Delay and Digital Filters

Music 171: Computer Music I
Winter quarter 2012,
University of California, San Diego

Tamara Smyth, tamaras@cs.sfu.ca
School of Computing Science,
Simon Fraser University

Tuesday February 28, 2012

Digital Filters

- Any medium through which a signal passes may be regarded as a filter.
- Typically however, a filter is viewed as something which modifies the signal in some way. Examples include:
  - audio speakers / headphones
  - rooms / acoustic spaces
  - musical instruments
- A *digital* filter is a formula for going from one digital signal to another.

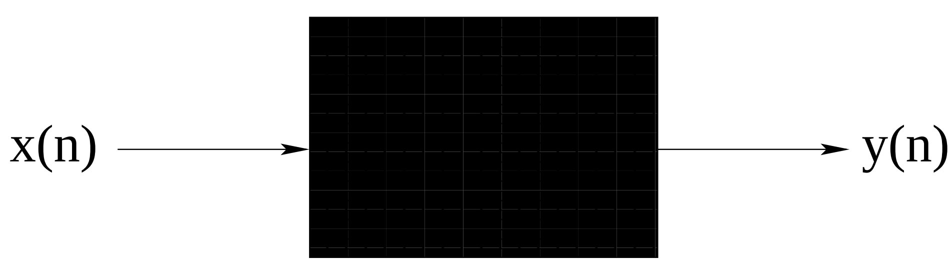

Figure 1: A black box filter.

Inside the Black Box—Pure Delay

- Time-domain implementations of digital filters involve signal *delay*, that is, *delayed* versions of input and/or output signals.
- What does it mean to delay an audio signal?
  - move it later (or earlier) in time
  - change the phase of signal, (the value at time=0)

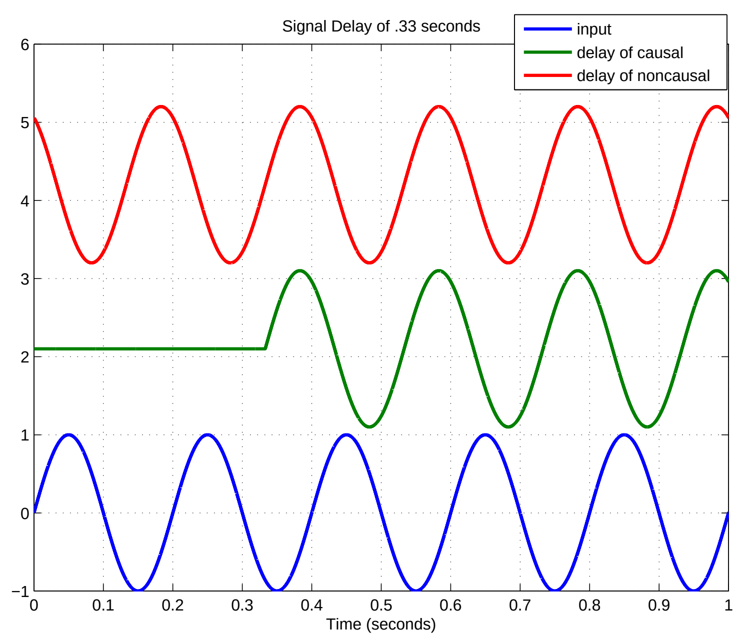

Figure 2: Timeshifting a signal will change the phase of the signal.

Time shifting a signal

- Whenever a signal can be expressed in the form

$$y(n) = x(n - M),$$

$y(n)$ is a *delayed* (time-shifted) version of $x(n)$.

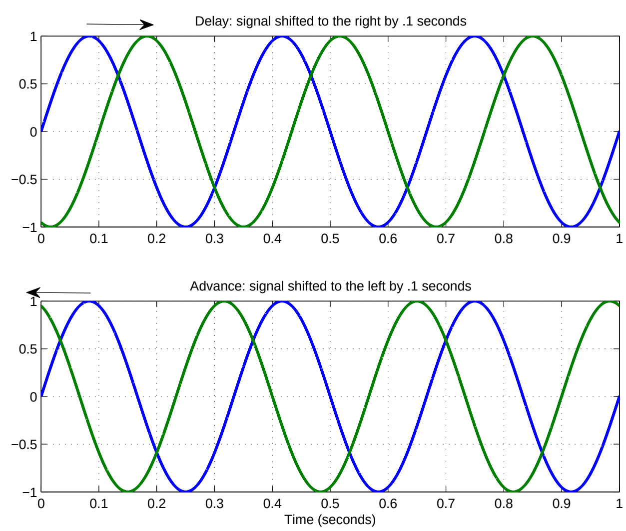

Figure 3: Delay, a shift to the right. Advance, a shift to the left.

