Universe. These binary systems are natural laboratories for tests of a wide range of astrophysics, from tests of general relativity, including gravitational waves, to tests of equations of state of the matter from which pulsars are made of. However detection of these pulsars proves to be hard and computationally expensive

The orbital motion of a pulsar which is locked in a binary system causes a frequency shift (a Doppler shift) in the pulsar's normally very periodic pulse emissions. This shift results in a reduction in the sensitivity of traditional periodicity searchers which use the Fourier transform to pick up periodic components in the signal. When the Doppler shifted signal is Fourier transformed the power of the pulsar is smeared into surrounding frequency bins which makes subsequent detection much harder.

To correct this smearing, Ransom et.al. (2001, 2002) developed the Fourier domain acceleration search (FDAS) which reduces the smearing by applying a special set of matched filters.

However, this approach is limited to constant accelerations, therefore Anderson and Ransom 2018 expanded this method to account also for linearly changing acceleration, or a constant orbital jerk, of the pulsar.




The smearing in the case of the 'jerk' search depends on two quantities z and w. These represent the first and second derivative of the pulsars frequency.
Top: Signal from a stationary pulsar.
Middle: The red line shows the smearing for different values of the first derivation (middle left) and second derivative (middle right).
Bottom: The combination of first and second derivatives. is show at the bottom.

## AstroAccelerate

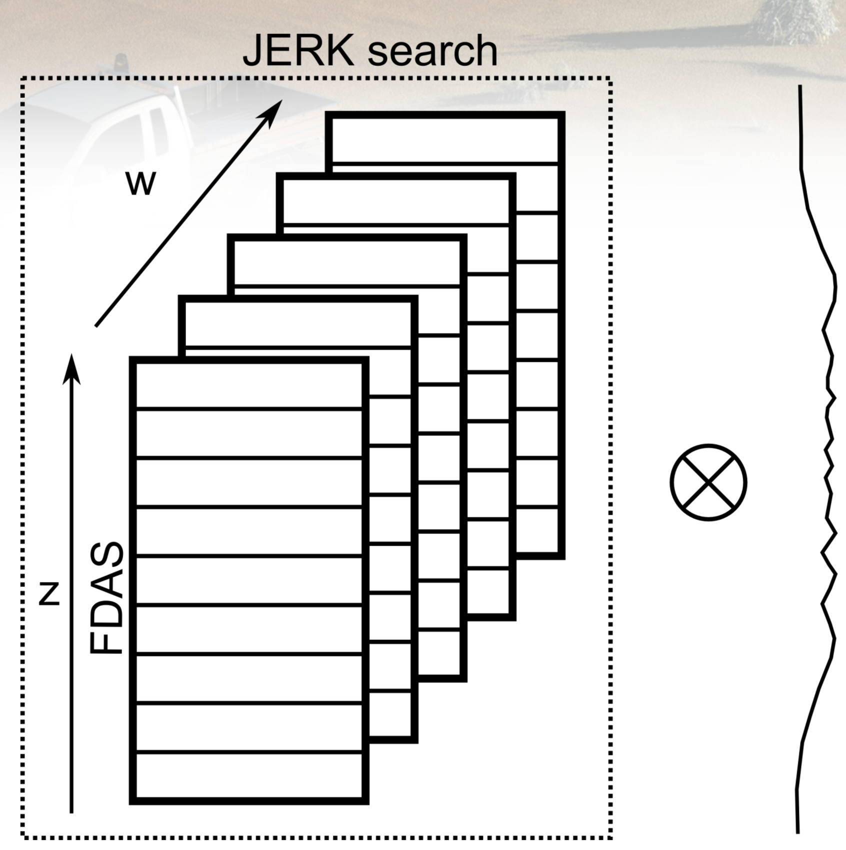
AstroAccelerate is a many core accelerated software package for processing time domain radio-astronomy data.
More information and source codes here:
**https://github.com/AstroAccelerateOrg/astro-accelerate**

## References

[1] Ransom S. M. Phd Thesis 2001
[2] Ransom, S. M., Eikenberry, S. S., & Middleditch, J. 2002, The Astronomical Journal, 124, 1788
[3] Anderson B. C. and Ransom S. M. 2018, The Astrophysical Journal Letters, 863, 1
[4] Presto: https://github.com/scottransom/presto
Picture: SKA Website https://www.skatelescope.org
GPU used: NVIDIA TITAN V (CUDA 9.2)

