CPE 367 – Experiment 5a – v6
Reverb Audio Effect – Dr F DePiero (34 Points)

Overview and Motivation

The signal processing in this experiment involves digital filters with particular delay characteristics and a with a specific style of impulse response. This is different than in many cases when the magnitude of the frequency response is the more critical aspect of the design.

Reverb is an effect that is naturally produced in a music hall or auditorium. It is caused when sound heard by the listener reflects from many surfaces. We will implement a digital filter that simulates the reverb effect. Our design is based on the original paper by Schroeder in 1962. The impulse response of Schroeder's reverb includes thousands of echoes per second. Schroeder used multiple stages (in parallel and cascade). See Fig 3. Each filter is IIR, thus generating many echoes before the response dies out. By combining multiple filters, each subsequent filter generates many echoes from the former, to generate the many echoes needed.

FYI, one of the filters used in the reverb is referred to as an 'all pass' meaning that it passes all frequencies without attenuation. This filter is designed specifically for its delay properties, allowing the full spectrum of the music to pass through.

Learning Objectives

- Find the equivalent difference equation for an IIR system
- Implement parallel and cascaded IIR filters
- Determine the impulse response for a simple IIR system algebraically
- Compute an IIR response, in software, for testing & verification

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) Finding IIR difference equations

The reverb effect by Schroder includes two types of filters, a comb filter (Figure 1) and an all-pass (Figure 2). See M. R. Schroeder, "Natural Sounding Artificial Reverberation," J of the Audio Engineering Society, v10 n3, July 1962.

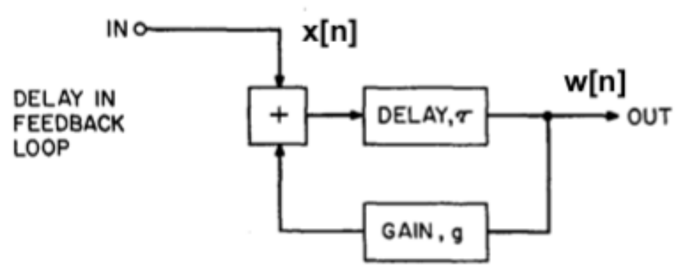

Figure 1. IIR comb filter with delay and feedback (from Schroeder, g = 0.7)

The comb filter does not have a flat frequency response. It eliminates certain frequencies, depending on the amount of delay. For example, consider if the delay was equal to half the period of a sinusoid. The feedback would result in a zero for the filter (adding trough to peak, for example).

- **1a) For the comb filter of Figure 1, find a difference equation an expression for w[n] in terms of x[] and w[] (4)**

See the handout "hardware realizations.pdf" for examples and suggestions for finding a difference equation, given a hardware block diagram

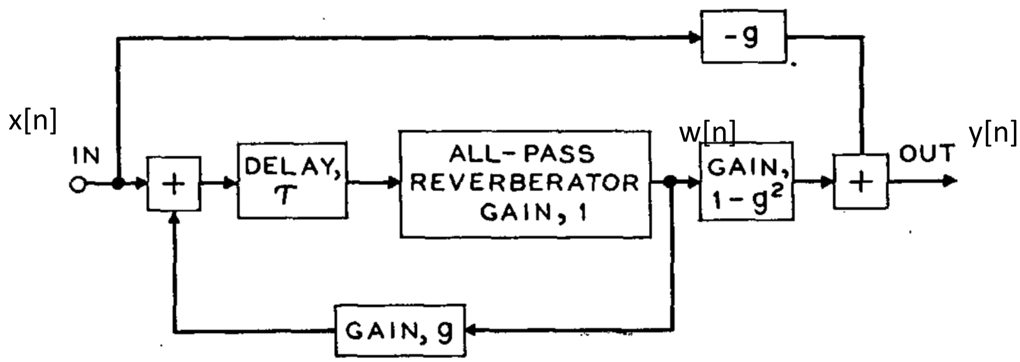

Figure 2. IIR all-pass filter. The "ALL-PASS REVERBERATOR GAIN" is a gain of 1. The other gains, g = 0.7, (from Schroeder). Consider calling the 'update' method at the input to the delay and calling the 'get' method at the output of the delay, for example.

- **1b) For the all-pass filter of Figure 2, find a difference equation for w[n] in terms of x[] and w[] (4)**
- **1c) For the all-pass, find a difference equation for y[n] in terms of w[n] and x[n]? (4)**

The IIR all-pass has a flat frequency response. This can be best explained via Z transforms which will be covered later in the course. FYI, there is a pole-zero cancellation that results in a uniform response for all frequencies.

2) Reverb Implementation

Implement the reverb in Python. This requires 6 IIR filters. Each will need two FIFOs, one for the history of the input and one for the history of the output. (Although the four comb filters in parallel can share the FIFO storing their input history).

- **2a) Documentation: Paste the sections of your Python code that implements the six digital filters, using your FIFO (10)**
- **2b) Documentation: Upload your output WAV file with the reverb effect (6)**

Parameters for the reverb – see Figure 3.

