

**LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY**  
**-LIGO-**  
**CALIFORNIA INSTITUTE OF TECHNOLOGY**

**Looking for Chirps in:**

**24 hours of Gaussian Noise**

**24hours of Accelerometer data**

(at south end station LLO)

Publication: # G000330-00-R 8/00

**by**

Anthony Rizzi

**Distribution:**

**all**

**This is a publication of the LIGO Project.**

Grades

$1.4M_{\odot} - 1.4M_{\odot}$  Binary Inspiral



## Optimal Filtering

$$S = 2 \int_{-1024 \text{ Hz}}^{1024 \text{ Hz}} df \frac{\tilde{h}(f) \tilde{T}(f)}{S_h(|f|)} e^{-2\pi i f t_0}$$

$S$  = "Signal" = Signal to Noise Ratio when Noise normalized to 1.

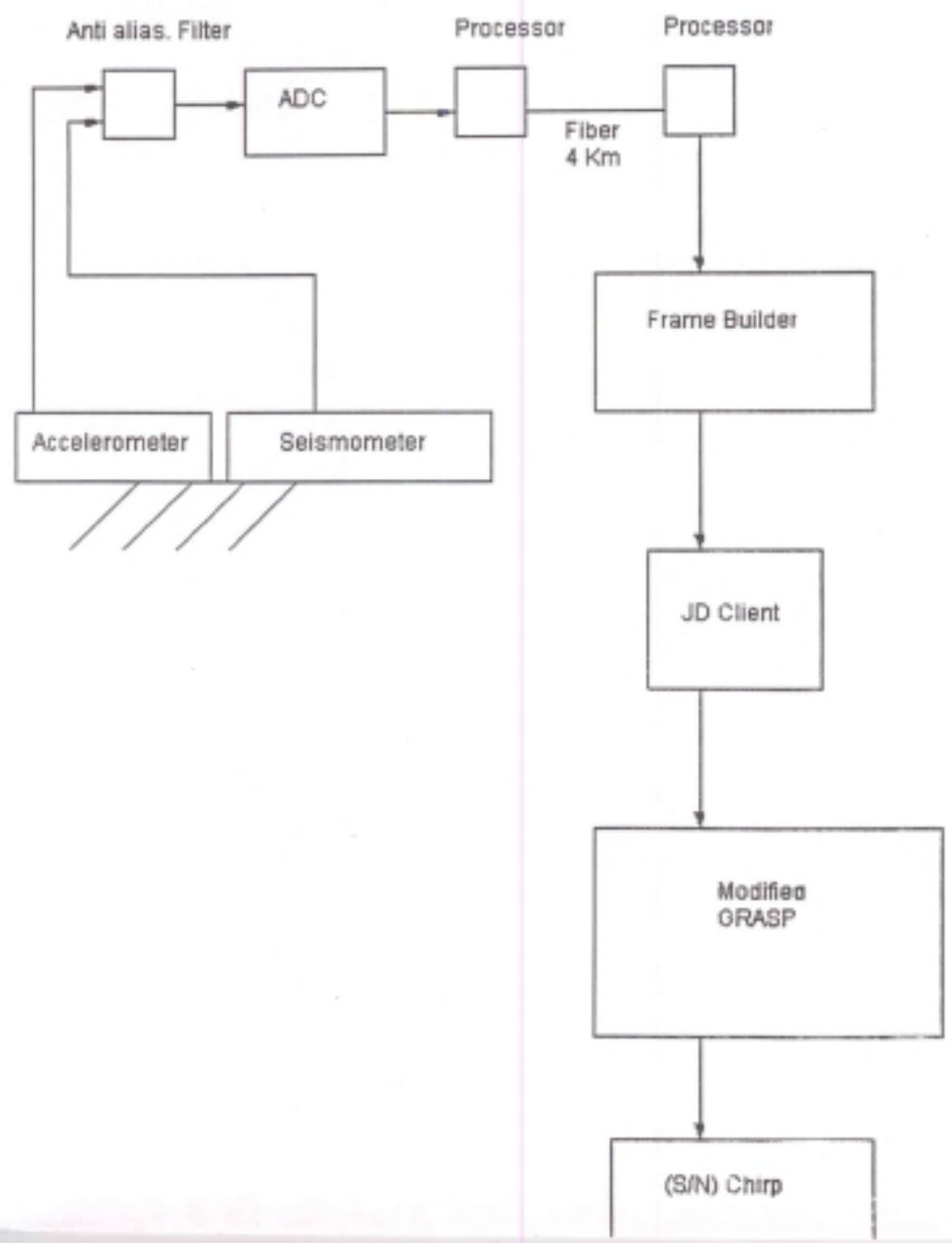
$h(f)$  = interferometer output, in frequency domain

