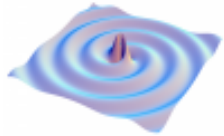


# Excess power DSO

W. Anderson, P. Brady, D. Brown,  
E. Flanagan, E. Katsavounidis, . . .

Burst Summit 12/06/01



## How it works

---

The basic idea:

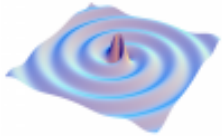
1. Pick a start time  $t_s$ , a time duration  $\delta t$  (containing  $N$  data samples), and a frequency band  $[f_s, f_s + \delta f]$ .
2. Fast Fourier transform (FFT) the block of (time domain) detector data for the chosen duration and start time.
3. Sum the power in the frequency band  $[f_s, f_s + \delta f]$ .
4. Calculate the probability of having obtained the summed power from Gaussian noise alone using a  $\chi^2$  distribution with  $2 \times \delta t \times \delta f$  degrees of freedom.
5. If the probability is significant, record a detection.
6. Repeat the process for all desirable choices of start times  $t_s$ , durations  $\delta t$ , starting frequencies  $f_s$  and bandwidths  $\delta f$ .



## Status

---

- LAL Code: source, documentation and test routines exist.
- DSO Code: source, documentation and example scripts exist.
- MDC: tested to run under LDAS using simulated data, produces events that can be inserted into the database:
  - Bug reports: none open, several serious bugs fixed.
  - Efficiency: improved by better communication (  $\sim 8 \times$  real time).
  - Duration/Band: corrected some problems with how these were computed.
- Run on inspiral challenge data:
  - Code could detect weakest signal injected: binary inspiral with  $m_1 = 2.2$ ,  $m_2 = 3.6$  at 55Mpc in simulated LIGO-I noise.
  - Matched filtering SNR  $\sim 10 - 12$ .



## Command line paramaters

---

`argv[0] = "-filterparams"`

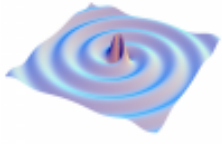
`argv[1] = numPoints` Number of data points in a segment. Note that the current implementation of the code must have this number of the form  $2^{p+1} + 2$ . The maximum expected time duration of the signal should be `numPoints` × sampling rate of data passed to the code.

`argv[2] = numSegments` Number of overlapping segments into which data should be divided for filtering.

`argv[3] = overlap` Number of points overlap between segments. This is an argument for completeness, but in general it should be `numPoints/2`

`argv[4] = overlapFactor` Amount of overlap between neighboring TF tiles. Currently see LAL `burstsearch` package for details.

`argv[5] = minFreqBins` Smallest extent in freq of TF tiles to search. Currently see LAL `burstsearch` package for details.



## Command line parameters (cont'd)

---

`argv[6] = minTimeBins` Smallest extent in time of TF tiles to search Currently see LAL `burstsearch` package for details.

`argv[7] = flow` Lowest frequency in Hz to be searched Currently see LAL `burstsearch` package for details.

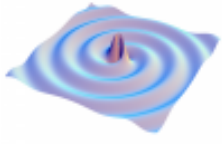
`argv[8] = deltaF` Frequency resolution of first TF plane Currently see LAL `burstsearch` package for details.

`argv[9] = length` Length ( $N_F$ ) of first TF plane (with  $N_T = 1$ ) Currently see LAL `burstsearch` package for details.

`argv[10] = numSigmaMin` threshold number of sigma Currently see LAL `burstsearch` package for details.

`argv[11] = alphaDefault` default alpha value for tiles with sigma < numSigmaMin. Currently see LAL `burstsearch` package for details.

`argv[12] = segDutyCycle` Number of segments sent to slave for analysis Currently see LAL `burstsearch` package for details.



## Command line parameters (cont'd)

---

`argv[13] = alphaThreshold` Identify events with alpha less than this Currently see LAL `burstsearch` package for details.

`argv[14] = events2Master` Number of events to be accepted from a search over `numPoints` of data.

`argv[15] = channelName` The name used to identify the data to be analyzed in the `multiDimData`.