Christopher Bero
CPE 381
Spring 2015

Project Phase 2 Report

Task

Create a program which can implement a pre-made FIR or IIR discrete filter on the signal embedded in a 44100 or 22050Hz, 16 bit, PCM wave file.

Proposed Solution

I intend to reimplement the main code structure developed in phase 1 of this project to open, read, and process wave files. In class we learned to use *fdatool* in Matlab for the purpose of generating a C style header with FIR or IIR coefficient arrays. The general algorithm for processing with this array, supposing it is FIR, will be:
- Keep a running buffer of samples the same size of the array.
- For each sample shifted into the buffer, multiply each corresponding element in the buffer against the filter.
- Sum the results and store them as the filtered output value.

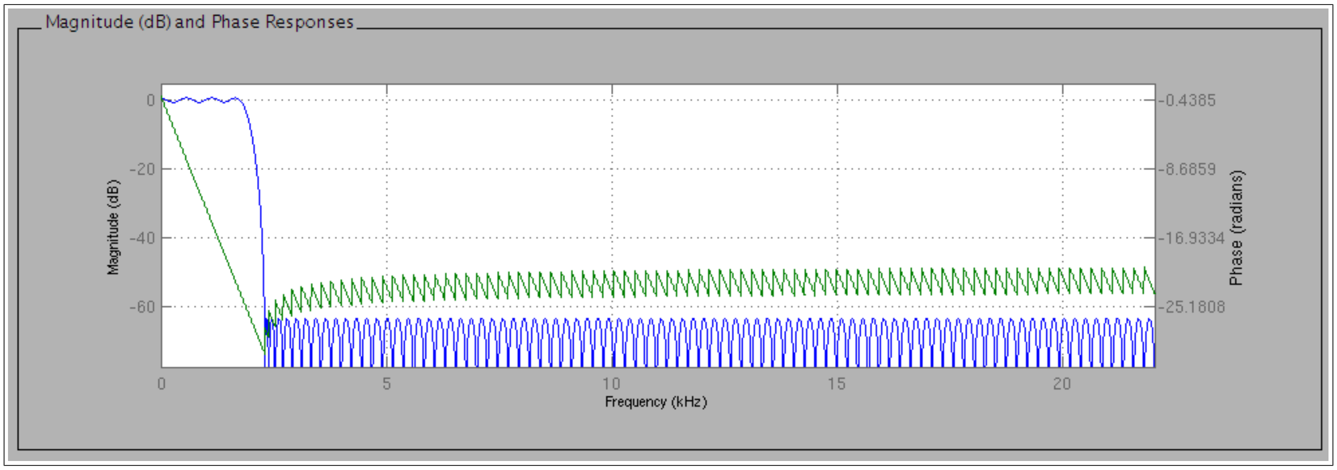
Filter Generation

Four filters need to be created for proper utility of this program: an IIR and FIR filter for each sampling frequency of 44100Hz and 22050Hz. The default usage includes directions for FIR filtering, which is considered the author's chosen method due to its simple implementation and desirable results coupled with the assertion that no lives depend on the real-time performance of the software.

FIR filtering follows this basic algorithm:
- An input buffer is shifted, the newly available space filled with a sample of the input signal.
- An accumulator receives the summation of a multiplication between every element of the input buffer and the FIR filter coefficients.
- The accumulator's final value is then written to the output signal's file.