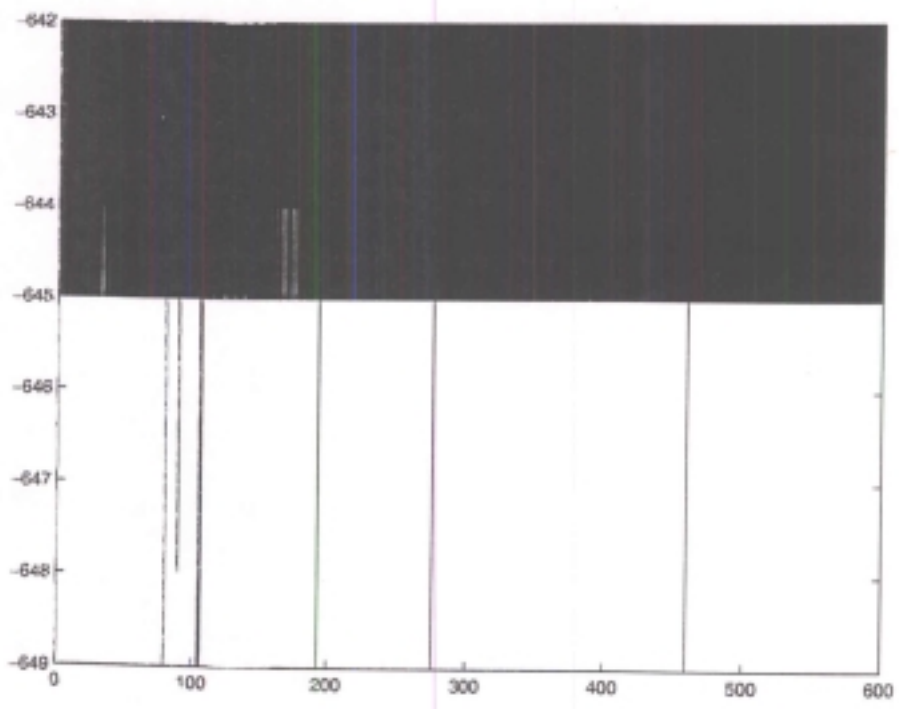
$T(f)$  = template. The chirp in the frequency domain.

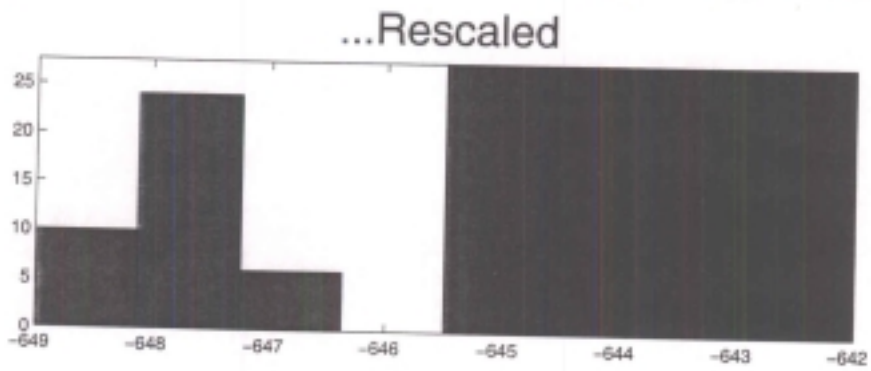
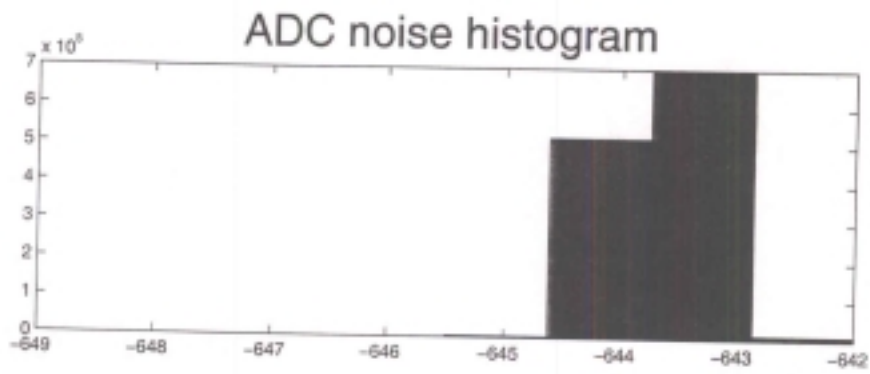
$S_h(|f|)$  = power spectrum

