Feb/06/06

written by: Chieh

Using matlab to simulate a system with an impulse and noise signals

Consider the transfer function:

$$P = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

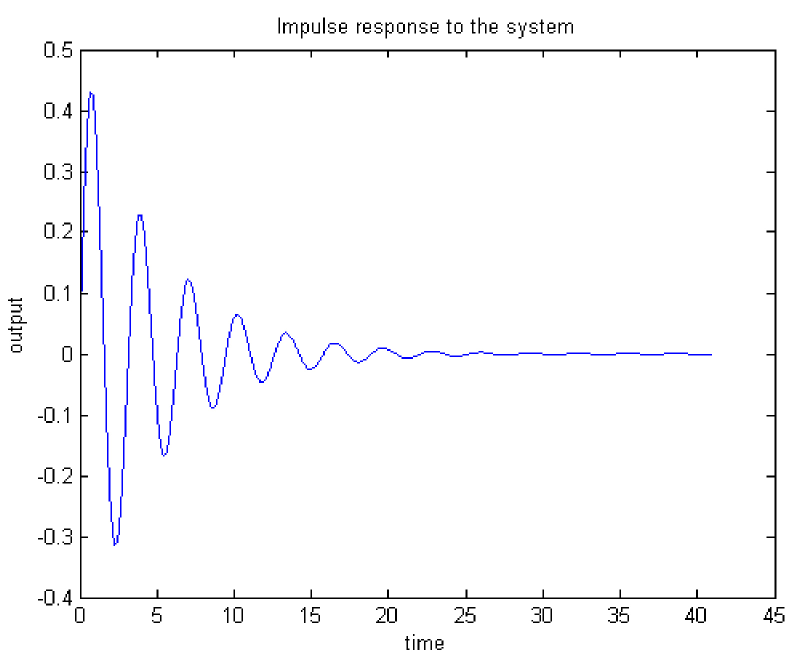
where ζ 1/10 and $\omega_n = 2$

Let's play with this system by simulate it with an impulse response.

You can easily do so in matlab by using these commands

```
%time_freq() is a seperate pre-written matlab function
[time,freq,df] = time_freq()
sys = tf(1,[1,2*c*w,w*w]);
[Y,T,X] = impulse(sys,time,0.01,0);
figure(1) plot(time,Y);
title('Impulse response to the system');
xlabel('time');
ylabel('output');
```

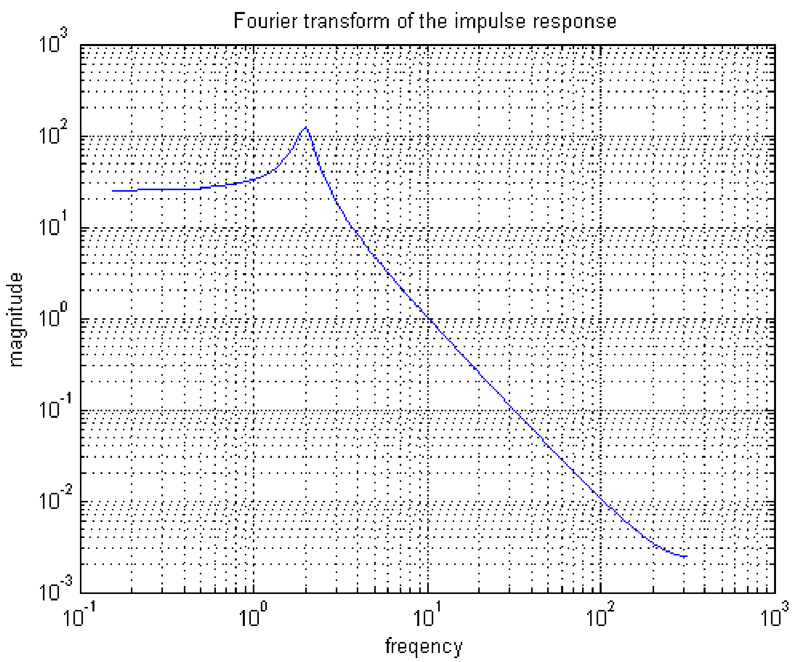
The output would look like this



Now if we take the FFT of the signal with these commands

```
f = abs(fft(Y,4096))
figure(2)
loglog(freq,f(1:2048));
grid on;
title('Fourier transform of the impulse response')
xlabel('freqency');
ylabel('magnitude')
```

