Efficient FFT-based $\mathcal{F}$-statistic for continuous gravitational waves

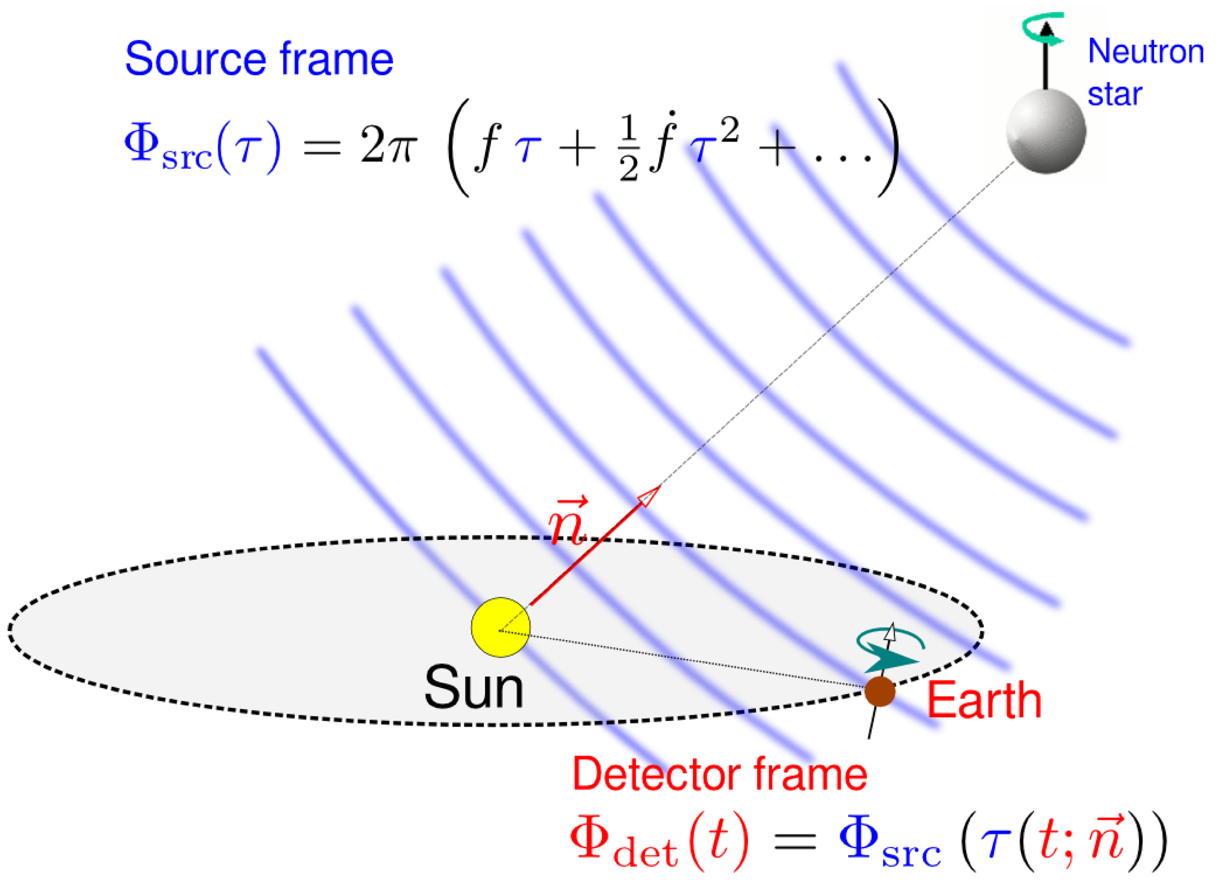
R. Prix, C. Messenger, M. Bensch, C. Choquet
Albert-Einstein-Institut Hannover

Introduction

We present a highly efficient implementation of the $\mathcal{F}$-statistic [1] for continuous-gravitational-wave searches. This method [4] leverages the power of the FFT for better scaling and speed improvements compared to the previous algorithm used by Einstein@Home [5], and also features a number of improvements compared to other existing implementations [2,3]. Porting this code to GPUs (using openCL) resulted in further efficiency gains by 1-2 orders of magnitude compared to the CPU version.