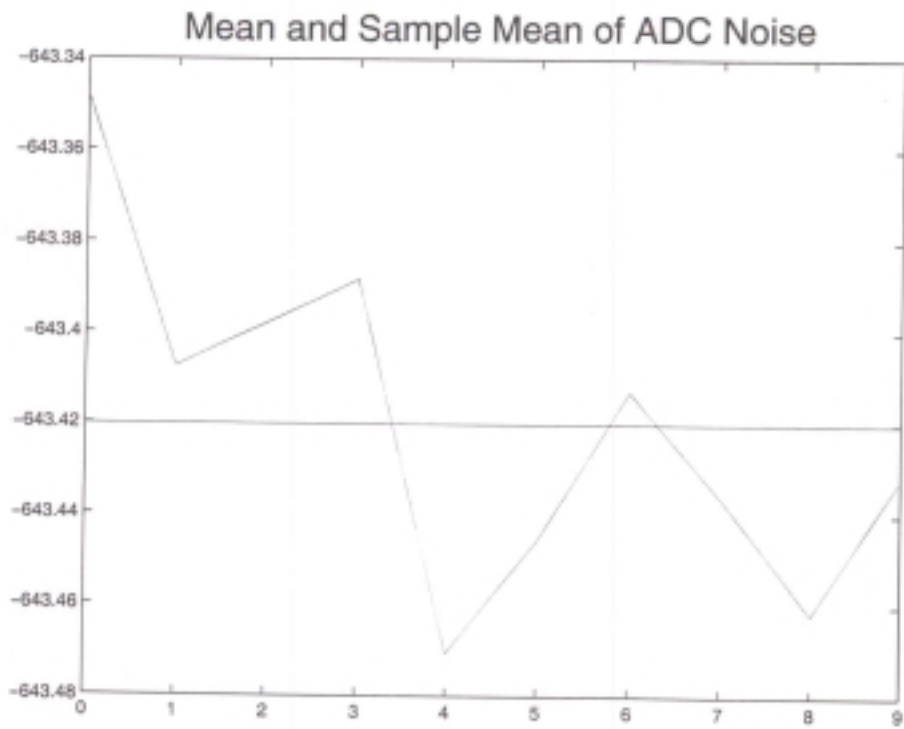
$t_0$  = offset in the time domain

# Block Diagram 1

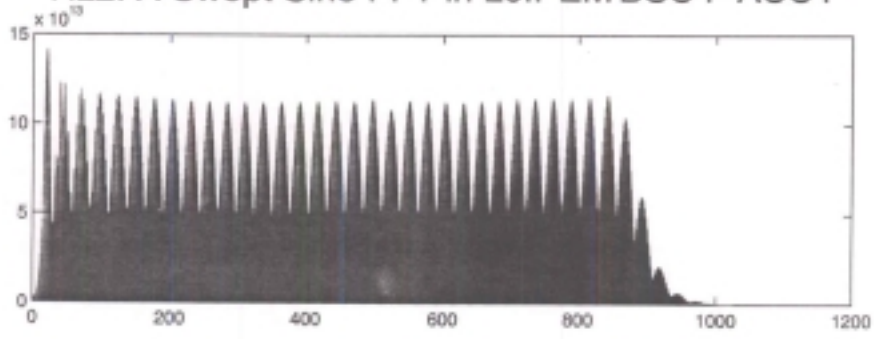




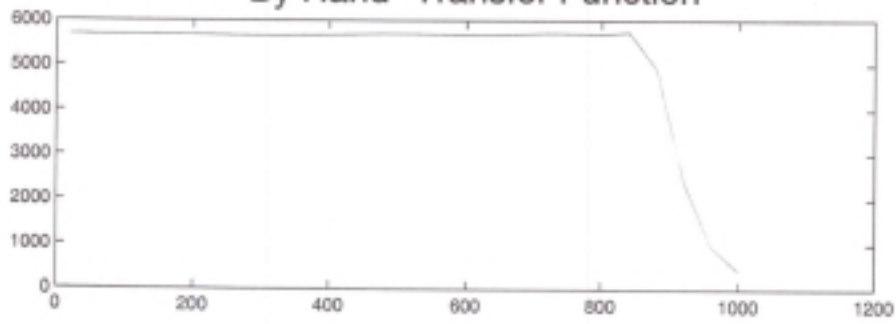




Hz2: A Swept Sine FFT in L0:PEMBSC4-ACCY

