CPE 367 – Experiment 4a – v3b
Delay Effects: Positional Audio and Echo – Dr F DePiero (48 Points)

Overview and Motivation

The signal processing in this experiment involves digital filters with particular delay characteristics. Often when creating digital filters, the magnitude characteristics of the frequency response are the most critical. (For example, a high pass or low pass filter). In this experiment we explore the delay characteristics instead. This experiment will also provide an introduction to IIR filters.

3-D positional audio is an emerging technology. In the near-term we can expect that media creators will be able to create a script for sound clips that is similar to actors' scripts and action choreography. The positional audio data defines **when** and **where** the sound clip should occur (in a home or theater, for example). The positional effect is accomplished by controlling the amount of delay for a given sound clip, as it is added into the signal for each speaker of a surround sound system. We will implement the positional audio by creating a stereo output signal, using an input mono signal. You'll get best results by listening with headphones.

In addition to positional audio, we will also implement echo effects. An FIR echo can be created using a difference equation in the form

$$y[n] = b_0 \ x[n] + b_d \ x[n-d] \qquad \text{(Eq. 1)}$$

This operation mixes together a delayed sample with the current sample to yield a single echo. Meaning that a sound such as a handclap would be repeated exactly once. Alternatively, we can use an IIR approach

$$y[n] = b_0 \ x[n] + a_d \ y[n-d] \qquad \text{(Eq. 2)}$$

In this approach the sound of a handclap would repeat theoretically forever. In practice, once the signal's absolute value drops below the smallest representable (integer) value, it will round to zero. For the 16-bit WAV files the lower limit is 1 (out of 2^16).

Learning Objectives

- Find the equivalent difference equation for an IIR system
- Implement both FIR and IIR filters in Python
- Determine the impulse response for a simple IIR system

Prerequisite Learning Objectives

- Implement a digital filter in Python
- Python programming with WAV file I/O and some experience with MatLab

Procedures, Questions and Deliverables

1) Positional Audio

A simple 1D version of positional audio can be implemented by creating a stereo signal. Equations 3a and 3b define outputs for the right and left channels of a stereo WAV file.

$$y_R[n] = x[n - d_R] \qquad \text{(Eq. 3a)}$$
$$y_L[n] = x[n - d_L] \qquad \text{(Eq. 3b)}$$

To create the illusion of a position for the sound source, delays d_R and d_L are chosen based on the speed of sound propagation. If a sound reaches each ear simultaneously then the source appears to be directly ahead of the listener. If the sound reaches the left ear first then the sound appears to be to the left, and similarly for the right. For a sound source that seems to be fully to the left of the listener, then the relative time difference should be approximately equal to the amount of time for sound to propagate through a distance equal to the width of a person's head. Assume the distance between the ears is ~8 inches and that the speed of sound is 343 m/s. Use with a sample rate of 16 kHz.

- **1a) How long does it take sound to propagate 8 inches? (in samples) (2)**

Write a Python program that uses the joy.wav file for input and generates a stereo WAV file that implements delays, as described by Equations 3a & 3b. Your program should gradually shift the apparent position from fully one side to the other in approximately 4 seconds. (This is twice as long as the example provided). Use your circular buffer class.

- **1b) Documentation: Paste the sections of your Python code that implements the difference equation and the gradual shift (6)**
- **1c) Documentation: Upload your output stereo WAV file (6)**

The file **cpe367_delay.py** is an example of a main Python program that you can modify. It includes the cpe367_way.py class for WAV file I/O. It also includes a do-nothing version of a my_fifo class, which you can replace with your own working version. The main program begins with a short test section to check the operation of your circular buffer. Feel free to comment this out once you are confident that it is working properly. The main program may be run at the command line. It does file I/O using hardcoded file names (modify as needed). A couple highlights of the main program follow. Open a stereo WAV file using

```
# configure the output WAV file
num_ch = 2
samples_8_16_bits = 16
sample_rate_hz = 16000
wav_out.set_wav_out_configuration(num_ch, samples _8_16_bits,sample_rate_hz)
```

Write two integers to the WAV file using

```
# output a stereo pair of samples to a WAV file
wav_out.write_wav_stereo(yout_left,yout_right)
```

2) FIR Echo

Initially let the delay, d, be 1 sample in the FIR-style echo in Eq. 1. Use a 50%/50% mix between the input signal and the echo, in other words $b_0 = b_d = 0.5$.

- **2a) What is the impulse response, h[n], for this system with d = 1? (2)**

Implement the FIR-style echo defined in Eq. 1, in Python. Use a 0.125 Second delay and a 50%/50% mix between the input signal and the echo. Generate a mono signal and use the joy.wav signal for the input. Use your circular buffer class.

- **2b) How long does the echo theoretically persist? (in samples) (2)**
- **2c) Documentation: Paste the sections of your Python code that implement the difference equation using your circular buffer code (6)**

3) IIR Echo

Initially let the delay, d, be 1 sample for the IIR-style echo in Eq. 2. Use a 50%/50% mix between the input signal and the echo, in other words $b_0 = a_d = 0.5$

- **3a) What are the first 4 values in the sequence of the impulse response? (2)**
- **3b) What is the impulse response, h[n], for this system with d = 1? Define a function in the general form of a^n. Specify parameters and the limits on n (4)**

Implement the IIR-style echo defined in Eq. 2, in Python. Use a 0.125 Second delay and a 50%/50% mix between the input signal and the echo. Generate a mono signal and use the joy.wav signal as the input. Use your circular buffer class.

- **3c) How long does the echo theoretically persist? (in samples) (2)**
- **3d) Documentation: Paste the sections of your Python code that implement the difference equation using the circular buffer code (6)**

4) Find an Equivalent Difference Equation Given a Block Diagram

In Figure 1 below, assume the delay of tau seconds is equivalent to a delay of d samples.

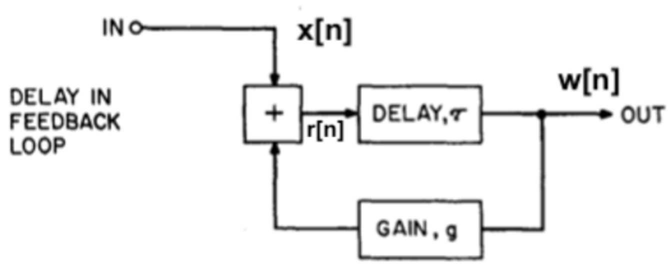

Figure 1. IIR comb filter with delay and feedback (from Schroeder, g = 0.7)

- **4a) If the input x[n] is an impulse, when does the output w[n] have its first nonzero value (in samples)? (2)**
- **4b) Give an expression for r[n] in terms of x[] and w[] (4)**
- **4c) Give an expression for w[n] in terms of x[] and w[] (4)**