- **If M is a positive number**, $y(n) = x(n - M)$, and the shift on the time axis is to the right.
- **If M is a negative number**, $y(n) = x(n + M)$, the signal has been advanced in time and the shift on the time axis is to the left.

Time shift and addition

- Other than hearing a possible silence (for causal signals) before the signal onset, there is no audible effect of a *pure* delay.

$$\boxed{\text{Signals } x(n) \text{ and } x(n-M) \text{ sound the same}}$$

- What happens, however, when a signal $x(n)$ is added to a delayed version of itself $x(n-M)$?

$$y(n) = x(n) + x(n-M)$$

A Running Averager

- Consider a simple case where $M = 1$.

$$y(n) = x(n) + x(n-1).$$

- We are taking a *running average* of the input signal $x(n)$.

$$\boxed{\text{This filter takes the average of two adjacent samples (with a gain of 2).}}$$

Intuitive Analyis at Low Frequencies

- The running average of a signal with little or no variation from sample to sample will be very close to the input signal.
- At DC

$$x_1(n) = [A, A, A, ...].$$

- The output of the filter is

$$\begin{aligned} y(n) &= x_1(n) + x_1(n-1) \\ &= \quad [A, A, A, ...] \\ &\quad + [0, A, A, A, ...] \\ &= \quad [A, 2A, 2A, 2A, ...] \end{aligned}$$

The output is effectively the same as the input, but with a gain of 2.

Intuitive Analyis at High Frequencies

- The running average of an input signal with significant variation from sample to sample will be very different from its input.
- At $f_s/2$ (Nyquist limit)

$$x_2(n) = [A, -A, A, ...].$$

- The output of the filter is

$$\begin{aligned} y(n) &= x_2(n) + x_2(n-1) \\ &= \quad [A, -A, A, ...] \\ &\quad + [0, A, -A, A, ...] \\ &= \quad [A, 0, 0, 0, ...] \end{aligned}$$

The output is different from the input—complete attenuation.

What about all the frequencies in between?

- This filter boosts low frequencies while attenuating higher frequencies. It is, therefore, a *lowpass* filter.
- We may find the frequency response of the filter by checking the behaviour of the filter at every possible frequency between 0 and $f_s/2$ Hz (sinewave analysis).

