ECE 280L Fall 2024

Laboratory 3:
Digital Audio Effects

Contents

1 Introduction

In this assignment, you will explore simple digital audio effects. The algorithms (or signal manipulations) used to create these effects will be implemented using the audio system toolbox on Simulink® so that your input signals—be they speech or music—can be processed in real-time. The effects you will explore include a simple echo generation, reverberation, and flanging.

2 Objectives

The objectives of this project are to:

- Become familiar with the software (specifically Simulink) that will be used in this laboratory
- Implement a simple audio I/O system using Simulink
- Implement simple audio effect algorithms in real-time using the audio system toolbox
- Manipulate system parameters to explore the effect on the audio output

3 Background

Audio effects such as echo, reverberation, and flange can be easily implemented using the audio system toolbox. Each audio effect is the result of some manipulation of the input signal, $x[n]$, resulting in some output signal, $y[n]$. The ***signal processing algorithm*** used to transform the original input signal, $x[n]$, into the output signal containing the audio effect, $y[n]$, can be represented mathematically using a ***difference equation***.

The first audio effect you will implement is a ***single echo***. Echoes are simply delayed, attenuated versions of the original signal. The single echo can be realized using the following difference equation:

$$y[n] = x[n] + a\,x[n-D], \qquad |a| < 1 \tag{1}$$

where $x[n]$ is the input signal, $y[n]$ is the output signal, a is a scale factor, and D is the delay in number of samples.

The second audio effect you will investigate is ***reverberation***. When listening to sound in an enclosed space (e.g., a concert hall or a classroom), the signal reaching your ear consists of several components including the original signal and echoes of that signal caused by the reflection of the original signal off surfaces such as the walls, floor, and other objects in the room. If an infinite number of echoes are added up, the effect imitates the reverberating nature of a room. A very simple (but not necessarily natural-sounding) reverberator can be thought of as an infinite sum of echoes added to the original signal. This form of reverberation can be implemented by the following difference equation:

$$y[n] = x[n] + a\,y[n-D]$$

where $x[n]$ is the input signal, $y[n]$ is the output signal, a is a scale factor, and D is the constant delay in number of samples. Notice that the output contains not only the original signal, but also a delayed version of the output itself (which is where the infinite number of echoes comes from). To achieve this, the signal processing algorithm must include ***feedback***.

A more realistic reverberator, the *allpass reverberator*, is represented by the following difference equation:

$$y[n] = a\,x[n] + x[n-D] - a\,y[n-D]$$

You will first design and implement the simple reverberator, then you will compare the output to that produced by the allpass reverberator (which will be given to you).

The third audio effect you will study is ***flanging***. Flanging was originally created using a two- step process. First, a piece of music was fed to two tape recorders. As we learned in class, if a signal is recorded and then played back (e.g., using a tape recorder), there is a delay between the input and the output. If the output of one recorder, $y_1[n]$, was delayed by D_1 samples and the output of the second recorder, $y_2[n]$, was delayed by $D_1 + D_2$ samples, then the combined output would be a single echo, as before. The equation describing this system would be:

$$y[n] = y_1[n] + y_2[n] = y_1[n] + y_1[n - D_2] \tag{2}$$

where $y_1[n] = x[n - D_1]$ and $y_2[n] = x[n - D_1 - D_2] = y_1[n - D_2]$. Notice that, if a scale factor is added to the second term in Equation 2, and if the first delay D_1 were set to 0, the system has a form identical to that of the single echo system in Equation 1.

To create the flanging effect, a second step must be taken. Instead of a constant difference in delay between the two outputs, $\Delta n = D_2$, a variable delay is used, $\Delta n = d(n)$. One way of creating this delay is to slow down one of the tape recorders by placing your thumb on the flange of the feed reel – thus the term "flanging." Analytically, the flanging effect can be represented by incorporating a time varying delay into the single echo system:

$$y[n] = x[n] + a\,x[n - d(n)],$$

where $x[n]$ is the input signal, $y[n]$ is the output signal, a is a scale factor, and $d(n)$ is the time-varying delay in number of samples. To create the flanging effect, the delay is varied using a low-frequency sinusoidal oscillator:

$$d(n) = D + \frac{D}{2}\left(1 - \sin\left(\frac{2\pi f_d n}{f_s}\right)\right)$$

where D is a fixed delay, f_d is the flanging frequency, and f_s is the sampling rate.