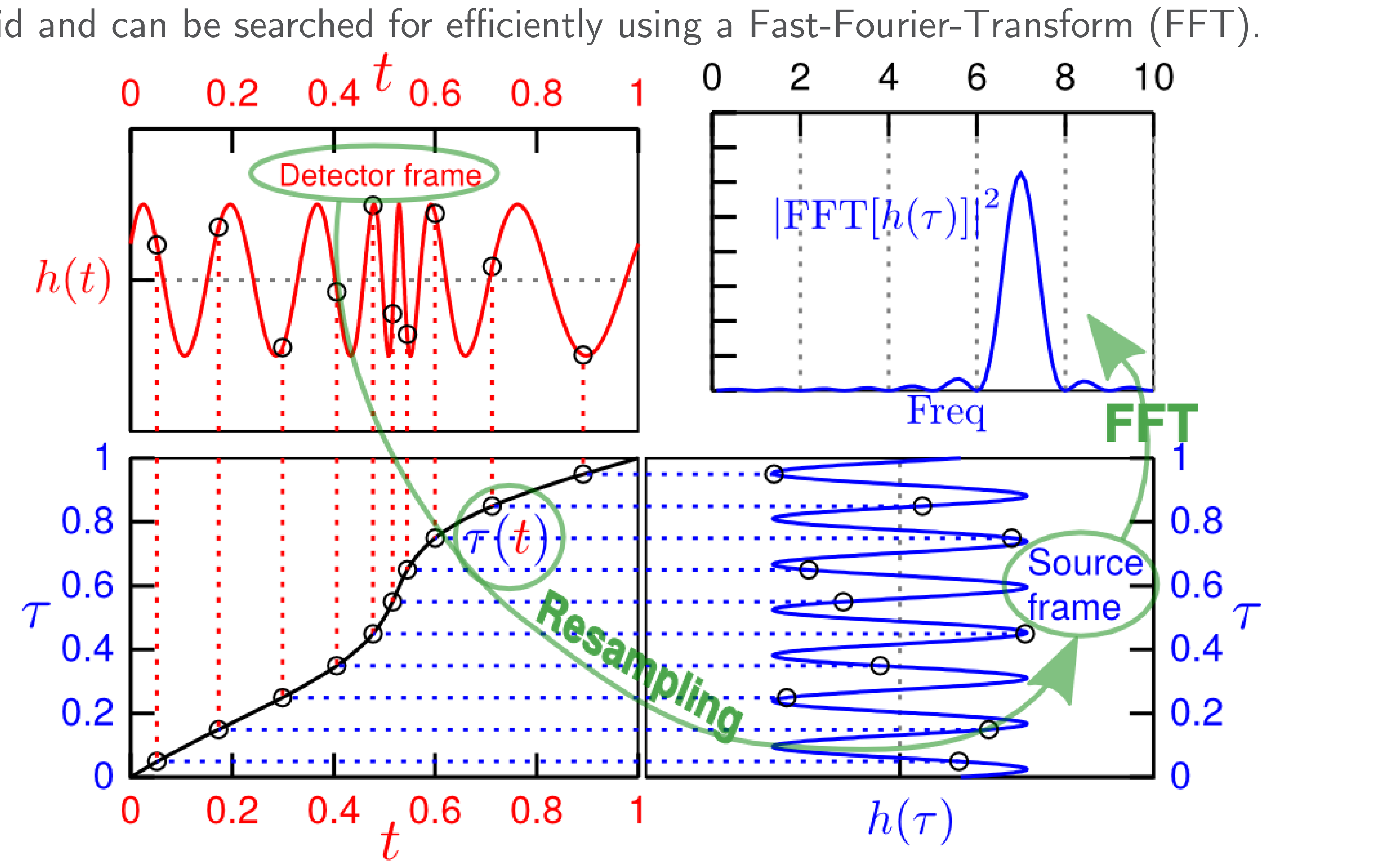
Continuous gravitational waves

The signal phase $\Phi_{\rm src}(\tau)$ in the source frame arriving at a detector from direction $\vec{n}$ is Doppler modulated to $\Phi_{\rm det}(t)$ by detector motion (around Earth+Sun) and intrinsic source motion.
The $\mathcal{F}$-statistic [1] is a coherent matched-filter maximized over the unknown amplitude and polarization parameters.



Source-frame resampling + FFT