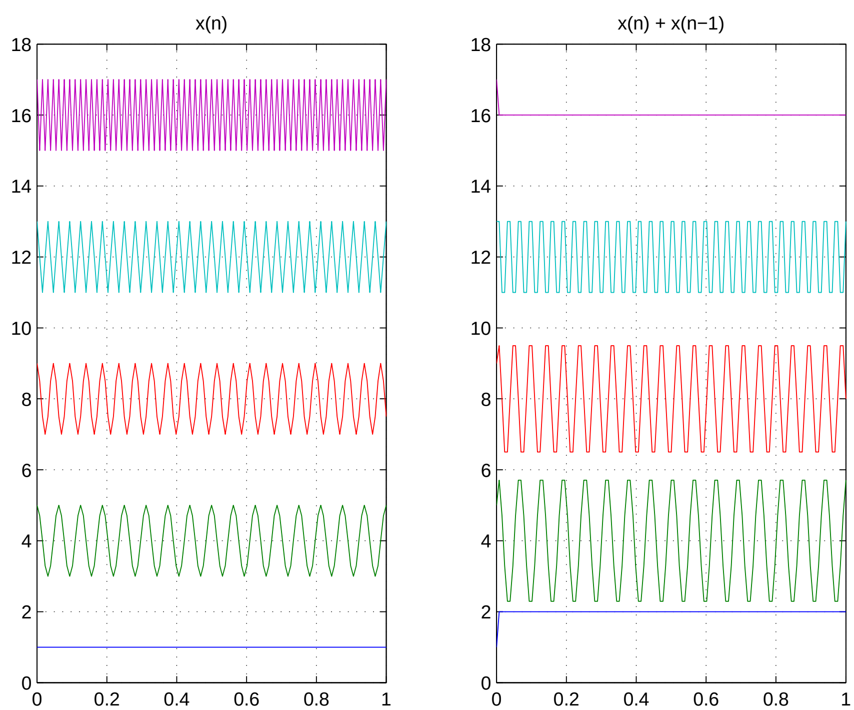

Figure 4: Sinewave analysis.

Impulse Response

- Alternatively, we can use an input signal that contains all frequency components, and then we only have to do the "checking" operation once.
- An input signal with the broadest possible spectrum would be an *impulse*.
- The response of a filter to an impulse is called an *impulse response*.

Any filter in a large class known as linear, time-invariant (LTI), is completely characterized by its impulse response.

- What is the impulse response of this filter?

Frequency Response

- The spectrum of the impulse response gives us the *frequency response* from which we may see how the filter modifies the *amplitude* and *phase* of a signal's sinusoidal components.

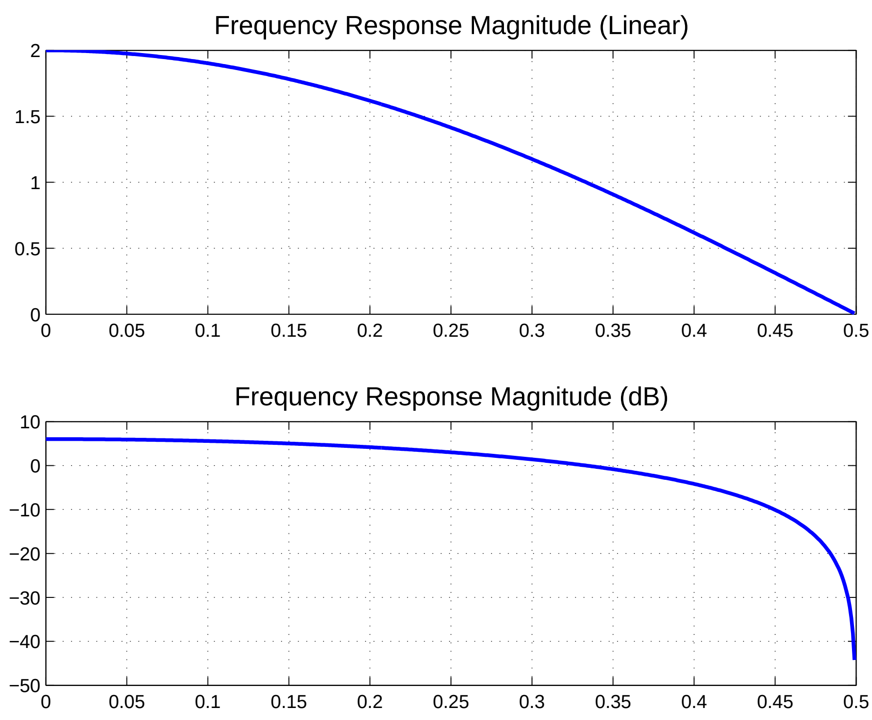

Figure 5: Magnitude of the Frequency Response shows a low-pass characteristic.

Response at the Cutoff Frequency

- Look a little closer at the filter's response to $f_s/4$.