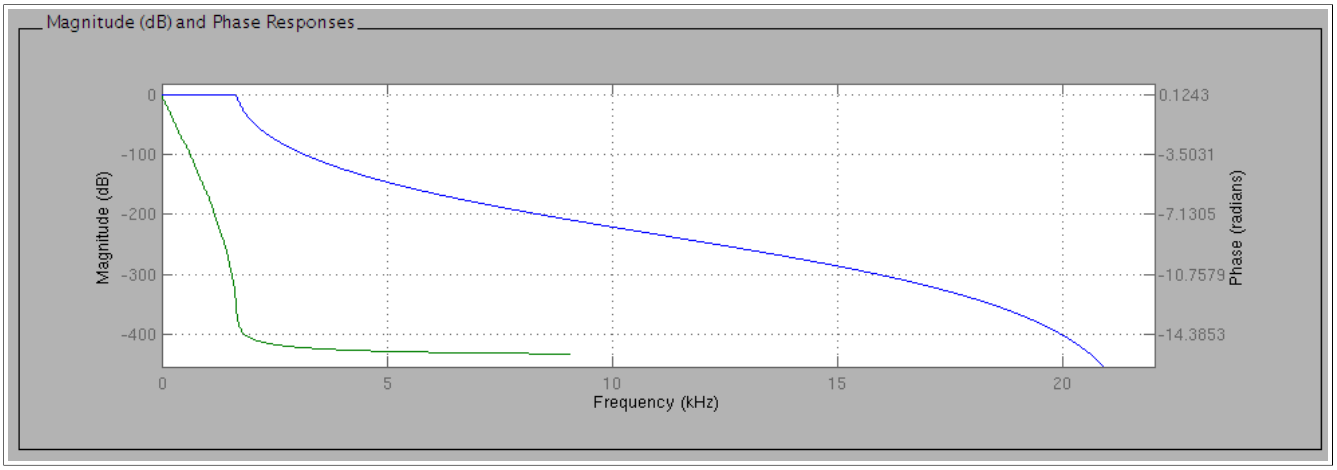
IIR filtering follows this basic algorithm:
- An input buffer is shifted, the newly available space filled with a sample of the input signal.
- An accumulator receives the summation of a multiplication between every element of the input buffer and the IIR Numerator/b filter.
- The output buffer is shifted.
- All but the newly vacant element of the output buffer is multiplied by the IIR Denominator/a filter.
- Swearing and cursing is exchanged as the filter does not work. The filter is regenerated in Matlab.
- The result of all buffer_in multiplications is summed; all buffer_out multiplications is subtracted, and the result is stored as the most recent output sample in the output buffer.
- The output sample is written to the output file.