4 Equipment Needed

- PC with MATLAB and Simulink
- Microphone and speakers
- Speech and music files

5 Instructions

This project will require you to use both equipment and software tools that you are already familiar with as well as several new hardware and software tools. A brief description of the tools you will be using is in Appendix A of this handout. If you have questions regarding the operation of any equipment, familiar or not, PLEASE ASK before attempting the trial and error method.

5.1 Exercise #1: Introduction to Simulink and the Audio System Toolbox

Your first task is to become familiar with the software tools that you will be using over the course of the next few weeks. To do this, you will design, build, and implement a basic audio processor using Simulink with the audio system toolbox.

1. Make sure that your external headphones/microphones are plugged into your computer before beginning this lab.

2. Open MATLAB. Change to the folder in which you plan to save files for this lab. Start Simulink by typing `simulink` at the MATLAB prompt. Once Simulink opens, create a new Blank Model.

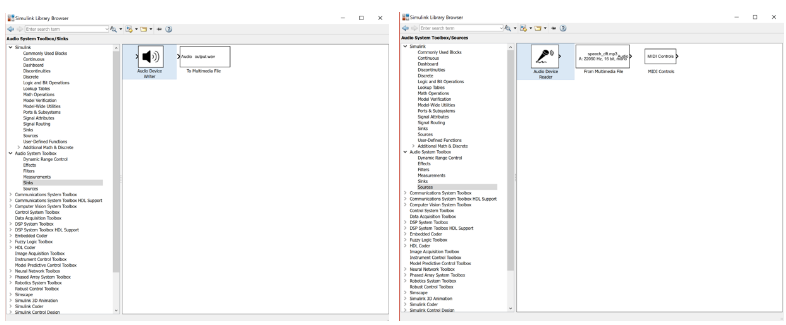

Figure 1. Sinks and Sources for Audio System Toolbox.

3. Navigate to the **Audio System Toolbox** in the library browser and place the Audio Device Reader and the Audio Device Writer, found in **Sinks (output)** and **Sources (inputs)** (see Figure 1), on the model diagram window.

4. Once they are on your model, connect the two, as shown in Figure 2.

Figure 2. Audio Device Reader and Writer Connections.

5. Double-clicking a block will display the block parameters. Some parameters are user-definable, while others are not.

    - The **Audio Device Reader** takes an incoming analog signal and converts it to a digital signal. The **Audio Device Reader Block Parameters** (left side of Figure 3) allow you to choose the driver and specify the Sample rate. We will normally be using default device and the DirectSound driver. The *Samples per frame* parameter tells the block board how large of a chunk of data to process at one time. This number should be a power of two (for more efficient processing) and should not be too large (as there can be memory issues).
    - After processing, the **Audio Device Writer** takes a digital signal from the block and converts it into an analog signal. The output is sent to both ouput jacks, allowing them to be used/monitored simultaneously. The **Audio Device Writer** Block Parameters (right side of Figure 3) must be set to inherit the sample rate, or else an error will occur in compiling.
    - For most applications in this laboratory, you will use a sampling rate of either 8kHz (speech applications) or 44.1kHz (music applications). The parameter should be the same for both reader and writer.

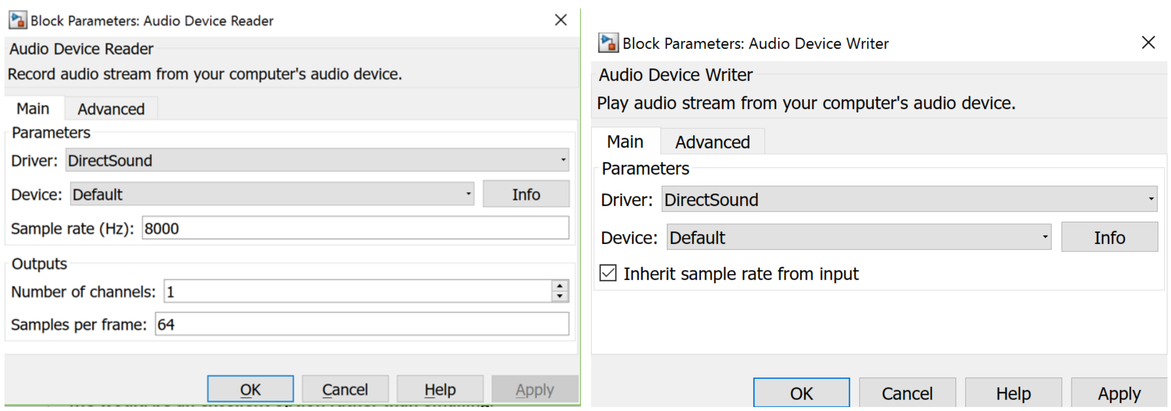

Figure 3. Audio Device Reader and Writer Block Parameters.