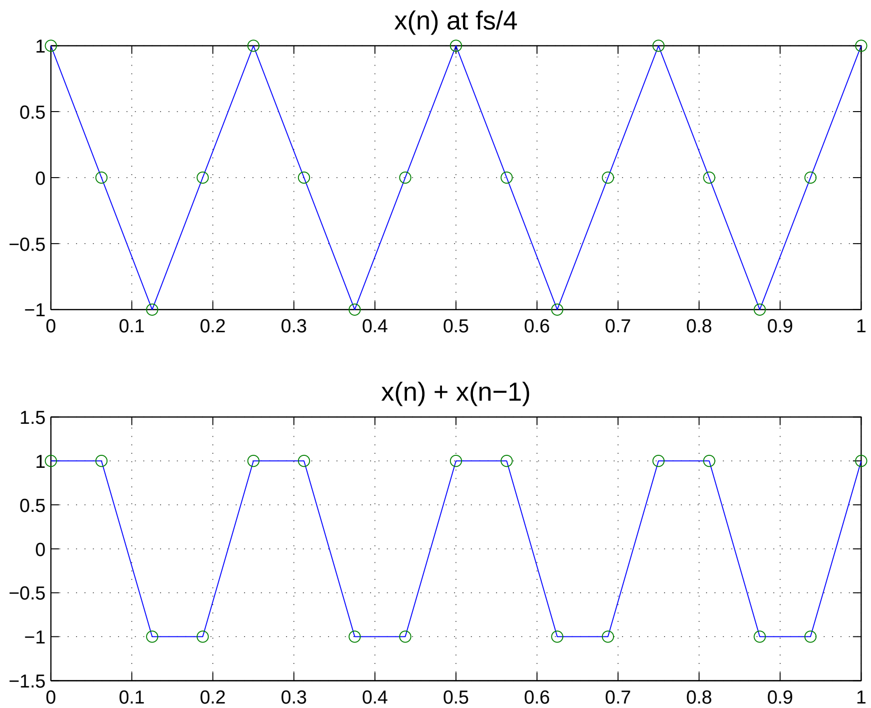

Figure 6: Filter behaviour at $f_s/4$.

Interpreting the Phase

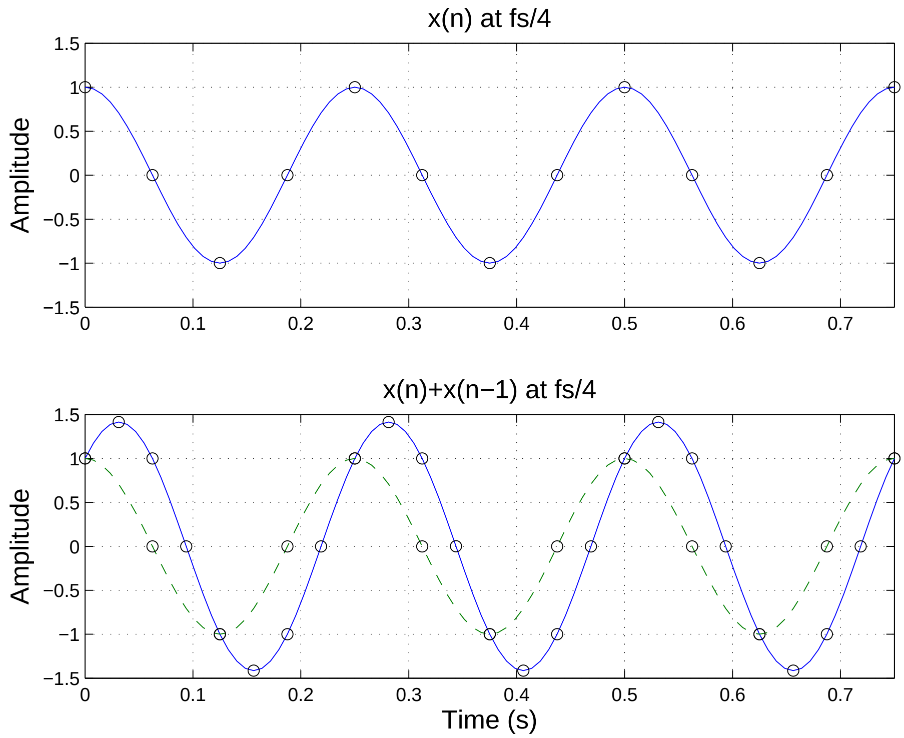

Figure 7: Filter behaviour at $f_s/4$.

- This filter is *delaying* this frequency by half a sample.
- In fact, this filter delays *all* frequencies by half a sample.

Linear Phase Filters

- Filters that delay all frequencies by the same amount are called *linear phase* filters.
- Linear phase filters have a symmetric impulse response.