the FFT plot would look like this

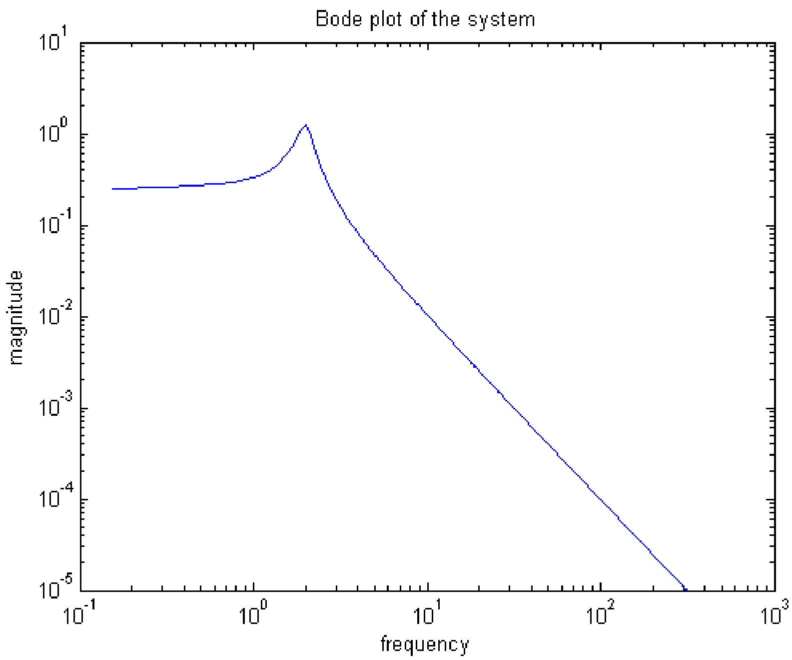


if we draw the bode plot they should look identical , the commands in matlab are

```
figure(3)
mag = bode(sys, freq);
```

```
mag = squeeze(mag(1,1,:));

loglog(freq,mag)

title('Bode plot of the system')

xlabel('frequency')

ylabel('magnitude')
```

the Boddi plot should look like this



Next, what if we want to input noise instead of an impulse function. First we would need to generate a noise function. For the sake of our problem we will generate 4096 samples off the normal distribution centered at zero with a variance of 1. The noise could be simulated using this command.

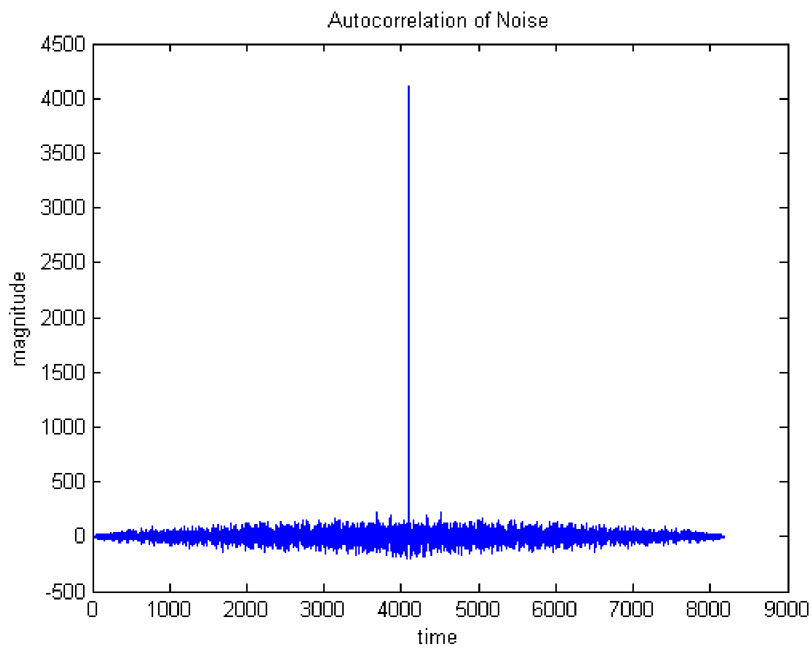
```
noise = randn(4096,1);
```

This command generates 4096 by 1 vector of the normal distribution. Let's take a look at the frequency of this noise by finding the Power Spectrum Density. But first, since the noise is a non-deterministic function, we can't just take the fourier transform to get the PSD. We would need to find the autocorrelation and take the fourier transform of that function to get the PSD.

To find the autocorrelation we would use these commands.

```
figure(4)
noise = randn(4096,1);
cmag = xcorr(noise);
plot([1:8191],cmag)
title('Autocorrelation of Noise')
xlabel('time')
ylabel('magnitude')
```

The autocorrelation looks like this.