- Delays tau1-tau4 should be distinct values, chosen on the range of 30-45 mSec.
- Delays tau5 and tau6 should be 5 and 1.7 mSec.
- Gains g1 – g6 should all be 0.7
- Gain g7 can be adjusted to your liking, and to prevent clipping

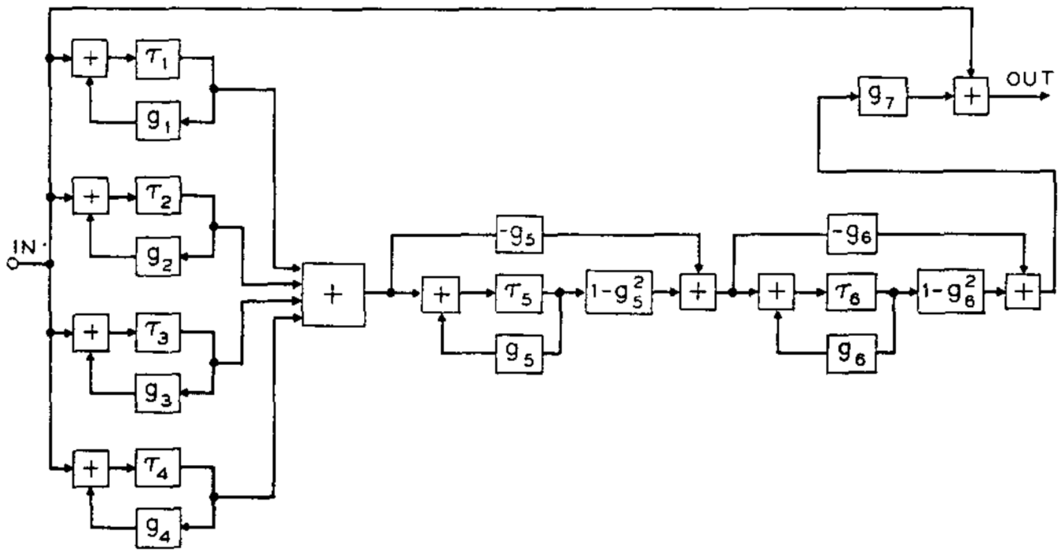

Figure 3. Reverb Effect (from Schroeder)

Introduce a fractional gain prior to "IN" to avoid clipping and distortion. Also adjust g7.

3) Testing and Verification via Impulse Response

To check the operation of your filter, generate an output that is due to a single impulse. This will be similar to h[n], the impulse response, except here use $\boldsymbol{x[n] = 30000\ \delta[n-100]}$**.** Save the response in a WAV file, with ~0.75 Sec duration. Listen to it the impulse response! Then use the **plot_signal** MatLab command to make a plot. You should see something similar to Figure 4.

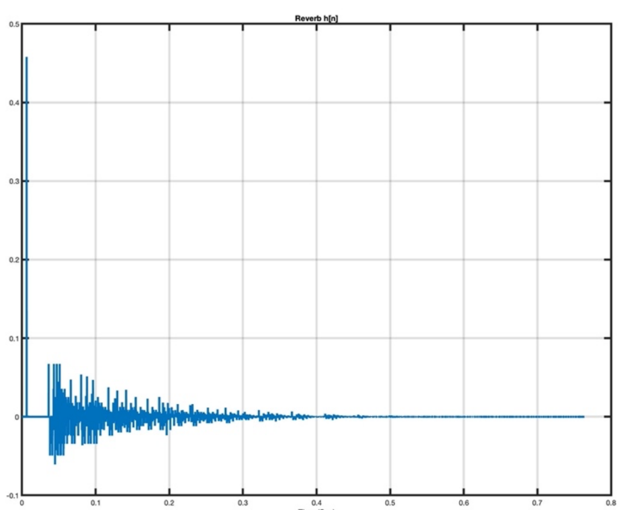

Figure 4. Response to an impulse, $\boldsymbol{x[n] = 30000\ \delta[n-100]}$**. File duration ~0.8 Sec.**

Note the initial part of the response (far left) due to the direct sound (or dry signal), which is followed by a brief delay and then the tapering part of the response associated with artificially-reflected sounds. Your results may vary, due to choices of gains (for the initial attenuation and g7) and due to choices for the various delay parameters.

- **3) Documentation: Paste a copy of your (scaled and delayed) impulse response in to your report (6)**

Suggestions

- To begin setting up the reverb effect, consider implementing and testing one stage of the comb filter or the all-pass first. A single stage should have negligible affect (the output should sound approximately the same as the input).
- Note the four comb filters are in parallel (on left) and share the same input
- Note the two all-pass filters are in cascade
- When implementing the whole reverb, consider computing the various signals working from the "IN" on the left and proceeding to the right.
- An example of the reverb is posted on the course Canvas site
- Use instances of your FIFOs as needed (at least six required). Consider calling the 'update' method to store the input of a delay unit. Call the 'get' method to access the output of the delay unit.
- See Schroeder's original paper, as you interest and time permit (posted on Canvas)
- Consider attenuating the input (from the WAV file) before the reverb "IN". There is gain because of the four parallel filters. Attenuate "IN" to avoid clipping and adjust g7.

Unanswered Questions

- How do we analyze the response of IIR filters, in particular the transient response?
- How can we find poles and zeroes (Answer: Z Transform)?
- How do poles and zeroes impact frequency response and what is pole + zero cancellation?