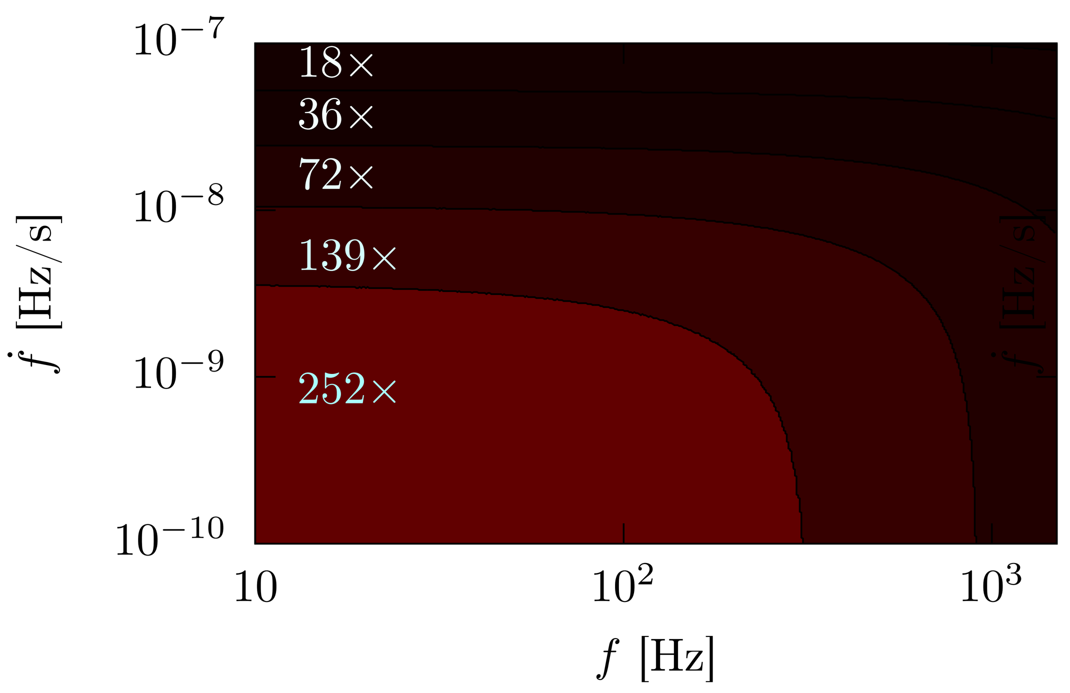
The idea of *resampling* [1] is to *revert* the Doppler time-delays $\tau(t;\vec{n})$ and to *reconstruct* a uniformly-sampled signal timeseries in the source frame, $\Phi_{\rm src}(\tau)$, by interpolation. After applying spindown phase-corrections $e^{i\,2\pi(\dot{f}\,t^2+\ldots)}$, the signal is effectively a sinusoid and can be searched for efficiently using a Fast-Fourier-Transform (FFT).