When you are taking the autocorrelation of a function, you use a flipped convolution. It makes sense that there's a spike at the center because noise itself does not correlate with itself unless it has overlapped with each other.

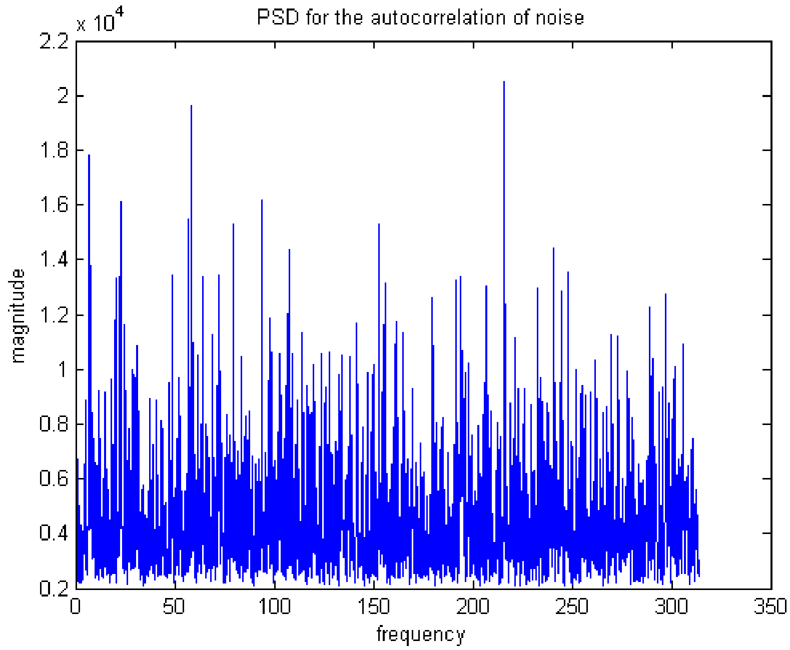
Now if we want to check out the PSD of this autocorrelation we would use these commands, following the previous commands.

```
fouri = abs(fft(cmag,4096));
figure(5)
plot(freq,fouri(1:2048))
title('PSD for the autocorrelation of noise')
```

xlabel('frequency')

ylabel('magnitude')

PSD for the noise signal



It makes perfect sense that the noise frequency is everywhere because, it is after all noise and is suppose to have an infinite range of frequency.

Ok, now that we finally have noise, let's stick it into the system and see how the system will respond. Shall we?? We would use these commands in matlab.

figure(6)

y = lsim(sys,noise,time);

Ymag = xcorr(y);

ps = abs(fft(Ymag,4096));

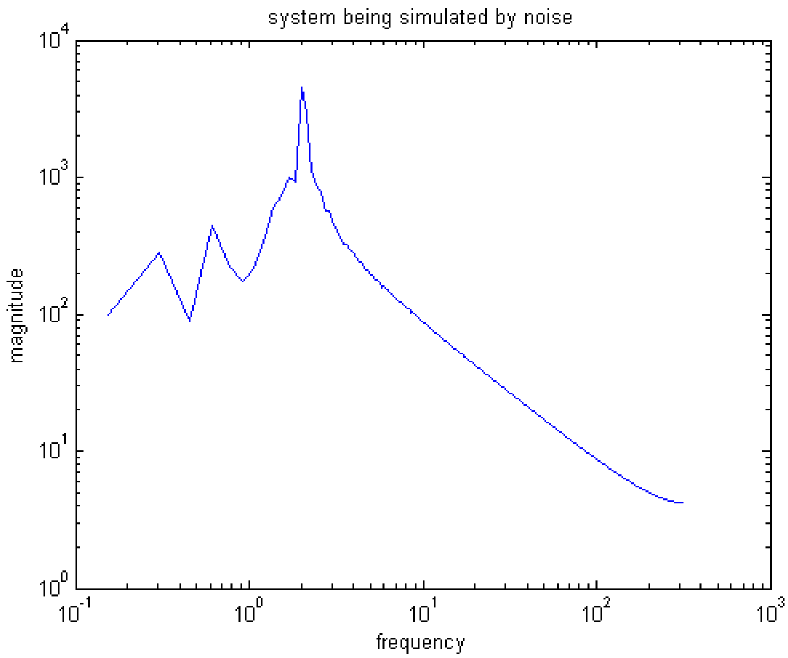
loglog(freq,ps(1:2048))

title('system being simulated by noise')

xlabel('frequency')

ylabel('magnitude')

The output signal would look something like this.



Now if we compare the response to the bode plot squared below, they should look very similar