FIR - 44100



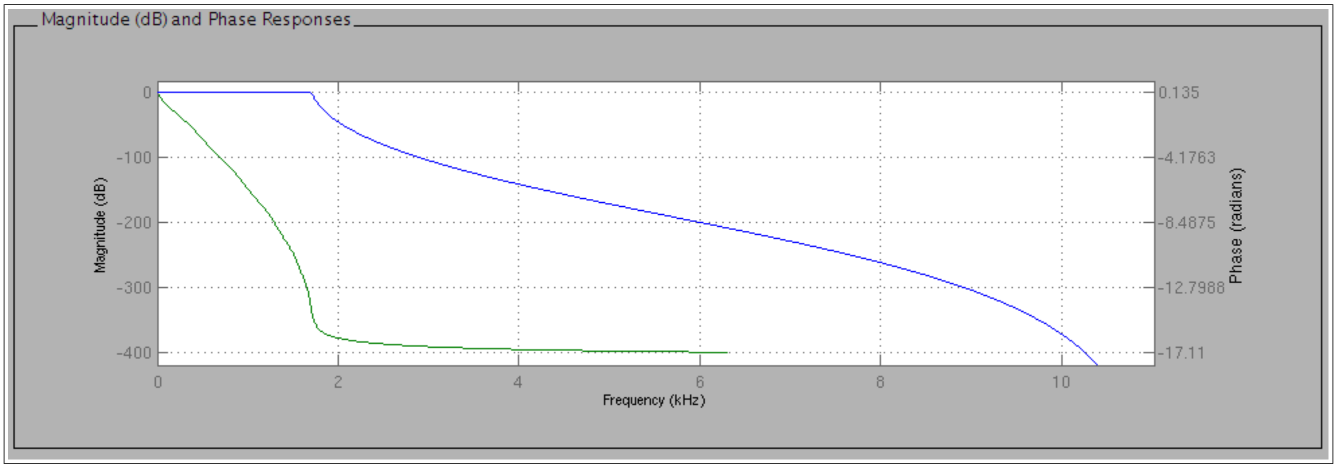

```
-0.0004812070553999,-0.0003125642873676,-0.0003815647819449,-0.0004314764216824,
 -0.00045009906821,-0.0004248185689225,-0.0003438374420322,-0.0001965944253318,
2.444188006655e-05,0.0003224224753277,0.0006948897312632,  0.00113461785515,
 0.001627245729824, 0.002152012823965, 0.002684125888767, 0.003192590331798,
 0.003645147025922,  0.00400798738371, 0.004249818317546, 0.004343195537262,
 0.004267849465504, 0.004012520948328, 0.003576940420594, 0.002972530425765,
 0.002223650574176, 0.001366009236875,0.0004458108771196,-0.0004828631624908,
-0.001361596909983, -0.00213118168972,  -0.0027363361839,-0.003129581962467,
-0.003275790587749,-0.003155431369362,-0.002767197667726,-0.002129417835021,
 -0.00128017864394,-0.0002754075821936,0.0008137305380801, 0.001905646620224,
 0.002913648554844, 0.003752416330844, 0.004344631184098,  0.00462739795557,
  0.00455830050505, 0.004119840714234, 0.003322574796032, 0.002206174551115,
0.0008382361069156,-0.0006892039254139, -0.00226506538324,-0.003766870710916,
-0.005069339919827,-0.006054281997877,-0.006620116545427,-0.006691196694796,
-0.006225520447146,-0.005220394959093,-0.003715558233848,-0.001793221342637,
0.0004252411800937, 0.002785667587129, 0.005110568201147, 0.007210991697518,
 0.008900097203456,  0.01000766713571,  0.01039421122427, 0.009964180755739,
 0.008676597399851, 0.006552734521419, 0.003679844218074,0.0002104994050588,
-0.003642703051721,-0.007616828370078, -0.01141249870968,  -0.0147109794705,
  -0.0171937774273, -0.01856321873454, -0.01856291945052, -0.01699676445099,
  -0.0137448174699,-0.008775397519662,-0.002152031134436, 0.005965155755119,
  0.01532398423426,  0.02559052047024,  0.03636475508121,  0.04720066857617,
   0.0576297635479,    0.067186289082,  0.07543312067557,  0.08198622804968,
   0.0865364751493,  0.08886733860919,  0.08886733860919,   0.0865364751493,
  0.08198622804968,  0.07543312067557,    0.067186289082,   0.0576297635479,
  0.04720066857617,  0.03636475508121,  0.02559052047024,  0.01532398423426,
 0.005965155755119,-0.002152031134436,-0.008775397519662,  -0.0137448174699,
 -0.01699676445099, -0.01856291945052, -0.01856321873454,  -0.0171937774273,
  -0.0147109794705, -0.01141249870968,-0.007616828370078,-0.003642703051721,
0.0002104994050588, 0.003679844218074, 0.006552734521419, 0.008676597399851,
 0.009964180755739,  0.01039421122427,  0.01000766713571, 0.008900097203456,
 0.007210991697518, 0.005110568201147, 0.002785667587129,0.0004252411800937,
-0.001793221342637,-0.003715558233848,-0.005220394959093,-0.006225520447146,
-0.006691196694796,-0.006620116545427,-0.006054281997877,-0.005069339919827,
-0.003766870710916, -0.00226506538324,-0.0006892039254139,0.0008382361069156,
 0.002206174551115, 0.003322574796032, 0.004119840714234,  0.00455830050505,
  0.00462739795557, 0.004344631184098, 0.003752416330844, 0.002913648554844,
 0.001905646620224,0.0008137305380801,-0.0002754075821936, -0.00128017864394,
-0.002129417835021,-0.002767197667726,-0.003155431369362,-0.003275790587749,
-0.003129581962467,  -0.0027363361839, -0.00213118168972,-0.001361596909983,
-0.0004828631624908,0.0004458108771196, 0.001366009236875, 0.002223650574176,
 0.002972530425765, 0.003576940420594, 0.004012520948328, 0.004267849465504,
 0.004343195537262, 0.004249818317546,  0.00400798738371, 0.003645147025922,
 0.003192590331798, 0.002684125888767, 0.002152012823965, 0.001627245729824,
  0.00113461785515,0.0006948897312632,0.0003224224753277,2.444188006655e-05,
-0.0001965944253318,-0.0003438374420322,-0.0004248185689225, -0.00045009906821,
-0.0004314764216824,-0.0003815647819449,-0.0003125642873676,-0.0004812070553999
```