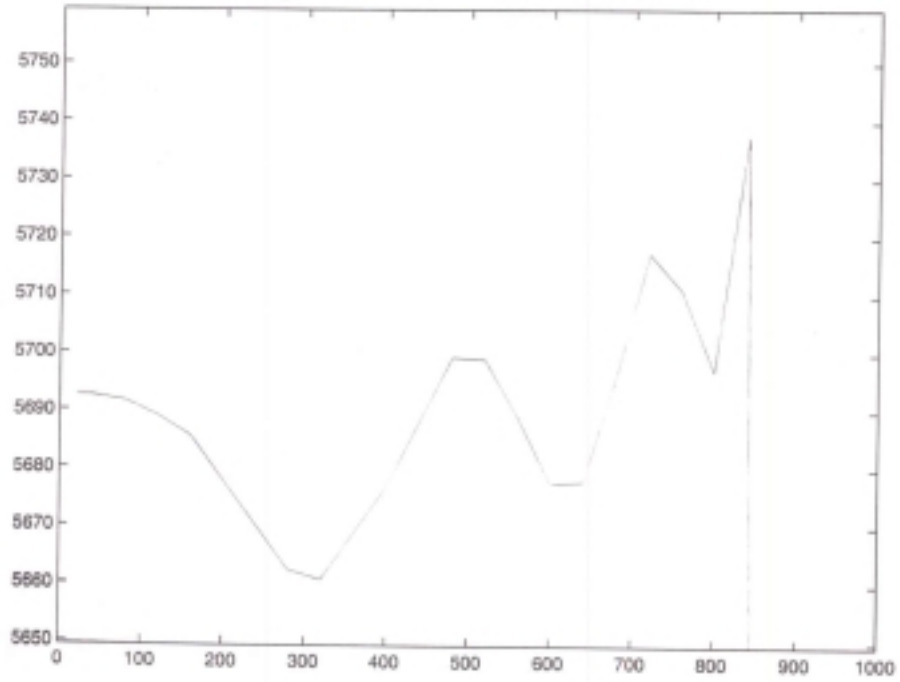


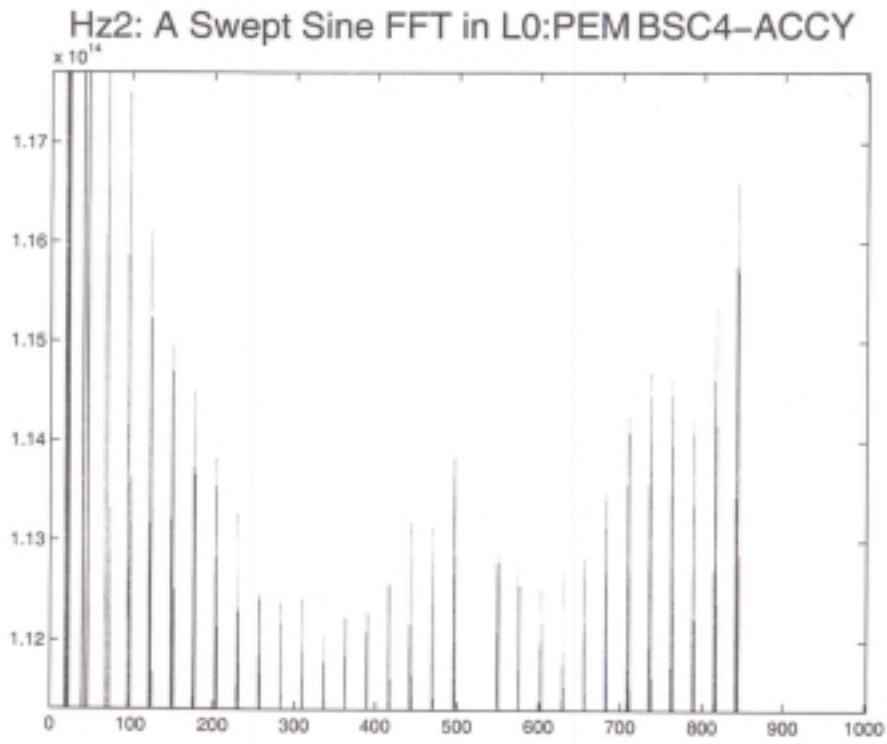
"By Hand" Transfer Function





"By Hand" Transfer Function

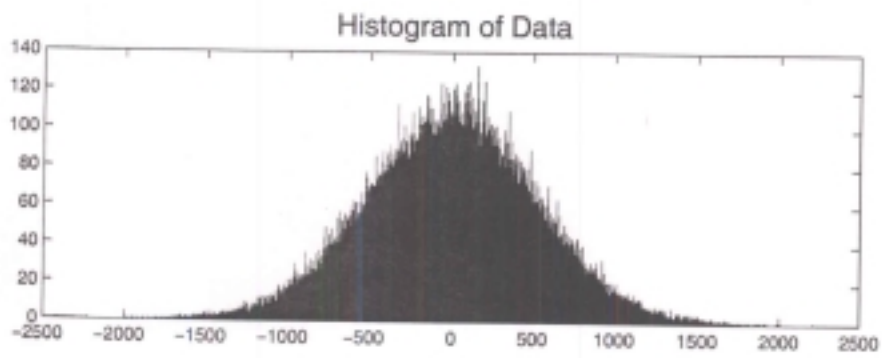
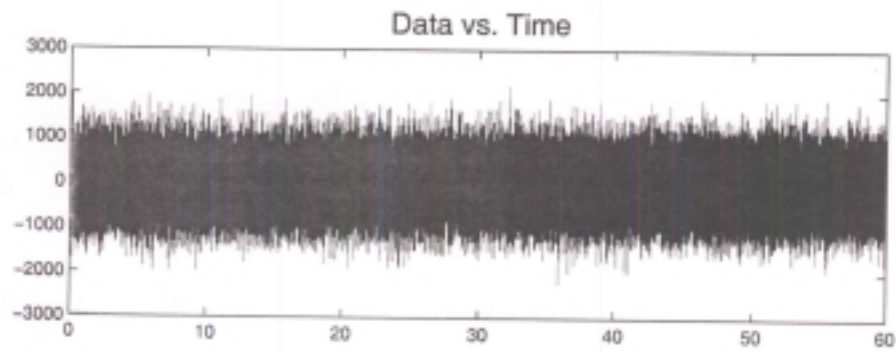