6. The Simulink program is now complete for simple audio input/output. You should now save your model again (and do so periodically so that you do not lose your work).

7. Under MODELING -> Model Settings on the Solver tab, be sure to set the stop time to inf to prevent the real time processing from stopping

8. You are now ready to test your system. Make sure the hardware is connected properly by:

    (a) Connecting the microphone to the MIC IN jack on the computer.
    (b) Connecting the headphones to the HEADPHONE jack on the computer.

9. The model should now be complete and ready to run. Hit the play button to run the model.

    Checkpoint (1/7): Show your TA your Simulink model and its output.

10. You will now experiment using different input sources and output devices.
    (a) Function Generator input (if you are *in person* and have access to lab equipment)
        i. Set the function generator parameters to produce a 400 Hz, 0.5-V sinusoid.
        ii. Verify your signal by connecting the function generator to the oscilloscope.
        iii. Now, connect the output of the function generator to the MIC IN on the computer using the BNC/audio plug cable.
        iv. Connect the HEADPHONE output of the computer to the oscilloscope and verify that the output is a 400 Hz sinusoid.
        v. Connect the headphones to the appropriate jack on the computer and listen to your signal.
    (b) .wav file input
        i. Switch the Audio Device Reader block with the From Multimedia File block and set the parameters accordingly to point to Tones.wav
        ii. Connect your headphones to the HEADPHONE jack of the computer.
        iii. Play the file Tones.wav (8 kHz sampling rate). This file consists of a sequence of tones (300 Hz, 400 Hz, 500 Hz, 300 Hz).
        iv. Verify that system works by listening to the computer output via headphones.

    Checkpoint (2/7): Share the output of your system with your TA.

5.2 Exercise #2: Single Echo Effect

For this experiment, you will explore the echo effect. As explained in the Background section, an echo is simply a delayed, attenuated version of the original signal. The difference equation describing a system with a single echo was given previously as $y[n] = x[n] + a\,x[n-D]$. You will now design a Simulink block diagram that implements this system and explore how changing the system parameters a and D (gain and delay) affects the output, using speech and/or music signals as your input.

1. Use the Simulink Library Browser window toolbar to open a new model. Initialize the Model Configuration Parameters and save your file as `SingleEcho.slx`.

   NOTE: Ultimately, you will want to operate this system in real-time. However, Simulink has a useful feature, the simulation tool (explored in Lab #2), that allows you to quickly test your system using pre-recorded signals. Thus, you will begin by designing a Simulation (SIM) version of the model and progress to a real-time model. **Make sure to use blocks from the DSP System Toolbox.**

2. You will use **Signal from Workspace** in the DSP System Toolbox as your input and will send your output to the MATLAB workspace (**To Workspace**). Find the appropriate input and output blocks and place them in your model.

3. Modify the blocks so that the system loads the signal `Hello` (file available on Canvas) from the MATLAB workspace and exports the output signal to the variable `Echo` in the MATLAB workspace. Make sure to change the "Save Format" of the **To Workspace** block to "Array."

   NOTE: The sampling rate of `Hello` is 8000 Hz and the signal should be

processed with a frame size of 64, with cyclic repetition.

4. Find the appropriate blocks and make the connections necessary to implement the single echo difference equation: $y[n] = x[n] + a\,x[n-D]$.

   - Set the parameters as follows: $a = 0.8$ and $D = 4000$.

   NOTE: You should use the delay block under DSP System Toolbox > Signal Operations.

   Checkpoint (3/7): Show your model to your TA and explain how you implemented the given difference equation.

5. Download the file `Hello.wav` from Canvas and load it into MATLAB using the command:

```
Hello = audioread('Hello.wav');
```

6. You can now run the simulation by clicking the "play" button (the black triangle in the middle of the model's toolbar). The text box to the left of the triangle indicates how long (in seconds) the simulation will run for. You can observe the progress of the simulation by looking at the bottom of the window frame where the simulation time is displayed (e.g., T = 0.00).