IIR - 44100



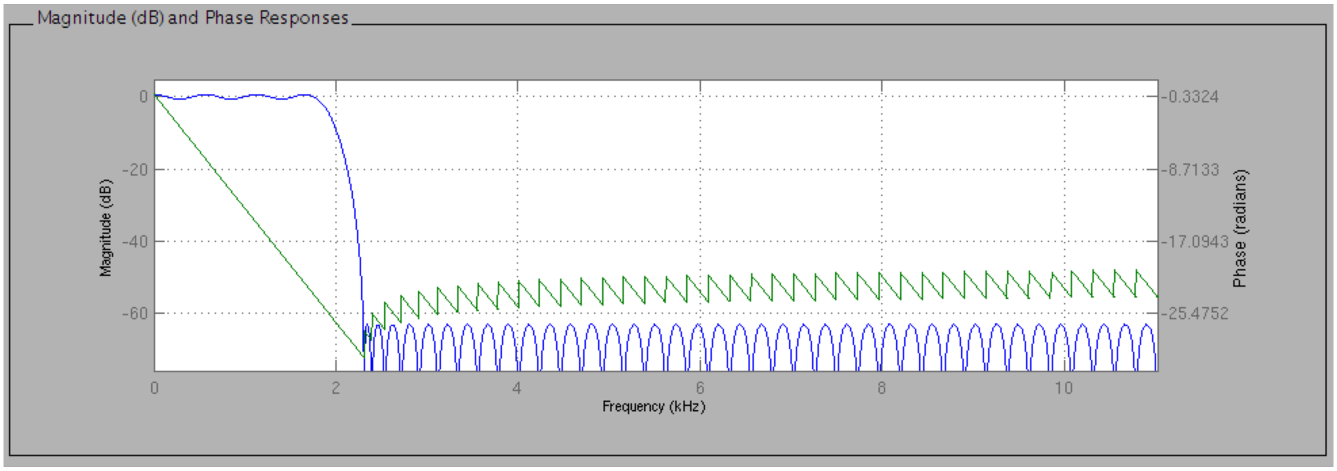

```
Numerator:
   1.75527846804e-12, 1.75527846804e-11,7.898753106178e-11,2.106334161647e-10,
  3.686084782883e-10, 4.42330173946e-10,3.686084782883e-10,2.106334161647e-10,
  7.898753106178e-11, 1.75527846804e-11, 1.75527846804e-12

Denominator:
                   1,    -9.64869768027,    42.02918268338,
-108.837094215,
      185.5445450459,   -217.5849281568,     177.748137936,
-99.87947573383,
      36.94568719833,   -8.123651191181,   0.8062941155919
```

IIR - 22050




```
Numerator:
  2.731758313976e-10,3.004934145373e-09,1.502467072687e-08,
4.50740121806e-08,
   9.01480243612e-08,1.262072341057e-07,1.262072341057e-07,
9.01480243612e-08,
   4.50740121806e-08,1.502467072687e-08,3.004934145373e-
09,2.731758313976e-10

Denominator:
                  1,   -9.932155633877,    45.42862235097,
-126.2638893873,
      236.8802991527,   -314.9017369565,    302.6376457588,
-210.2455345471,
      103.4637393698,   -34.34900807516,    6.924121412451,
-0.6421028852943
```

FIR – 22050




```
-0.0006564713840821,-0.0007018805798646,-0.0007303922289611,-0.0003656212836586,
  0.0005426616749244, 0.002033598815067, 0.003976021659248,  0.00604859335482,
   0.007784330003411, 0.008670209282028, 0.008295522647032, 0.006500215173513,
   0.003481717849447,-0.0001904068306367,-0.003672314978061,-0.006047833518055,
  -0.006590627697197,-0.005011835798849,-0.001612989310247, 0.002722426872299,
   0.006719834457731, 0.009055315514958, 0.008762559697508, 0.005585259340265,
  0.0001555556265177,-0.006084137497514, -0.01122623630974,  -0.0134395164151,
   -0.01156354925514,-0.005577668502544, 0.003219770936555,  0.01241815132468,
    0.01907740511422,  0.02056225409773,  0.01541166177714, 0.003991670774493,
   -0.01128653222981, -0.02629963326467, -0.03606780729486, -0.0359548626500 8,
   -0.02292234882193, 0.003492052389193,  0.04075276821247,  0.08361228055539,
     0.1250785471803,   0.1578587387493,   0.1759426917796,   0.1759426917796,
     0.1578587387493,   0.1250785471803,  0.08361228055539,  0.04075276821247,
   0.003492052389193, -0.02292234882193, -0.03595486265008, -0.03606780729486,
   -0.02629963326467, -0.01128653222981, 0.003991670774493,  0.01541166177714,
    0.02056225409773,  0.01907740511422,  0.01241815132468, 0.003219770936555,
  -0.005577668502544, -0.01156354925514,  -0.0134395164151, -0.01122623630974,
  -0.006084137497514,0.0001555556265177, 0.005585259340265, 0.008762559697508,
   0.009055315514958, 0.006719834457731, 0.002722426872299,-0.001612989310247,
  -0.005011835798849,-0.006590627697197,-0.006047833518055,-0.003672314978061,
  -0.0001904068306367, 0.003481717849447, 0.006500215173513, 0.008295522647032,
   0.008670209282028, 0.007784330003411,  0.00604859335482, 0.003976021659248,
   0.002033598815067,0.0005426616749244,-0.0003656212836586,-0.0007303922289611,
  -0.0007018805798646,-0.0006564713840821
```

