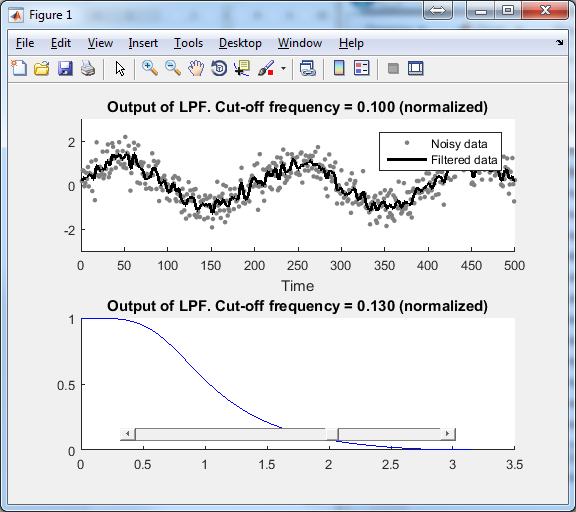
Lab 7 Assignment

YingTa Lin (ytl287)

Saturday 17th November, 2015

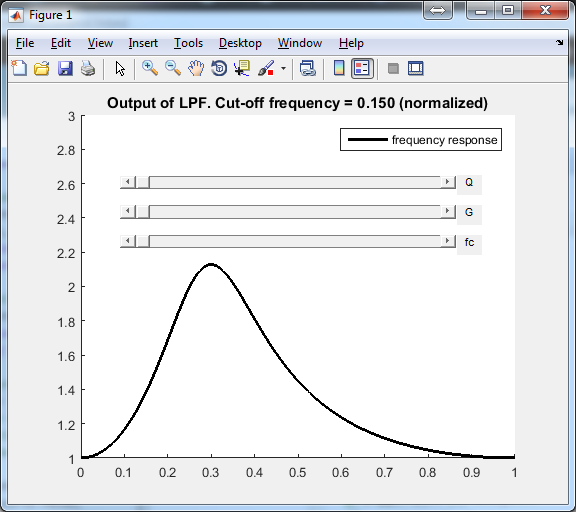
**Assignment 1:**

See “Lab7\_1\_ytl287.m”



**Assignment 2:**

See “Lab7\_2\_ytl287.m” (This code needs Matlab2015a to run)



**Assignment 3:**

Nothing need to be submit

**Assignment 4:**

An All-pass filter is defined by the reverse coefficient of the nominator and denominator of the H(z) function. By using coefficients [1+alpha -2\*cos(wc) 1-alpha], wc can be defined. Instead of using noise data, I create a chirp wav sound to verify that the notch filter is working.

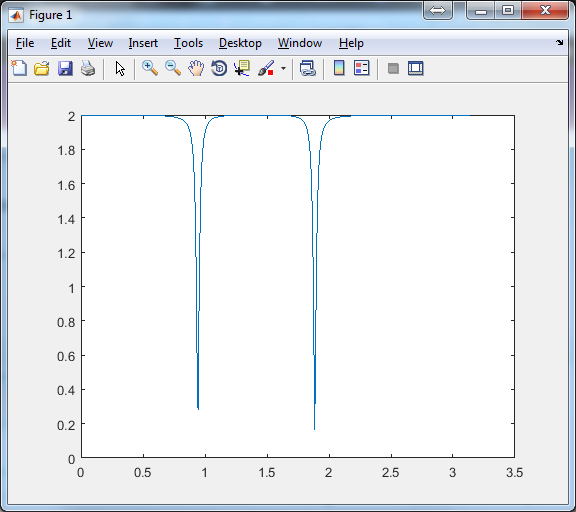
**Assignment 5:**

See “Lab7\_5\_ytl287.pdf”

By cascade two 2-order all-pass notch filters, I get two notches. The frequency can be set by wc1 and wc2 used by the code fragment below.

|  |
| --- |
| %% Order-2 example  wc1 = 0.3\*pi;  alpha1 = 0.02;  a1 = [1+alpha1 -2\*cos(wc1) 1-alpha1];  b1 = flip(a1);    wc2 = 0.6\*pi;  alpha2 = 0.02;  a2 = [1+alpha2 -2\*cos(wc2) 1-alpha2];  b2 = flip(a2);    a = conv(a1,a2);  b = conv(b1,b2);  [H, om] = freqz(b, a);  plot(om, abs( Id + H ) );  % Now the total system is a notch filter with a null.  % Where is the null?  % It is at the frequency where the all-pass filter is -1.  % At om = pi, the frequency response of the total system  % is 2 because the all-pass system is 1 at that frequency  % so it adds to the direct path. |

The frequency response of the LTI system looks like the figure below.



**Assignment 6:**

See “Lab7\_6\_ytl287.py”

Compare to flanger filter, it sounds the same because we didn’t assign oscillation to the weep depth, it is a fixed frequency phasor.