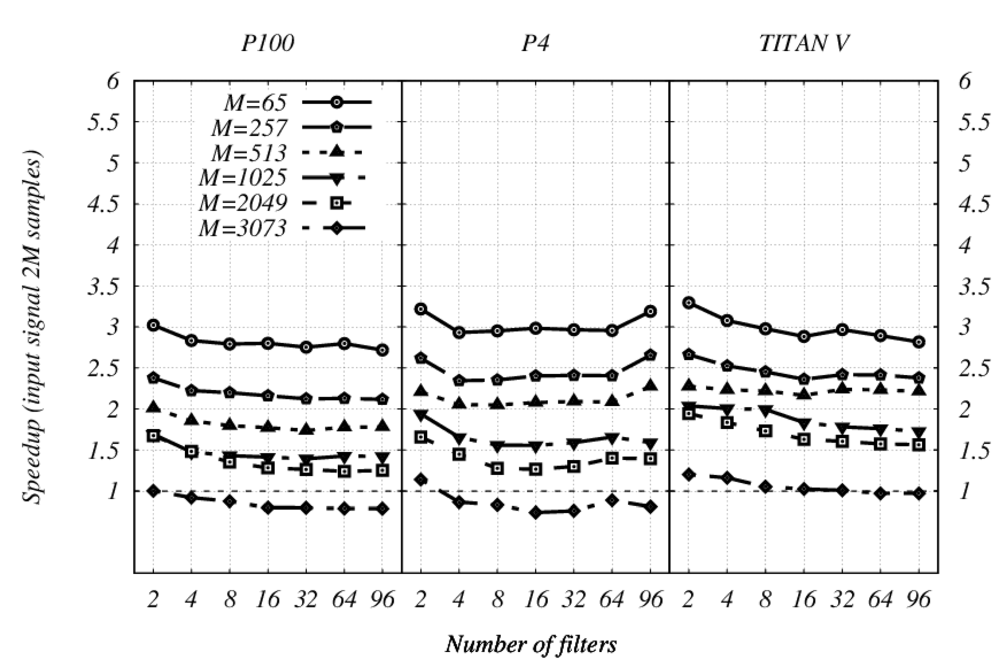
## Computation

The 'jerk' search uses matched filters to remove the smearing cause by the Fourier transform. Each filter fits a specific Fourier response for a given first frequency derivative $z = \dot{f}T^2$ and second frequency derivative $w = \dot{z} = \ddot{f}T^3$. The number of filters used in a 'jerk' search is greatly increased when compared to the standard Fourier domain acceleration search (FDAS).



