MUS424:
Signal Processing Techniques for Digital Audio Effects `Handout #11`
Jonathan Abel, David Berners `April 25, 2011`

Homework #6: Laboratory Exercise 2
Due Date: May 3, 2011

Time-Varying Resonant Filters

In this lab, we will explore resonant and time-varying filters. The provided file contains two plug-ins incorporating a resonant low-pass filter. The first plug-in, shown in Figure 1, explores a simple resonant low-pass filter with varying cutoff frequency and resonance. The second plug-in, shown in Figure 2, uses an LFO to vary the filter cutoff frequency for a Wah-Wah effect.

Problem 1. [30 Points]

The resonant low-pass filter is specified by its cutoff frequency ω_c and resonance Q. Its transfer function is given by

$$H(s) = \frac{1}{(s/\omega_c)^2 + \frac{1}{Q}(s/\omega_c) + 1}. \tag{1}$$

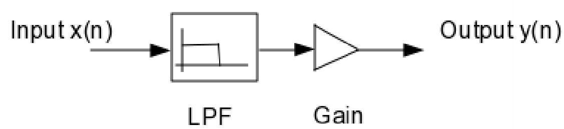

Figure 1: *Resonant Low Pass.*

1(a). [10 Points] Plot the transfer function magnitude and phase, and sketch the trajectory of the transfer function poles for

- $Q = 2^{[-4:2]}$ $\quad \omega_c = 2\pi \cdot 1000$ radians/second
- $Q = 2$ $\quad \omega_c = 2\pi \cdot 1000 \cdot 2^{[-2:2]}$ radians/second

1(b). [20 Points] Complete the functions used to convert the cutoff frequency and resonance into the needed second-order digital filter coefficients. Apply the filter to the white noise sequence supplied, using various cutoff frequencies and resonances.

Problem 2. [30 Points]

In this exercise, the resonant low-pass filter of Problem 1 will be modified to include a low-frequency oscillator (LFO) which drives the filter cutoff frequency.

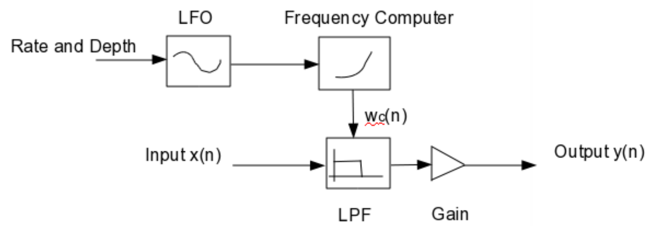

Figure 2: *LFO-Driven Wah-Wah*

2(a). [10 Points] So as to make the filter smoothly varying over time, the user controls will be tracked using leaky integrators. For example, the resonance control will generate a sequence of targets Q_T, which drive the tracking filter,

$$Q(n) = (1 - \lambda)Q_T + \lambda Q(n - 1), \tag{2}$$

to produce a smoothly varying resonance sequence $Q(n)$, according to the forgetting factor λ. Implement leaky integrators for the controls ω_c, Q, and the filter gain g. Design λ to produce a 20 msec time constant. You will only need to implement a few lines in the *slewedParameter* struct. Note: You may also need to add an additional leaky integrator on the time-varying filter center frequency to avoid unwanted clicks or pops.

2(b). [10 Points] Implement a low-frequency oscillator (LFO) with output signal $\mu(n)$ either by

- Forming the sine of a phase counter,

  $$\mu(n) = \sin(\phi(n)), \qquad \phi(n) = \mathrm{rem}(\phi(n - 1) + \delta(\omega_m), 2\pi), \tag{3}$$

  where the phase increment $\delta(\omega_m)$ is given by

  $$\delta(\omega_m) = \omega_m/f_s, \tag{4}$$

  with f_s being the sampling rate. See the matlab `rem` for help.

- The *magic circle* algorithm. See the coursework site for a paper reference.

2(c). [10 Points] Form a *frequency computer* to convert the LFO state $\mu(n)$ and tracked cutoff frequency control $\omega_c(n)$ into a filter cutoff frequency which sweeps exponentially between a factor $1/\sqrt{\rho}$ and $\sqrt{\rho}$ of the specified cutoff frequency, ρ being the frequency ratio control or depth. Use the guitar track provided to verify that the filter cutoff frequency smoothly changes with time.