make\_file.m  
/home/dfabryck/noiseinj/

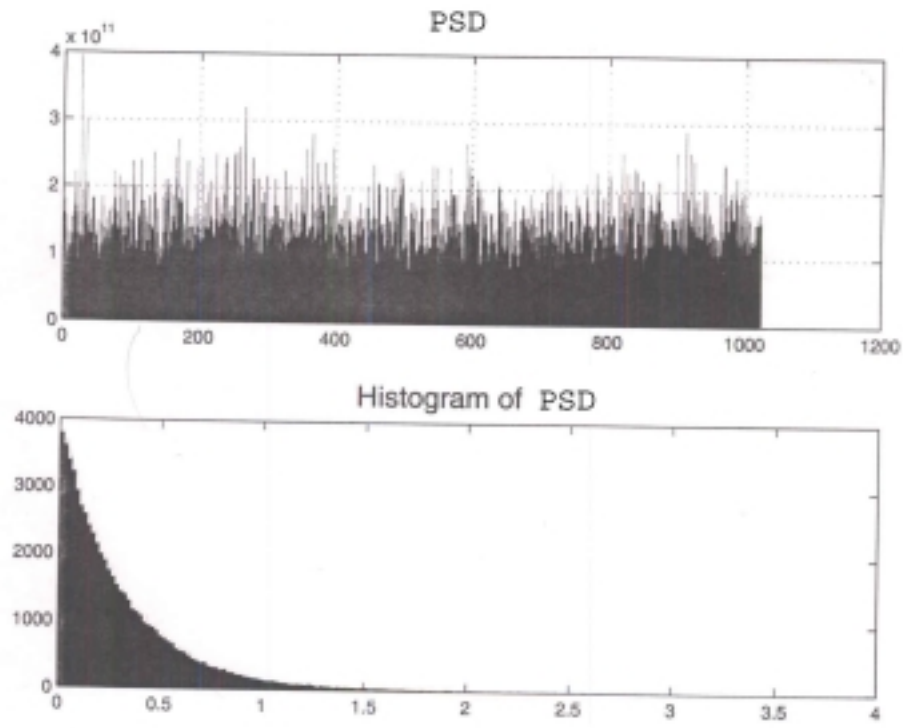
1/1  
Aug 12 2000

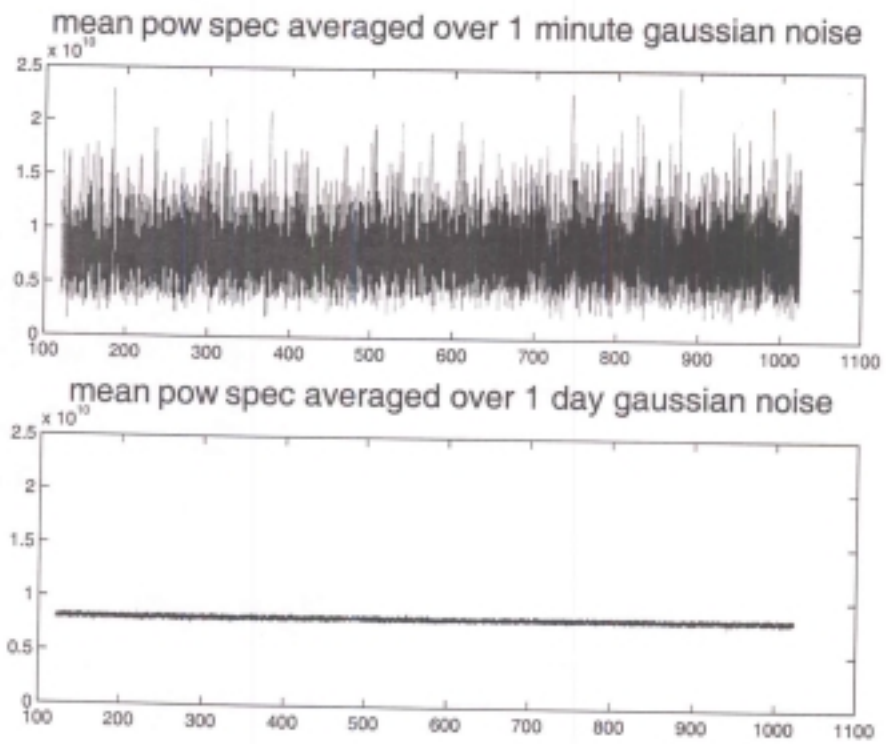
```
T=60  
t=(1/2048:1/2048:T);  
SD=500;  
y=SD*randn(size(t));  
Y=floor(y);  
Y=transpose(Y);  
  
fid=fopen('matlab_noise', 'w');  
fprintf(fid, '%4.0f \n', Y);  
status=fclose(fid);
```

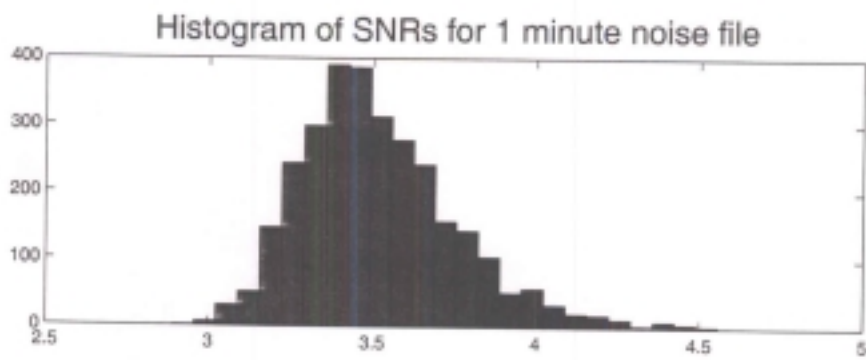
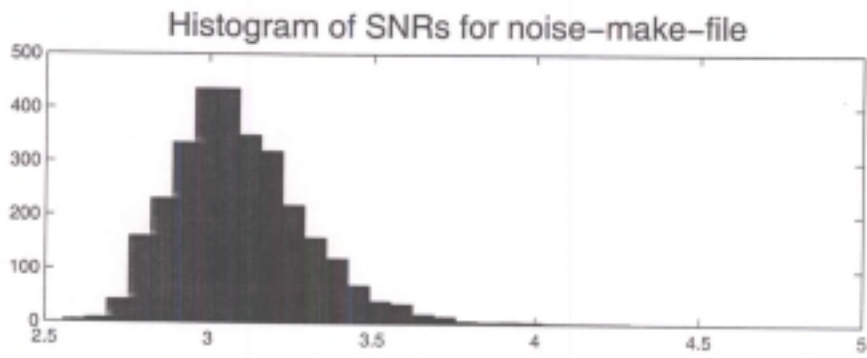