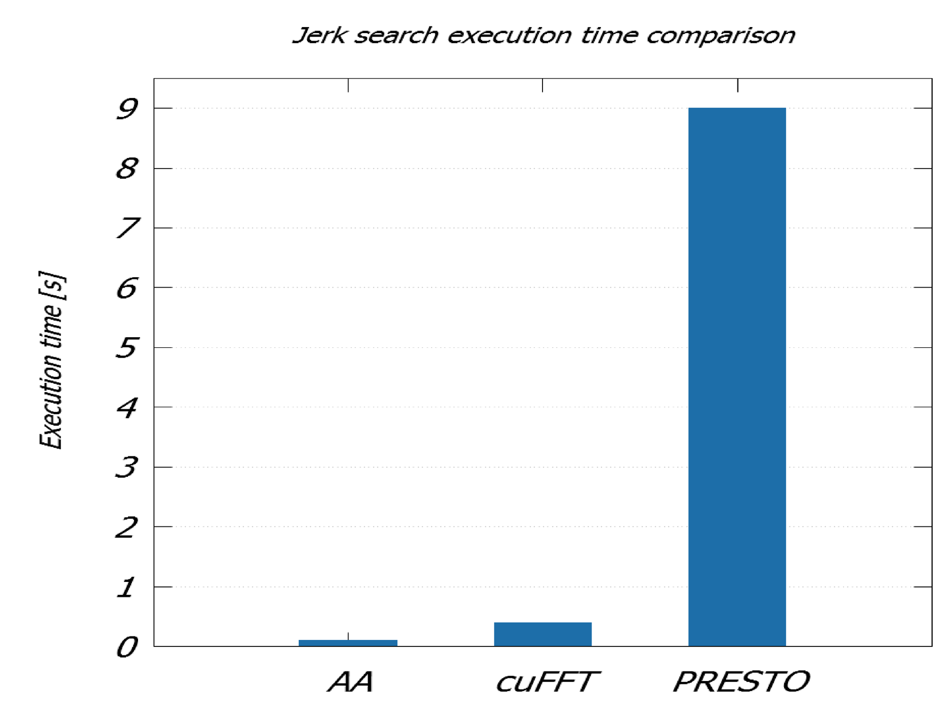

Jerk searches can be significantly larger than FDAS acceleration searches (40x-80x larger depending on sensitivity). This can mean processing as much as 400GB of data for every 10 minute observation.

To accelerate jerk searching we have used our shared memory implementation of the overlap-and-save method for the calculation of one-dimensional convolutions. This offers speedups of approximately 2.5x when compared to conventional implementations of one-dimensional convolutions on GPUs (filter sizes from 64 to 500 samples).

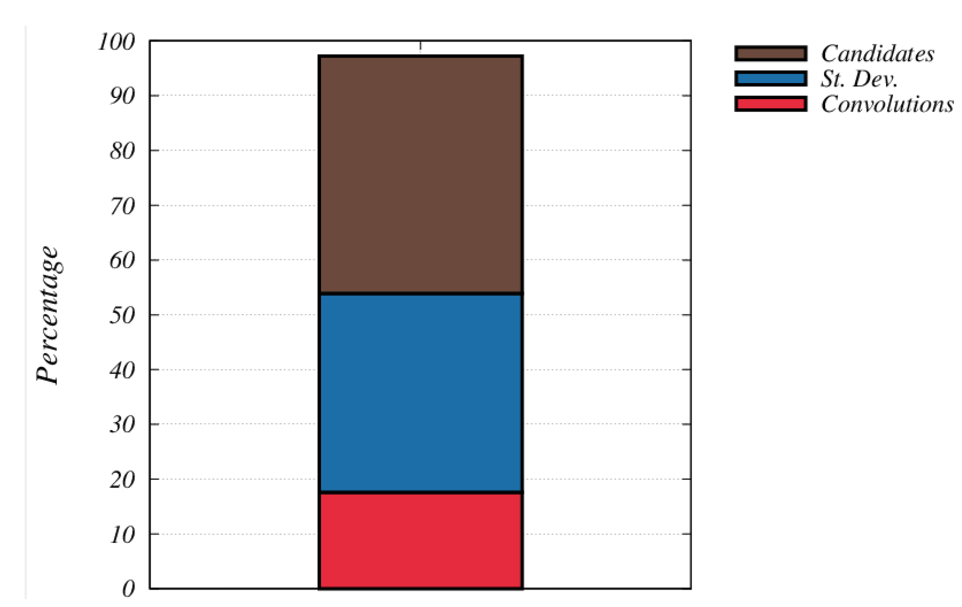



## Performance of the GPU 'jerk' search

We have compared the performance of out GPU implementation of the jerk search which uses our implementation of shared memory overlap-and-save method (AA) with cuFFT implementation of the overlap-and-save method with jerk search from PRESTO [4] which uses 4 threads. Our GPU implementation (TitanV GPU) is almost 90x faster then PRESTO which uses CPU (i7-4770K 3.5GHz).




The breakdown of the jerk search execution time on GPU shows that our implementation of the convolution is no longer the limiting factor. The execution time is dominated by candidate selection (43%) followed by calculation of mean and standard deviation (36%). The convolution account for only 21% of execution time. This is shown below.




## Future work

- **Harmonic sum** – improves sensitivity significantly but it is computationally expensive.
- **Improvement of the candidate selection** – improve performance of the mean and standard deviation calculation and peak finding algorithm
- **Classification using machine learning** – replacing current candidate selection with classification might improve sensitivity, but also the performance of the candidate selection.