7. When the simulation has stopped, switch back to the MATLAB window and listen to the output that was created using the command: `soundsc(out.Echo, 8000)`. Do you hear an echo?

   Checkpoint (4/7): Play the system's output for your TA.

   Discussion (1/5): What do you hear when a signal is processed using the single echo system? Why does the output sound this way? (Relate your observations to the equation describing the system.)

8. Now, refer back to the difference equation. There are two parameters that affect the output (a and D). Change these parameters in your model by double-clicking on the corresponding block and explore the effect of each parameter (one at a time) on the output signal.

   Discussion (2/5): What effect does changing each of the parameters, a and D, have on the output signal?

9. Now you are ready to implement the Single Echo system in real-time using the Audio System Toolbox. To do this, save your model as `SingleEcho.slx`. Then change the input and output blocks of your model:

   (a) Change the input to an **Audio Device Reader** block that accepts input from the microphone. Set your block parameters carefully.
   (b) Change the output to an **Audio Device Writer** block

10. Change the configuration parameter stop time to **inf** and run the model

11. Test your system using the microphone (MIC IN) and a pair of headphones (HEADPHONE). Do you hear an echo?

Deliverable (1/3): For your lab report, answer the following questions/prompts:

1. How is the Simulink model related to the difference equation?
2. What do you hear when a signal is processed using the single echo system? Why does the output sound this way? (Relate your observations to the equation describing the system.)
3. Provide an analytic expression and a sketch for the impulse response, *h[n]*, of the single echo system.
4. How can you interpret the expression of *h[n]* in terms of your observations?
5. What effect does changing each of the parameters, a and D, have on the output signal?
6. Assuming a sampling rate of 8000 Hz, how would you choose the delay, D, to achieve a time delay of 0.25 seconds? (Recall that the exponent of 'z' in the delay block corresponds to the number of samples the signal is delayed by.)
7. What properties does the single echo system have (linear, TI, stable, memoryless, causal)?

5.3 Exercise #3: Reverberation

For this experiment, you will explore the reverberation effect. As explained in the Background section, reverberation is the result of an infinite number of echoes added together. You will begin by building your own simple reverberator, then you will have the opportunity to listen to the more natural allpass reverberator and make qualitative comparisons.

1. Load a speech or music file into MATLAB, saving the file to the variable `Hello`.
2. Use the Simulink Library Browser window toolbar to open a new model. Initialize the Model Configuration Parameters. Save your file as `SimpleReverb.slx`. (Alternatively, you may start with your `SingleEcho.slx`, as some of the blocks will be the same.)
3. Find and connect the appropriate blocks so that your model loads a signal (`Hello`) from the MATLAB workspace, implements the simple reverberator $y[n] = x[n] + a\, y[n - D]$, and sends the output (`Reverb`) to the MATLAB workspace. Set the parameters as follows: $a = 0.8$ and $D = 4000$.

   Checkpoint (5/7): Show your TA your completed Simulink model.

   Discussion (3/5): How is the Simulink model related to the difference equation?
4. Run the simulation and listen to the result using the `soundsc` command in MATLAB.
5. Now, modify the a and D parameters (one at a time) and explore the effect of each on the output by running the simulation again.

Discussion (4/5): How does changing a and D affect the output?

6. Once you feel you have a good understanding of how the system functions, modify your model so that it can run in real time using the audio system toolbox.
7. Using the MIC IN and HEADPHONES, speak into the microphone and listen to the response of the system.

Checkpoint (6/7): Play the output of your real-time reverberator for your TA.

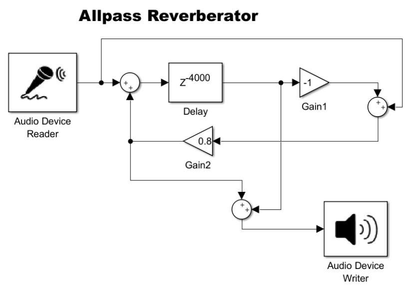

Figure 4. Allpass Reverberator.

8. Download the allpass reverberator Simulink file, `reverb_audio_toolbox.slx`, from Canvas and ensure that it looks like the model shown in Figure 4. Run the system and, using the MIC IN and HEADPHONES, speak into the microphone and listen to the response of the system.

