ELEC484/532 Audio Signal Processing

Assignment 2. Summer 2007 (15 pts)

The main goal of this assignment is to motivate you to experiment with various digital audio effects for creative purposes and to gain insight about some of the ideas we dicussed in class.

Read the questions carefully and do not hesitate to contact me via email if you have any questions or need some clarifications.

The assignment is due **July 12th** in class. For each question you should describe what you did, what programming environment you used, show screenshots/figures/plots and provide all the relevant code. For the audio files you create the easiest would be if you could put them on a website and provide me with a link. If you don't have access to a website then you can send it by email, give me a CD or transfer them using a usb key to my laptop. You can use any text processing environment you want but what you hand me must be in **pdf** format. Any programming environment, tool is also acceptable unless I explicity specify which tool you use. Any choices you make should be documented in your report.

Enjoy, George Tzanetakis

Question 1 (5 + (1 Extra Credit) pt)

A wah-way effect is created by a bandpass filter with variable center frequency and small bandwidth. The center frequency is controlled by a low frequency oscillator (LFO) with frequency around 1-2 Hz, or can be controlled by an external control input device such as a mouse or slider. The center frequency varies from near 0 to a frequency less than half the sampling rate. The output of the bandpass filter is added to the direct (input) signal.

In this assignment you will study a 2-pole 2-zero bandpass filter in detail, and then use it to add a wah-wah effect to an audio file.

You are asked to design a bandpass filter, test the filter characteristics, and document your results. Realistic specifications for the bandpass filter to be used to create a wah-wah effect using a biquad structure (two poles and two zeros) are:

```
sampling rate fs = 44100 Hz
center frequency fc = 44100 / 64 = 689 Hz
```

```
3dB bandwidth B = 100 Hz
```

Alternate specifications which result in simpler algebra are:

```
sampling rate fs = 8000 Hz
center frequency f1 = 2000 Hz
3dB bandwidth B = 100 Hz
```

- Design the filter and specify the transfer function H(z). Scale the filter for unity gain at fc (bandpass)

- An approximate formula for the 3 dB bandwidth B of the filter as a function of the magnitude $|a|$ of the poles (distance of the pole from origin) is $\delta\omega = 2\frac{|1-a|}{\sqrt(a)}$ where $\delta\omega$ is the normalized bandwidth B in radians. This shows that the distance of the pole from the unit circle $|1-a|$ controls the bandwidth. Find a numerical value for $|a|$ such that B = 100Hz

- Plot the frequency response (both amplitude and phase) of H(z). Check that the power response id down 3 dB at $fc+B/2$, as it should be with the correct value of $|a|$. What is the phase shift at fc and $fc+B/2$?

- Demonstrate that the filter works correctly by computer simulation as follows: Evaluate the filter output y(n) with sinusoidal input x(n) by using the difference equations. Use sampled cosine waves at the center frequency fc, repeat again at the 3dB down frequency $fc+B/2$. Plot the input and output on the same graph. Verity from the graph that both the amplitude and the phase of the y(n) cosine wave output relative to the input x(n) are as you should expect from the frequency response.

- Use your bandpass filter to a dd a wah-wah effect to an audio file consisting of white noise, and compare the result using the wah-wah effect build into Audacity

- **1 point extra credit** Implement the above filter and wah-wah effect in *Marsyas*. The documentation covers topics such as how to compile, install and use *Marsyas*. You can look at the files BiQuad.cpp, OnePole.cpp, Filter.cpp for ideas about how to implement the filter and at the examples subdirectory for code that uses MarSystems. Use the marsyas-users mailing list at sourceforge for any questions.

Question 10 (10pts)

This qustion explores spatialization of sounds for a museum exhibit for World War II. As the sound designer you are given a large set of monophonic sound recordings from that era including airplanes, guns, people talking etc. Each visitor in the exhibit room will be wearing a set of wireless headphones equiped with sensors that track the position of the visitors in the room. It is not desired to use loudspeakers as some of the visitors might choose to be in silence but as a designer you want to achieve a sense of spatial immersion. After 'some thinking you decide to try out the following idea: Render each sound source using the room-within-a-room model assuming a square room with 4 "virtual" loudspeakers in each corner and calculating direct and first-order reflections as image sources. After calculating the signals that will be fed in each "virtual" loudspeaker you will need to calculate the signals to be fed to the lef and right headphones for each visitor. The idea will be to consider each "virtual" loudspeaker as a sound source to be rendered using headphones in a visitor-specific way. For the headphone rendering you decide to use only panorama, precedence effect and distance rendering with a single reflection (no HRTFs).

For every 0.25 seconds the input to your system will be the positions of M original sound sources in the "outside" room and the positions of N listeners in the "inside" room. A helper function will convert the M original sound sources to M' sound sources (that include the image sources for the first order reflections).

- Write in pseudo-code the helper function that given the original source positions returns the positions of the image sources for the first-order reflections. You can assume equal angles of incidence and reflection. Use a diagram to explain the calculation.

- Make a block diagram that shows the process and basic signal processing building blocks that are required in detail.

- For each original source there will be 4 image sources corresponding to the reflections on the walls of the outer room. Describe in detail how a single delay line could be used to render all of these 5 (1 real + 4 image) sources. How could you calculate the size of the delay line ?

- Consider a "outer" room of 100 meters on a side and an "inner" room of 8 meters on a side. A sound source is located at (15, 23). Calculate the delays and attenuations (inverse square law) for rendering this source in each corner "virtual" loudspeaker of the "inner" room. Show your calculations geometrically in a diagram. Calculate the left-right headphone attenuations and delays for a listener at the center of the inner room as well as the middle of one of the sides.

- A software company is going to implement the system. It has excellent programmers but they don't know much about audio signal processing. You are required to provide a detailed specification of the entire system that they will use to implement it. Use any type of information you think is appropriate equations, block-diagrams, pseudo-code, actual code, english to convey all the necssary information (feel free to refer to reuse parts of the answers to the previous questions)