EC380 Mini Project 5 – Note Detection

Work alone for this mini project. You are free to discuss ideas with others. This MATLAB project is part of a sequence of mini projects leading up to a simple communication system.

Background:

For this mini project you will write a simple MATLAB function that listens to a tone and identifies what note it is. The files **`noteN.wav`** (where *N* is a number from 1 to 8) in the `G:ee\ece380\Mini Projects\04 NoteDetection` directory have one note each. Your job is to decide what note it is. The lowest note is a G at about 750 Hz (my ocarina is a bit out of tune, so the notes aren't right on standard pitch). The highest note is one octave above at 1500 Hz.

Approach

You are to design 13 filters. Each filter will pass just one note. The first filter will be tuned to pass 750 Hz, the next filter will pass $750 * 2^{1/12}$ Hz, and so on. The 13th filter will pass 1500 Hz. See Lab 1 if you don't understand the $2^{1/12}$. You will then pass the unknown signal through each of the filters. i.e. The input signal is the input to each filter. You then measure the power coming out of each filter and the filter with the largest power corresponds to the note that was played.

Sample Code

The following MATLAB code is a good starting point:

```
function maxNote = toneDetect(xx, fs, DEBUG)
%toneDetect Returns a number indicating what note was played.
%   toneDetect(xx, fs) xx in the input sound, fs is the sample rate.
%
%   toneDetect(xx, fs, 1)   Like above, but turns DEBUGing on.
%

%   Copyright 2003 Mark A. Yoder, Rose-Hulman
%

if (nargin < 3) % Turn off debugging if last argument not present.
    DEBUG = 0;
end

all = xx';
max = 0;
maxNote = -1;

for note = 0:12;
        freq = 750 * 2.^(note/12);
        what = Insert code to compute $\hat{\omega}$ (what)  based on freq and the sampling rate.
        aa = Insert code to make a bandpass filter with center at $\hat{\omega}$ (what).
        bb = 1;
        yy = filter(bb, aa, xx);

        pow = Insert code to find the total power in the signal yy


        if pow > max            % Is this the loudest note?
                max = pow;          % Yes, remember it.
                maxNote = note;
        end
        all = [all yy'];        % Save output for debugging.
end

if(DEBUG)       % If debuging in on, play outputs of all the filters.
    soundsc(all, fs)
end
```

Test your code with something like:

```
     [xx, fs] = wavread('note1');
     noteNum = toneDetect(xx,fs);
```

Filter the Frequencies

You should know enough to pick the **a** coefficients for your filter. Be sure the **a** coefficients are real so your output will be real. Also be sure your poles aren't on the unit circle; it could become unstable. Think in terms of poles and zeros and where you would like to place them. Don't use any more poles/zeros than needed. Hint: MATLAB's `poly` command may come in handy. In your memo, note the pole/zero locations you are using.

Due Date:

This assignment is due Tuesday 27-Jan-2004 at the start of class.
What is due:

1. One page memo describing what you did. Have a table listing the wav file name (e.g note1.wav), and the note you think is in the file for each wav file. Highlight any **extras** you did.
2. Your MATLAB code.

Print (don't' email) your memo and MATLAB code and hand it in class.