E85.2607: Lecture 3 – Delay-based effects
Basic delay

- Delay signal by $M$ samples (or $\tau = M/f_s$ seconds)
- Not much to hear on its own
- Useful for compensating for other delays
  - Acoustic delays in sound reinforcement
  - Processing delays in e.g. long FIR filter
- Implement using delay line or buffer
- Poles and zeros? Frequency response?
What if $M$ is not an integer?

Use interpolation to simulate fractional delay

- Estimate the value of sample that doesn’t exist
Fractional delays – strategies

1. Linear interpolation

\[ y[n] = x[n - (M + 1)] \frac{\text{frac}}{M} + x[n - M](1 - \text{frac}) \]

What kind of filter is this?

2. Allpass interpolation

\[ y[n] = x[n - (M + 1)] \frac{\text{frac}}{M} + x[n - M](1 - \text{frac}) - y[n - 1](1 - \text{frac}) \]

3. Sinc interpolation, Many more...
FIR comb filter

Mix input with a delayed version of itself:

\[ y[n] = x[n] + g \times [n - M] \]

\[ H(z) = 1 - g \times z^{-M} \]
FIR comb filter – frequency response

![Diagram of a comb filter with magnitude response](image)

- Positive coefficient
- Negative coefficient

Magnitude vs. Frequency
Mix input with a delayed version of filter output:

\[ y[n] = c \times[n] + g y[n - M] \]

\[ H(z) = \frac{c}{1 + g z^{-M}} \]
IIR comb filter – frequency response

**Diagram:**

- Input: $x(n)$
- Gain: $c$
- Delay: $z^{-M}$
- Output: $y(n)$
- Positive coefficient: $1/(1-g)$
- Negative coefficient: $1/(1+g)$

**Graph:**

- Magnitude
  - Positive coefficient: $1/(1-g)$
  - Negative coefficient: $1/(1+g)$

**Legend:**

- Positive coefficient
- Negative coefficient

E85.2607: Lecture 3 – Delay-based effects

2010-02-04
Universal comb filter

Combine FIR and IIR comb filters into a single structure

\[ y(n) = x(n) + x_h(n-M) - (x(n-M) \times x_h(n)) \]

<table>
<thead>
<tr>
<th>Type</th>
<th>BL</th>
<th>FB</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIR comb filter</td>
<td>X</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>IIR comb filter</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>allpass</td>
<td>(a)</td>
<td>(-a)</td>
<td>1</td>
</tr>
<tr>
<td>delay</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
What do these comb filters sound like?

PD time...
Delay-based effects: Echo

- Mix input with a long delay (> 50ms)
  - FIR comb filter...

- Haas/Precedence Effect
  - If the same sound comes from two different locations, the sound will seem to come from the direction of the sound which arrives first. Our sensory system ignores subsequent sounds.
  - If the delay is larger than about 50 ms, the sounds will be heard as distinct events.

- Different delays lead to different effects
  - Slapback/doubling: echo with short delay << 50ms

- Let's build one
Delay-based effects: Vibrato

- Time-varying delay (5–10 ms)
- LFO to control delay variation (5–14 Hz)
Delay-based effects: Flanger

- Time varying slapback (delay < 15 ms)
- Use low-frequency oscillator to vary the delay (\(\sim 1\) Hz)
Delay-based effects: Chorus

- Mix input with randomly delayed versions of itself
- Sounds like a *chorus* of sounds that are not quite in sync
Bringing it all together

Common universal comb filter structure:

![Comb Filter Diagram]

Typical parameters:

<table>
<thead>
<tr>
<th>Effect</th>
<th>BL</th>
<th>FF</th>
<th>FB</th>
<th>Delay</th>
<th>Depth</th>
<th>MOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vibrato</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0 ms</td>
<td>0-3 ms</td>
<td>0.1-5 Hz Sine</td>
</tr>
<tr>
<td>Flanger</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0 ms</td>
<td>0-2 ms</td>
<td>0.1-1 Hz Sine</td>
</tr>
<tr>
<td>(white) Chorus</td>
<td>0.7</td>
<td>1</td>
<td>-0.7</td>
<td>1-30 ms</td>
<td>1-30 ms</td>
<td>lowpass noise</td>
</tr>
<tr>
<td>Doubling</td>
<td>0.7</td>
<td>0.7</td>
<td>0</td>
<td>10-100 ms</td>
<td>1-100 ms</td>
<td>lowpass noise</td>
</tr>
</tbody>
</table>
- Introduction to Digital Filters
  - Analysis of a Digital Comb Filter
  - through “Pole-Zero Analysis”
- DAFX, Chapter 3 (if you have it)