$$\sigma = 500$$

$$\bar{x} = 0$$

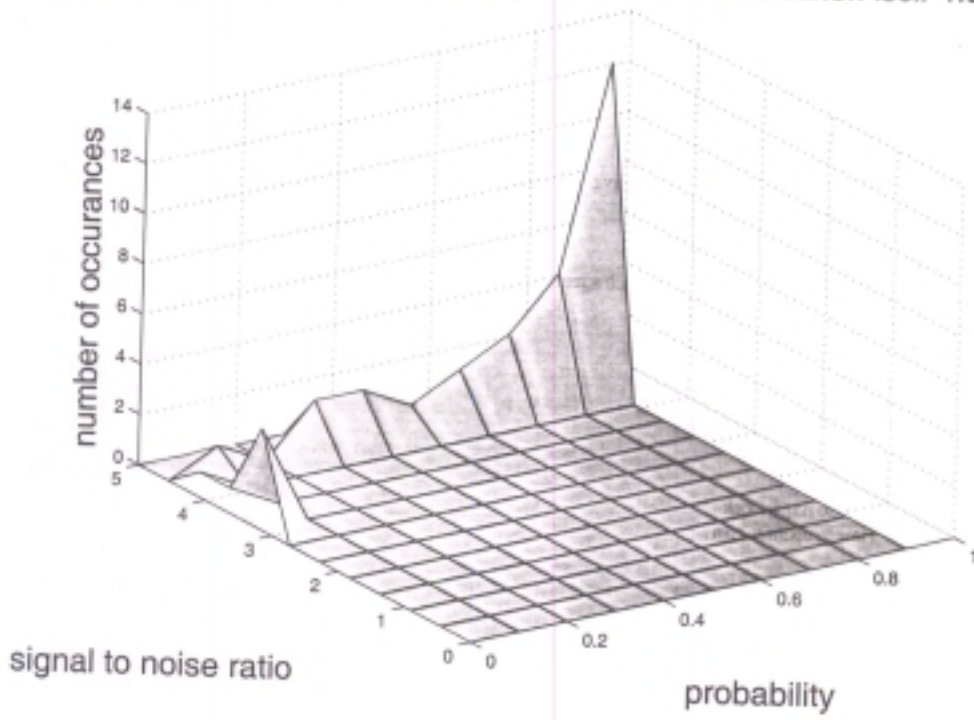






Optimal Filtering on  
Gaussian Noise file  
using  
non-local noise average  
of 1st 10 segments

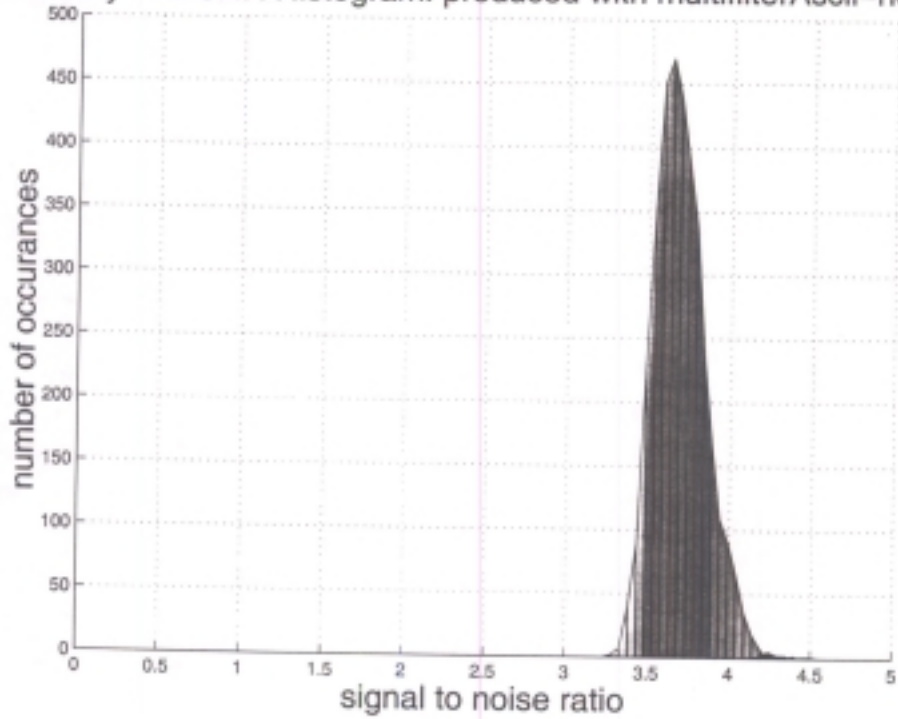
Probability and SNR Histogram: produced with multifilterAscii-noise use