File Analysis

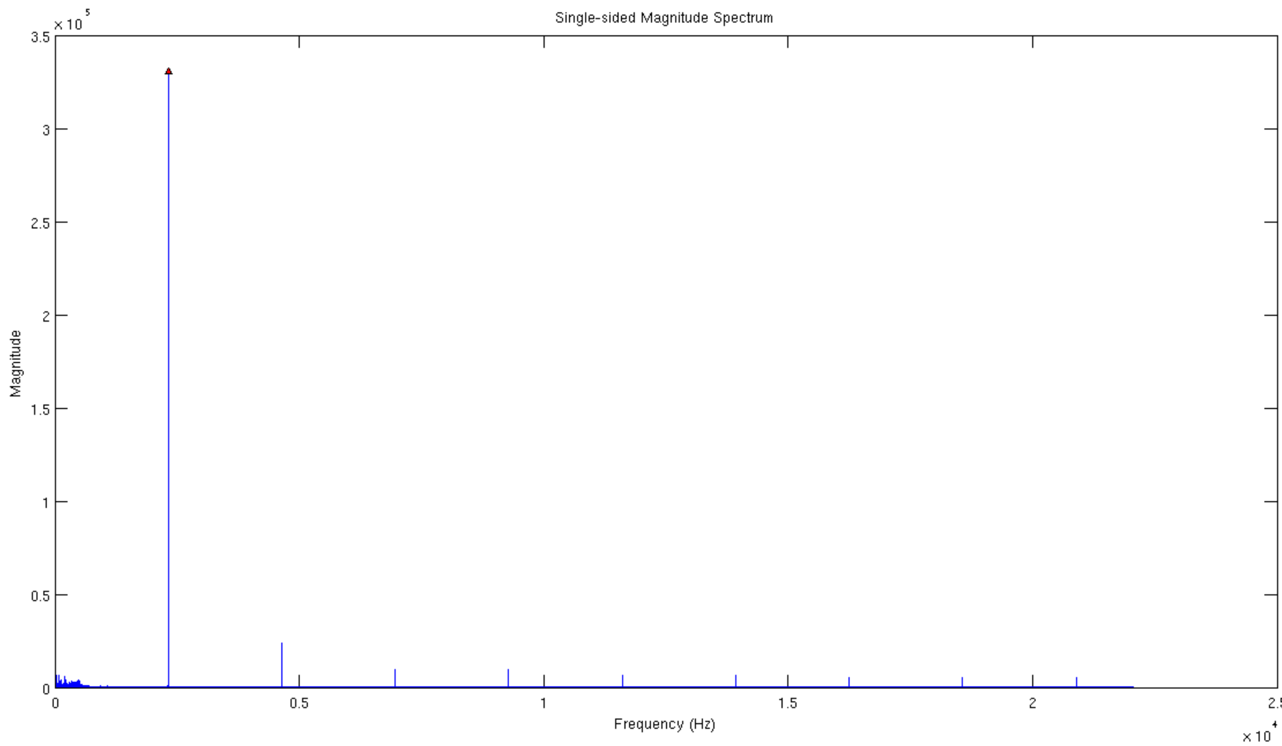
Before I was aware of the ease in which *sptool* could be wielded for spectral analysis, I created a small helper script which plots the spectrum of a signal and indicates the dominant spectral component graphically.

```
% Plot_Signals.m
% use plot_fft function to display available audio
% in frequency domain. Marks dominant spectral component.
% Assembled by Christopher Bero [csb0019@uah.edu]

clear all;
%echo plot_fft on;

[signal_sine, fs_sine]=wavread('/home/berocs/Documents/MATLAB/bero_c_sine.wav');
[signal_orig,fs_orig]=wavread('/home/berocs/Documents/MATLAB/bero_c_orig.wav');
[signal_mod,fs_mod]=wavread('/home/berocs/Documents/MATLAB/bero_c_mod.wav');

fig=figure(1);
plot_fft(signal_sine, fs_sine, fig);

fig=figure(2);
plot_fft(signal_orig, fs_orig, fig);

fig=figure(3);
plot_fft(signal_mod, fs_mod, fig);
```