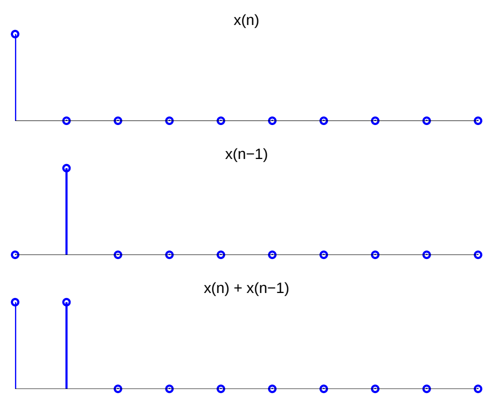

Figure 8: Filter impulse response.

- For this filter, the impulse response is symmetric about sample 0.5, which corresponds to a waveform delay of one-half sample at all frequencies.

Phase Delay

- The value about which the impulse response is symmetric is the *phase delay* of the filter.

> A "simple waveform delay" means the waveform will not change with a change in frequency.

- Linear phase is desirable because it delays all frequencies by the same number of samples and that means **no phase distortion**.

Changing Filter Coefficients

- Consider the following variation on the two-point averager (lowpass filter):

$$y(n) = x(n) - x(n-1).$$

How does changing the addition to a subtraction change the filter?

> Changing the addition to a subtraction changes the *coefficients* of the filter.

- At DC the output becomes

$$\begin{aligned} y(n) &= x_1(n) - x_1(n-1) \\ &= \quad [A, A, A, ...] \\ &\quad - [0, A, A, A, ...] \\ &= \quad [A, 0, 0, 0, ...] \end{aligned}$$

- At the Nyquist limit the output becomes

$$\begin{aligned} y(n) &= x_2(n) - x_2(n-1) \\ &= \quad [A, -A, A, ...] \\ &\quad - [0, A, -A, A, ...] \\ &= \quad [A, -2A, 2A, -2A, ...] \end{aligned}$$

Notch and Bandpass Filters

- Consider next, changing the delay value of the second term:
$$y(n) = x(n) + x(n-2).$$
- This changes the filter order to 2 and effectively sets the $x(n-1)$ term to zero.

$\boxed{\text{The filter } \textit{order} \text{ is the value of its highest delay.}}$

- This filter passes both DC and the Nyquist limit, but attenuates $f_s/4$. It is a notch filter.
- The filter given by
$$y(n) = x(n) - x(n-2)$$
rejects DC and the Nyquist limit, and boosts $f_s/4$. It is a bandpass filter.