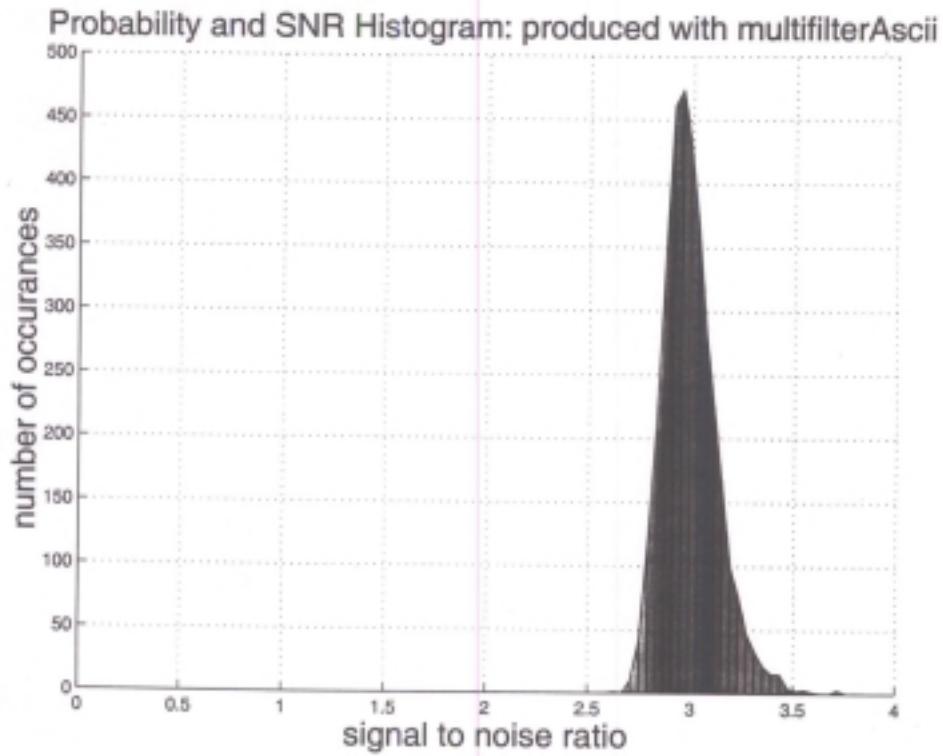




Optimal filtering on Gaussian Noise File  
using  
non-local noise calculation  
on 10 data segments at beginning  
of file

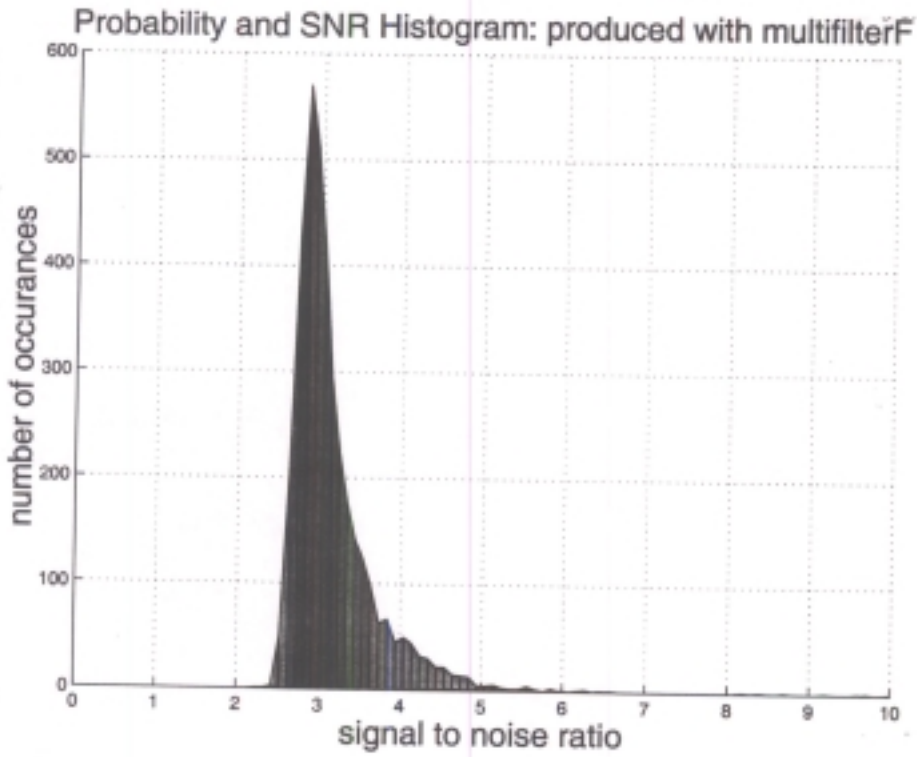
Probability and SNR Histogram: produced with multifilterAscii-noise use





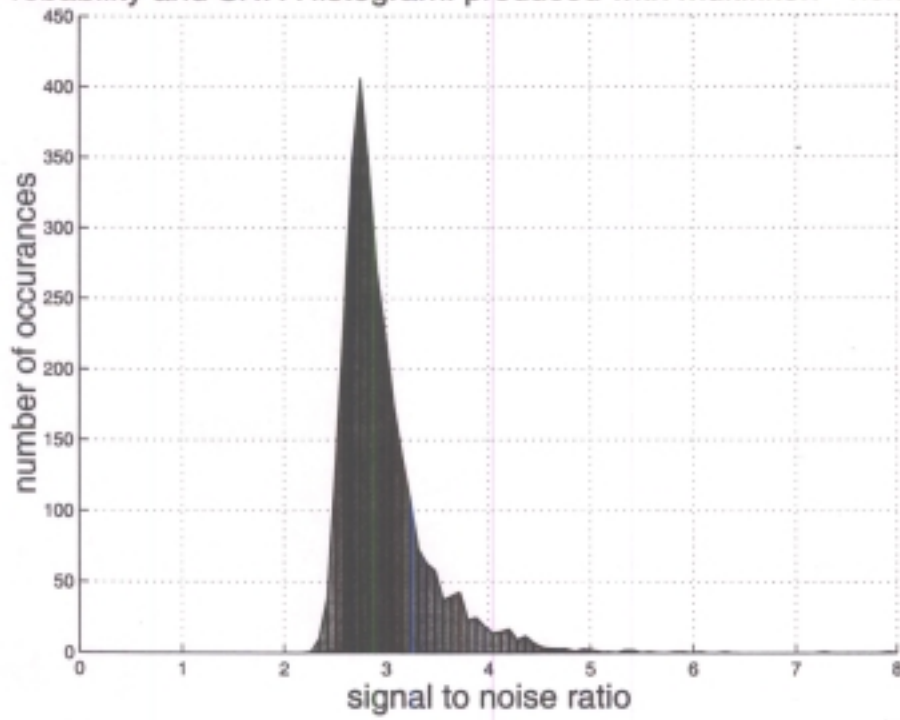
Optimal Filtering on Gaussian File  
using noise calculated from  
local (exponential) average