Deliverable (2/3): For your lab report, answer the following questions/prompts:

1. How is the Simulink model related to the difference equation?
2. What do you hear when a signal is processed using the simple reverberation system?
3. What effects does changing the parameters a and D have?
4. Compare the simple reverberation filter with the single echo filter with the same delay and gain. How do they differ?
5. Compare the simple and allpass reverberators. How are they similar? How are they different? Why do you think the allpass reverberator sounds more natural?

5.4 Exercise #4: Flange Effect

For this final experiment, you will explore the flange effect. As explained in the Background section, the difference equation describing a flanging system is $y[n] = x[n] + a\,x[n-D]$ where the time-varying delay is given by:

$$d(n) = D + \frac{D}{2}\left(1 - \sin\left(\frac{2\pi f_d n}{f_s}\right)\right)$$

Figure 5 shows the Simulink block diagram, `flange_audio_toolbox.slx`, which implements this system.

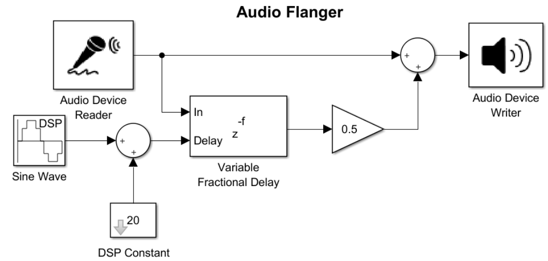

Figure 5. Audio Flanger.

1. Download the file `flange_audio_toolbox.slx` from Canvas and load the file in Simulink.
2. Now you are ready to implement the Flange Effect system in real-time using the audio system toolbox. It should look like the figure at the beginning of this section. Run the simulation with your real-time microphone input.

   Checkpoint (7/7): Show your TA the Flange model and run it for them with real-time microphone input.

3. Load the speech file, `Hello.wav`, or your own recorded speech or music files (e.g., the music you created in the Music Synthesis Project) into MATLAB using the `audioread` command, saving the loaded file into the variable `Hello`.
   (a) Replace the Audio Device Reader block with a Signal From Workspace Block
   (b) Change both the sine wave and DSP constant blocks to output doubles
4. Run the simulation and listen to the output (`Flange`) in MATLAB.
5. Now, modify the parameters of the model one at a time (DSP constant, flanging frequency, gain) and observe the effect of each parameter on the output signal.

   Discussion (5/5): What effect does changing the gain, delay, and flange frequency have?

6. Test your system using recorded music or speech played using an audio player (a music file, `Song.wav` has been provided) (LINE IN) and a pair of headphones (HEADPHONE).
   (a) Be sure to switch your source block to a From Multimedia File block.
   (b) `Song.wav` is sampled at a higher sampling frequency (44.1 kHz) than `Hello.wav` because music has a higher frequency range than speech (we will talk about this more in the Sampling and Aliasing lab). A few changes must be made to the blocks to account for this:
      i. Change both the sine wave and DSP constant blocks to output doubles
      ii. Change From Multimedia File block samples per channel to 64
      iii. Change the sine wave block sample time to 1/44100

iv. Change the DSP constant block to a sample time of 128/44100

Deliverable (3/3): For your lab report, answer the following questions/prompts:

1. What do you hear when a signal is processed using the flanging system?
2. What effect does changing the gain, delay, and flange frequency have?
3. Plot the low-frequency oscillator, $d(n)$, for these parameters: D = 4000 samples, f_s = 8000 Hz, f_d = 2000 Hz. You should use the MATLAB function `stem`, as we are working with discrete time signals.
4. What is the maximum delay when D=4000 and f_s = 8000? What is the minimum delay?

5.5 Wrap-Up

When you are done with laboratory work and ready to leave, please follow this sequence:

1. Log off of the 'Student' computer account
2. Clean up laboratory space (return headphones, microphone, and cables to storage boxes)

6 Lab Report

6.1 Objectives (5)

Summarize the stated objectives in your own words.

6.2 Background (10)

Briefly describe each audio effect (echo, simple reverberation, allpass reverberation, and flange) in your own words (include the relevant equations).

6.3 Results and Discussion (80)

Write up your answers to deliverables (1-3).

6.4 Conclusions (5)

Summarize what you have learned through this laboratory exercise.

Include the statement “I have adhered to the Duke Community Standard in completing this assignment”, along with your signature, on the cover page and properly cite any sources you reference in your written report.

7 References

8 Updates

- August 2021: Converted to LaTeX.
- Fall 2020: Updated to MATLAB 2020a.