Plots of simple filters

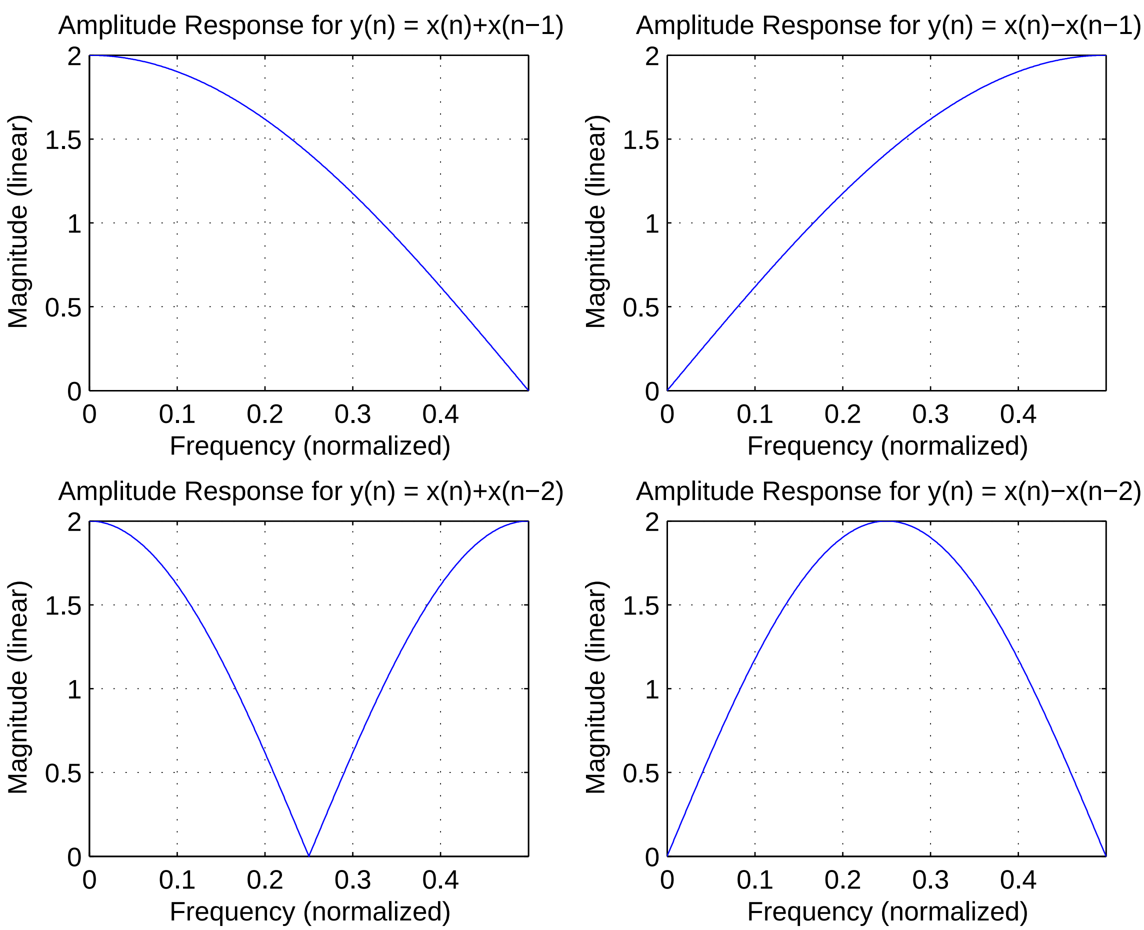

Figure 9: Amplitude Responses for simple filters

.

Increasing the Filter Order

- Let's return now to the simple low-pass filter

$$y(n) = x(n) + x(n-1)$$

- Increasing the order will increase the number of samples averaged

$$y(n) = x(n) + x(n-1) + x(n-2),$$

and the waveform will be smoothed (with a more gentle slope to zero) which corresponds to a lowered cutoff frequency

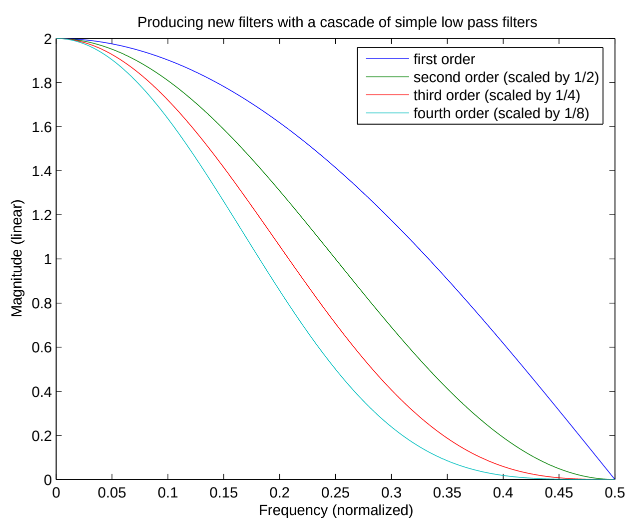

Figure 10: Lowpass filters of increasing order.

Generalized FIR filter

- Several different (nonrecursive) filters can be made by changing the delay and the coefficients of the filter terms,

$$\begin{aligned} y(n) \;=\; & b_0 x(n) + b_1 x(n-1) + \ldots \\ & b_2 x(n-2) + \ldots + b_M x(n-M), \end{aligned}$$

  where M is the maximum delay and thus the order of the filter.

- A filter can be defined simply by a set of coefficients. For example if

$$b_k = \{1, 3, 3, 1\},$$

  the filter is third order (has a maximum delay of $M = 3$), and can be expanded into the difference equation

$$y(n) = x(n) + 3x(n-1) + 3x(n-2) + x(n-3)$$

Increasing the phase delay

- Again returning to the simple low pass filter...
- What happens to the spectrum when the delay of the second term is increased?

