EE 367 Lab 4

Delay Effects: Positional Audio and Echo

Report by:

Ethan Vosburg (evosburg@calpoly.edu)
Isaac Lake (ilake@calpoly.edu)

February 20, 2024

Table of Contents

1 Positional Audio

In part 1 we will be simulating an imaginary directional location for the sound by delaying the notes in one ear slightly compared to the other tricking your brain. If your right ear hears slightly before your left then you will perceive the sound source as somewhere to the right of you.

1. How long does it take sound to propagate 8 inches? (in samples)

   Calculating the propagation of sound can be done fairly simply using only the distance 8 inches, the speed of sound $343m/s$, and our current sample rate $16000samples/s$. First, we must convert inches into meters,

   $$8in = 0.2032m$$

   then we divide our distance by the speed of sound to get the time it will take to cross that distance

   $$\frac{0.2032m}{343m/s} = 0.000592s$$

   finally we can multiply the time by our sample rate to see how many samples we will need to delay by

   $$0.592ms * 16000samples/s = 9.48samples$$

2. Python code that implements the difference equation and the gradual shift, with comments:

Listing 1: Positional Audio FIFO Setup

```
##########################################################################
# students - allocate your fifo, with an appropriate length (M)
M = 10
right_fifo = my_fifo(M)
left_fifo = my_fifo(M)

# Delay on ears
# L: 9 8 7 6 5 4 3 2 1 0
 # R: 0 1 2 3 4 5 6 7 8 9

# sample counter for stereo effect
sample_count = 0
delay = 0

##########################################################################
```

Listing 2: Positional Audio Filter

```
    ###########################################################################
    # students - there is work to be done here!

    # update history with most recent input
    right_fifo.update(xin)
    left_fifo.update(xin)
    sample_count += 1
    # evaluate your difference equation to yield the desired effect!
    #  this example just copies the mono input into the left and right channel
    if (sample_count % 6400 == 0):
       delay += 1
       if delay > 9:
          delay = 0
       print(delay)

    yout_right = right_fifo.get(9 - delay)
    yout_left = left_fifo.get(delay)

    # students - well done!
    ###########################################################################
```

2 FIR Echo

In part 2 of this lab we will implement an FIR style difference equation

1. What is the impulse response, h[n], for this system with d = 1?

   Given $y[n] = 0.5x[n] + 0.5x[n-1]$ we can easily grab h[n] as [0.0, 0.5, 0.5, 0.0]
2. How long does the echo theoretically persist? (in samples)

   It should theoretically persist for 0.125 seconds or 2000 samples since we have it delayed by that much and it only lasts a handful of samples
3. Paste the sections of your Python code that implement the difference equation using your circular buffer code

Listing 3: FIR Echo FIFO Setup

```
###########################################################################
# students - allocate your fifo, with an appropriate length (M)

# Creating a 0.125 second echo
M = 2000
right_fifo = my_fifo(M)
left_fifo = my_fifo(M)

# students - well done!
###########################################################################
```

Listing 4: FIR Echo Filter

```
    ##########################################################################
    # students - there is work to be done here!

    # update history with most recent input
    right_fifo.update(xin)
    left_fifo.update(xin)

    # Writing out echo
    yout_right = 0.5 * xin + 0.5 * right_fifo.get(1999)
    yout_left = 0.5 * xin + 0.5 * left_fifo.get(1999)

    # students - well done!
    ##########################################################################
```

3 IIR Echo

In part 3 of this lab we will implement an IIR echo that should theoretically persist indefinitely using the equation $y[n] = 0.5x[n] + 0.5y[n-1]$

1. What are the first 4 values in the sequence of the impulse response?

   This is a very simple sequence to grab, straight from the equation assuming we are starting from a relaxed state. The sequence would be $[0.0, \underline{0.5}, 0.25, 0.125, 0.0625]$

2. 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.

   the impulse response in a more general form would be $(0.5)^n$ where n is the current sample since the original, in other words, a positive integer larger than 0

3. How long does the echo theoretically persist? (in samples)

   The echo will persist indefinitely since it will continue to be non-zero forever as an infinite response system.

4. Paste the sections of your Python code that implement the difference equation using the circular buffer code.

Listing 5: IIR Echo FIFO Setup

```
##########################################################################
# students - allocate your fifo, with an appropriate length (M)

# Creating a 0.125 second echo
M = 2000
right_fifo = my_fifo(M)
left_fifo = my_fifo(M)

# students - well done!
##########################################################################
```

Listing 6: IIR Echo Filter

```
        ###########################################################################
        # students - there is work to be done here!

        # update history with most recent input
        right_fifo.update(xin)
        left_fifo.update(xin)

        # Writing out echo
        yout_right = 0.5 * xin + 0.5 * right_fifo.get(1999)
        yout_left = 0.5 * xin + 0.5 * left_fifo.get(1999)

        # students - well done!
        ###########################################################################
```

4 Find an Equivalent Difference Equation Given a Block Diagram

In part 4 of the lab we will be interpreting a block diagram into a signal response.

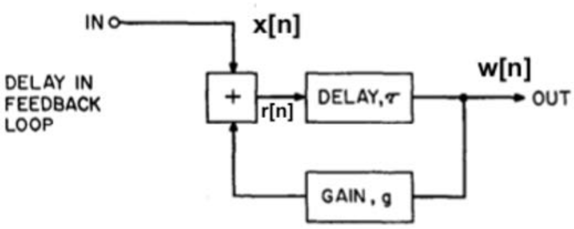

Figure 1: IIR Comb Filter with delay and feedback (Schroeder, g = 0.7)

1. If the input x[n] is an impulse, when does the output w[n] have its first nonzero value (in samples)?

   w[n] should have its very first nonzero value after τ samples, considering that is how long it will take the x[n] input to initially reach it
2. Give an expression for r[n] in terms of x[] and w[]

   $r[n] = x[n] + 0.7w[n-\tau]$
3. Give an expression for w[n] in terms of x[] and w[]

   $w[n] = x[n-\tau] + 0.7w[n-2\tau]$