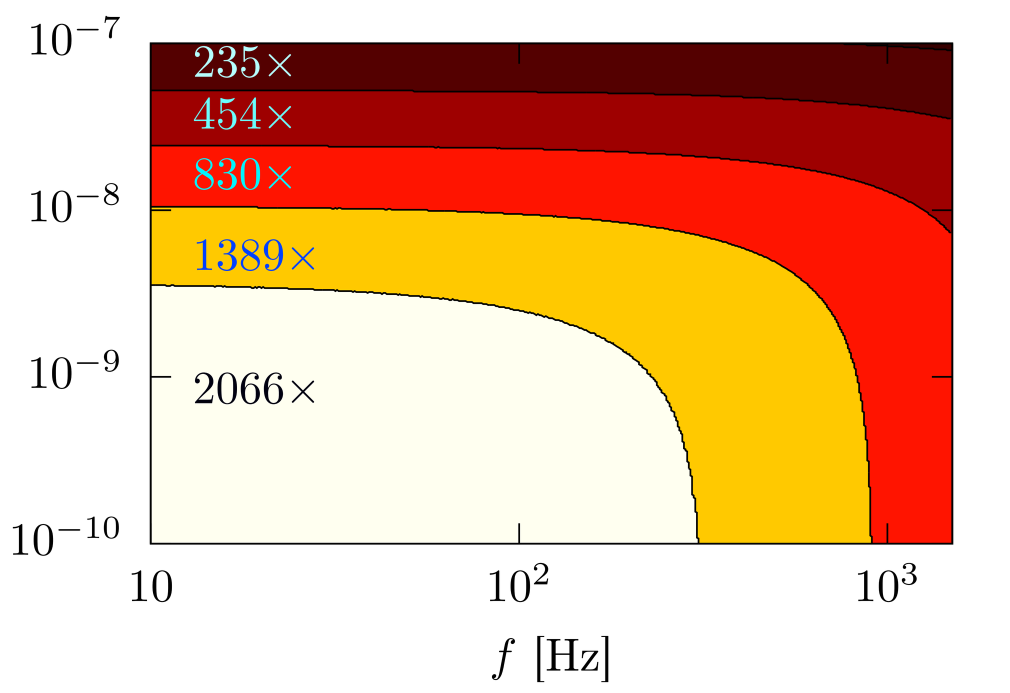
Example: Einstein@Home speedup on `Thinkpad T520`

Speed-up factors compared to previous ("Demod") $\mathcal{F}$-implementation:

CPU version (Xeon E3-1505M 3GHz):



GPU version (Quadro M2200, 1024 cores):



☞ speedup of GPU version compared to CPU: $\sim\mathcal{O}(10)$ (with FFT-length $\in[2^{19},2^{23}]$)

[Assumed run setup: $T_{\rm span} = 120\,\mathrm{days}$, $T_{\rm seg} = 20\,\mathrm{days}$, $df = 3\times10^{-7}\,\mathrm{Hz}$, $\Delta f = 50\,\mathrm{mHz}$]

Improvements

over previous resampling codes [2,3]:

- arbitrary length and frequency resolution $df = 1/T_{\rm source}$ by zero-padding source timeseries
- optimized FFT length by interpolating onto $N_{\rm source} = 2^n$ samples in the source frame
- optimal sinc-interpolation used for resampling into source frame
- reduced memory footprint via a shared "workspace" across semi-coherent segments

Status and future work

- In use since first aLIGO all-sky Einstein@Home search
- Completed GPU port (using openCL) to be rolled out soon
- Large speedup (CPU $\sim\mathcal{O}(100)$, GPU $\sim\mathcal{O}(1000)$) compared to our previous $\mathcal{F}$-statistic implementation

References

[1] P. Jaranowski, A. Królak, B. F. Schutz, PRD**58**, 063001 (1998).

[2] P. Patel, X. Siemens, R. Dupuis, J. Betzwieser, PRD**81**, 084032 (2010).

[3] P. Astone, K. M. Borkowski, P. Jaranowski, A. Królak, PRD**65**, 042003 (2002).

[4] `XLALComputeFstat_Resamp()`, in `http://github.com/lscsoft/lalsuite`

[5] `http://einstein.phys.uwm.edu`

einstein @ home