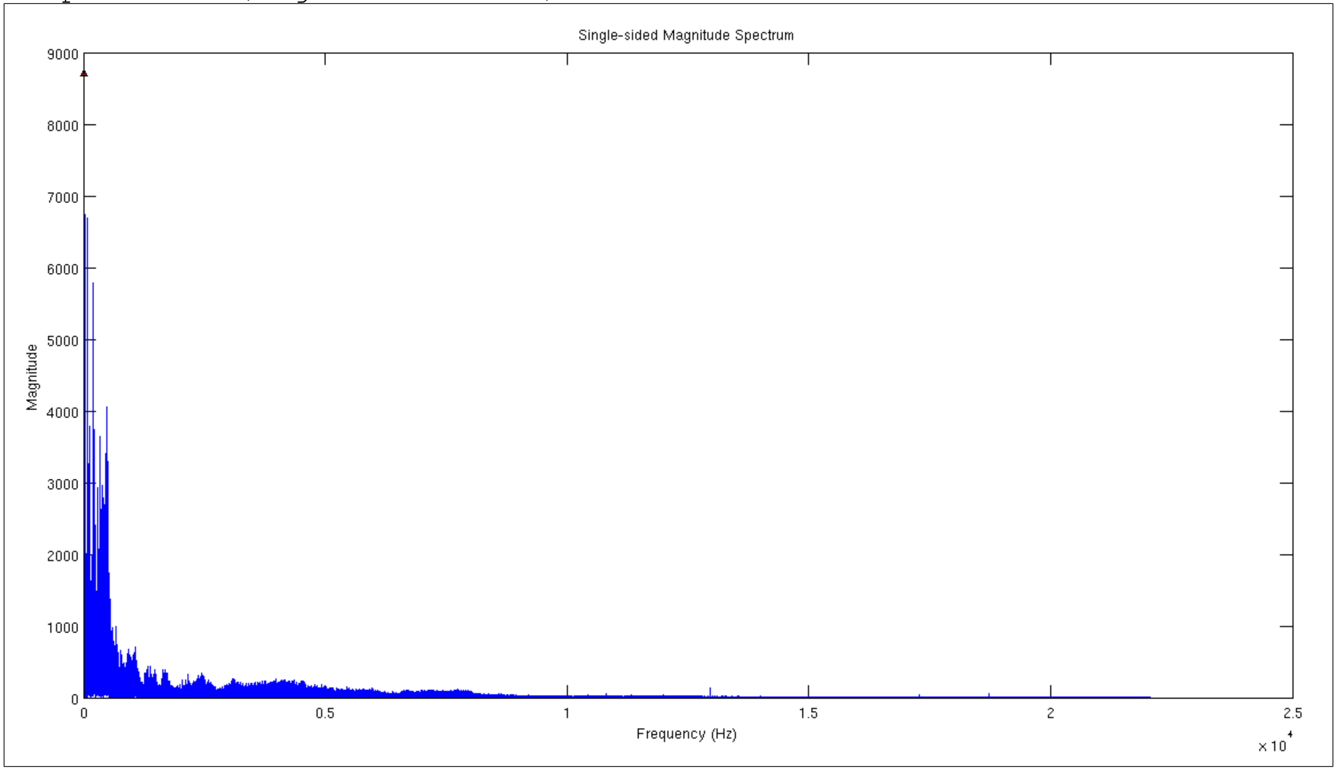
```
function [] = plot_fft( signal, fs, fig )
% Helper function for CPE381 project phase 2
% Assembled by Christopher Bero [csb0019@uah.edu]

% get number of samples
signal_len=length(signal);
N_2 = ceil(signal_len/2);

% Nyquist frequency
fnyquist = fs/2;

% Magnitude of fft gives signal in f domain
signal_mag = abs(fft(signal));

% From http://www.lumerink.com/courses/ece697/docs/Papers/
% "The Fundamentals of FFT-Based Signal Analysis and Measurements.pdf"
%
% Multiply every element but the center point by 2 to get Magnitude for a
% one-sided plot. This is because half of the power is represented as
% negative values, and those are discarded for this plot, being as they are
% typically redundant in real-world signals.
signal_mag=[ signal_mag(1) ( signal_mag(2:length(signal_mag)) * 2.0 ) ];
signal_mag=signal_mag(1:N_2);

bin_vals = 0:signal_len-1;
fax_Hz = bin_vals*fs/signal_len;

% Begin plotting Magnitude in Freq Domain
figure(fig);
plot(fax_Hz(1:N_2), signal_mag);
hold on;
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Single-sided Magnitude Spectrum');

% reimplemented from MATLAB help browser
% takes the largest peak and places a symbol over it
```

```
[peaks,locations] = findpeaks(signal_mag);
x=fax_Hz(1:N_2);
[C,I]=max(peaks);
max_loc=locations(I);
max_peak=peaks(I);
plot(x(max_loc),max_peak+0.05,'k^','markerfacecolor',[1 0 0]);

end
```

