MATLAB FFT / Spectrum NORMALIZATION

There is some useful information in the following links:

<http://www.mathworks.cn/matlabcentral/newsreader/view_thread/270119>

<http://www.mathworks.com/matlabcentral/newsreader/view_thread/242155>

<http://groups.google.com/group/comp.soft-sys.matlab/msg/2222327db2ea7f51>

There are two conventions for normalizing the FTT/DFT: The DFT and IDFT formulae required for Parseval's theorem to hold are

N

X(k) =1/sqrt(N) \* sum x(n)\*exp(-j\*2\*pi\*(k-1)\*(n-1)/N), 1 <= k <= N

n=1

N

x(n) = 1/sqrt(N) \* sum X(k)\*exp( j\*2\*pi\*(k-1)\*(n-1)/N), 1 <= n <= N.

k=1

Instead of using the above, MATLAB uses the following convention (most common amongst Engineers):

N

X(k) = sum x(n)\*exp(-j\*2\*pi\*(k-1)\*(n-1)/N), 1 <= k <= N

n=1

N

x(n) = 1/N \* sum X(k)\*exp( j\*2\*pi\*(k-1)\*(n-1)/N), 1 <= n <= N.

k=1

Normalization Examples:

Fs=10000;

x=randn(1,1024);

X=fft(x);

plot((0:length(X)-1)/length(X)\*Fs,abs(X)/1024)

note that for this example Parsevals holds as follows:

mean(x.^2)=sum(sum(abs(X).^2/1024.^2))

For zero padded data:

Fs=10000;

NFFT=1024\*8;

x=randn(1,1024);

X=fft(x);

plot((0:length(X)-1)/length(X)\*Fs,abs(X)/1024)

Note that for parsevals to hold I still normalizat by the length(x) (“length of the data”) not NFFT!

mean(x.^2)=sum(sum(abs(X).^2/1024.^2))

So the proper normalization requires that I compute:

X=fft(x,NFFT)/length(x);

How about some power spectral density examples. In this case:

Fs=10000;

NFFT=1024\*4;

x=randn(1,1024\*128);

Pxx=psd(x,NFFT,Fs);

Parseval holds as follows:

mean(x.^2)=sum(Pxx/(NFFT/2)) or sum(Pxx/numel(Pxx))

Note that

Pxx/NFFT\*2 is the average power in each frequency channel.

NORMALIZING as V^2/Hz (NEED TO CHECK THIS!!!!!)

In reality, the normalizing for PSD is in power over Hz. For example, if the signal is in units of volts the power spectral density is expressed in units of V^2/Hz. How do we do this normalization? Simply take

Pxx\_normalized=Pxx / Fs\*NFFT

Note that spectral resolution per band is

df=Fs/NFFT

CHIRP SIGNALS

When we use the chirp signals for calibration I’ve noticed the following. Pxx/NFFT\*2 does note give the desired value. I believe there are two reasons for this:

1. Pxx/NFFT\*2 gives the average power. Note however, that for the chirp the power is averaged over 10 seconds even though there is power at a given frequency for only a brief instant. I believe the way to correct for this is to add 20\*log10(T) where T is the stimulus duration. This will give a power which is approximately the MAXIMUM power at any given frequency.
2. Also, it appears that windowing effects may be limiting the resolution of the measurement. Typically, even if (1) correction is applied the MAXIMUM power is still underestimated. The reason for this may be because of windowing effects. If NFFT is too large, you average over several time segments and cannot accurately estimate the maximum power. If NFFT is too small, then you smooth too much in the frequency domain and the power across multiple frequencies gets averaged out.