Optimal Filtering on  
accelerometer data (12/18)  
using  
local exponential averaging



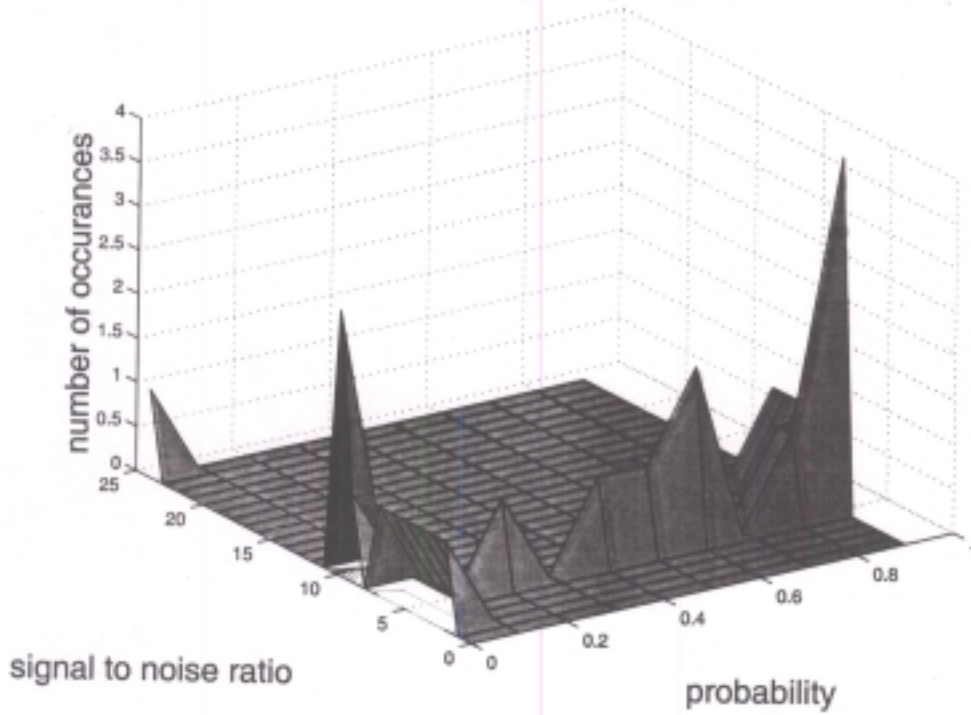
Optimal Filtering on  
Accelerometer data  
using non-local average  
(1st 10 segments)

Probability and SNR Histogram: produced with multifilterF-noise use



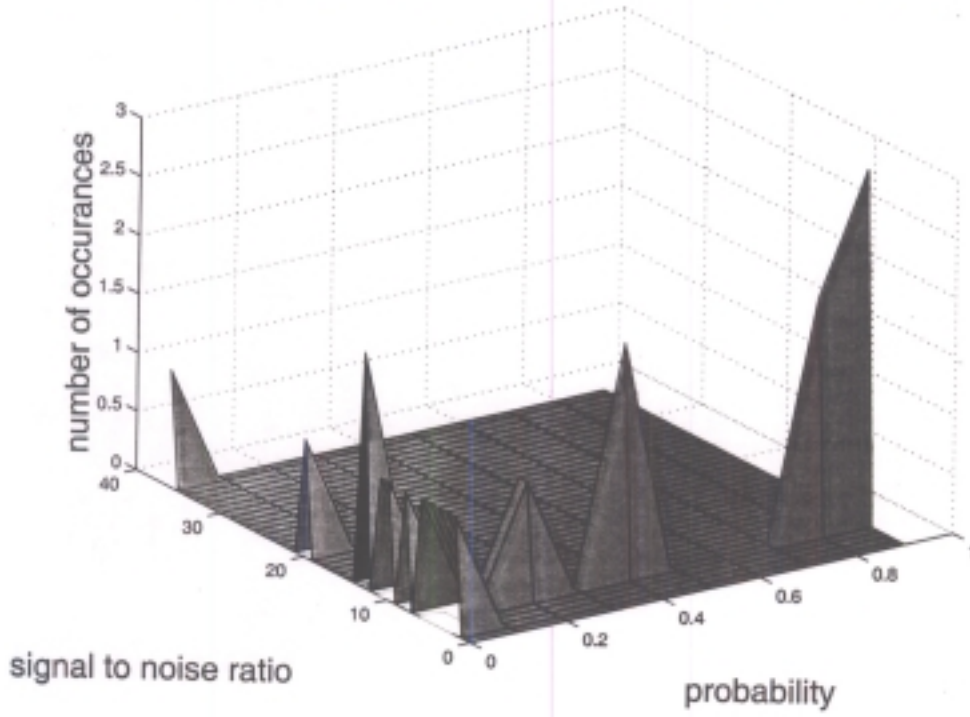
Optimal Filtering on  
Accelerometer data  
using local exponential averaging

Probability and SNR Histogram: produced with multifilterF



Optimal Filtering on  
Accelerometer data  
using  
non-local noise  
(1st 10 segments)

Probability and SNR Histogram: produced with multifilterF-noise use



Optimal Filtering on  
Gaussian noise file  
using local exponential averaging

Probability and SNR Histogram: produced with multifilterAscii