Input File (signal with sine wave):



Output File (signal filtered):



Default Usage

To run the program with defaults (FIR filtering):

```
./rtspc_filter -s summary.txt -i mod.wav -o fil.wav
```

Compilation

This program should compile with the following gcc commands:

```
gcc -g -c main.c -o main.o
g++ -o rtspc_filter main.o
```

Real Time Processing

If we are to discuss only the program's algorithm while running on my personal laptop, then it may be said that this program executes in real-time. This is due to the fact that the program can process a wave file with IIR or FIR technique in less time than the wave file would take to play.

Sample Summary File

```
Summary File - CPE 381 - Christopher Bero
=========================================
Filter Technique:                     FIR
Sampling Frequency:               44100Hz
Audio Time:                    45.583832s
Program Execution Time:        14.311573s
Ran in:                        Real Time!
```

Conclusion

I've found this project to be a welcome aspect of CPE381 since much of the semester has been spent attempting to recoup what I feel to be poor prerequisite preparation for the material coupled with radio silence between departments here at UAH. Though needless, I jumped at the chance to pit FIR and IIR filtering against one another because I now have concrete empirical information to take back into discussions on FFT, sampling, and real-time in class. Seeing the delay difference between the two techniques, and the difficulty involved in finding a somewhat stable IIR coefficient list, has already changed the way I consider my music collection and microcontroller projects. Perhaps it will also lend toward a better understanding of the material in class as well.

Resources

Phase 1
[1] http://www.cplusplus.com/doc/tutorial/files/
[2] http://www.cplusplus.com/reference/cmath/sin/
[3] http://www.cprogramming.com/tutorial/cfileio.html
[4] http://www.codingunit.com/c-tutorial-binary-file-io
[5] https://gcc.gnu.org/onlinedocs/gcc/Type-Attributes.html
[6] http://c2.com/cgi/wiki?NonNullTerminatedString
[7] http://stackoverflow.com/questions/1921539/using-boolean-values-in-c
[8] https://www.thinkage.ca/english/gcos/expl/c/lib/atof.html
[9] http://blog.regehr.org/archives/1139
[10] http://www.cplusplus.com/reference/ctime/clock/
[11] http://www.tutorialspoint.com/c_standard_library/limits_h.htm
[12] http://www.topherlee.com/software/pcm-tut-wavformat.html

Phase 2
[13] http://www.eas.uccs.edu/wickert/ece2610/lecture_notes/ece2610_chap8.pdf
[14] http://askubuntu.com/questions/340835/simple-example-of-gnuplot
[15] http://c.learncodethehardway.org/book/ex16.html
[16] http://www.tutorialspoint.com/cprogramming/c_passing_pointers_to_functions.htm
[17] http://www.codingunit.com/c-reference-stdlib-h-function-calloc
[18] http://www.cs.hmc.edu/~vrable/gnuplot/using-gnuplot.html