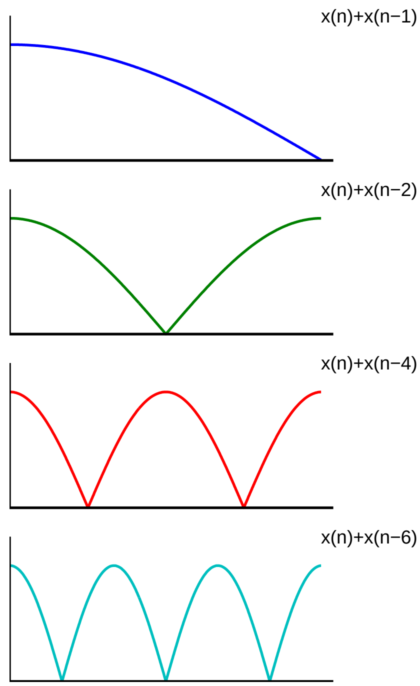

Figure 11: Increasing the filter order (the value of the delay in the second term), causes the appearance of regularly apaced peaks and notches in the magnitude spectrum.

Recursive (IIR) Filters

- Using FIR filters to reproduce a desired frequency response often requires using a high order filter.
- A high order filter means a long impulse response, a greater number of coefficients, and more computation.
- It is often possible to reduce the number of **feedforward** coefficients by introducing **feedback** coefficients.
- A simple first-order recursive low-pass filter is given by

$$y(n) = x(n) + .9y(n-1)$$

First-Order Recursive Lowpass Filter

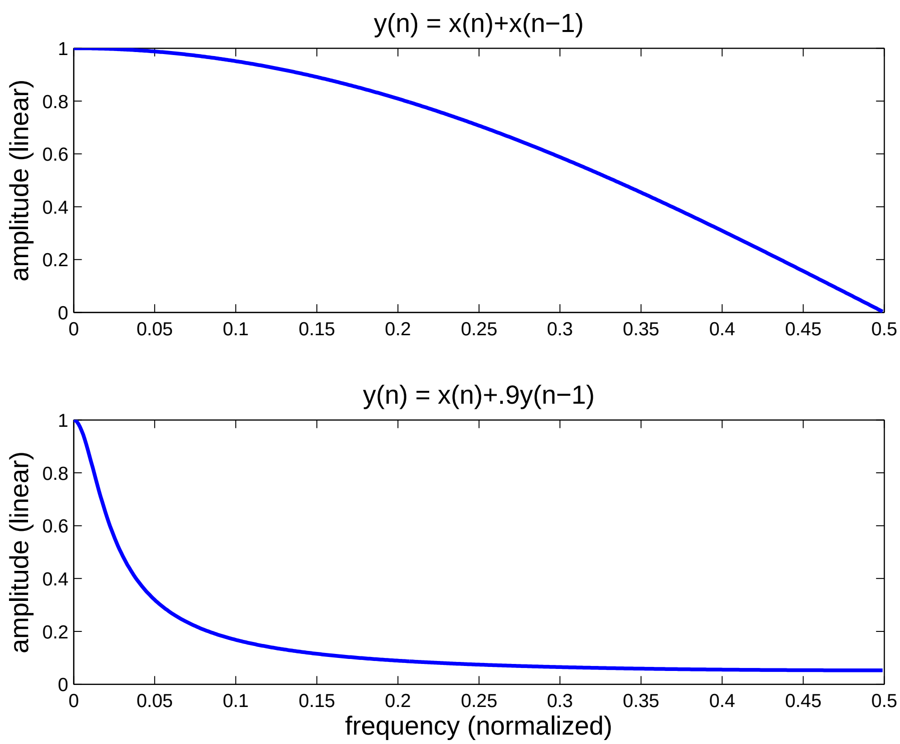

Figure 12: The spectral magnitude of the first-order FIR and IIR (recursive) lowpass filters.

- The general difference equation for LTI filters therefore, includes feedback terms, and is given by

$$\begin{aligned} y(n) = b_0 x(n) \; &+ \; b_1 x(n-1) + \cdots + b_M x(n-m) \\ &- \; a_1 y(n-1) - \cdots - a_N y(n-N) \end{aligned}$$

The Delay Line

- The delay line is an elementary functional unit which models *acoustic propagation delay*.
- It is a fundamental building block of both digital waveguide models and delay effects processors.

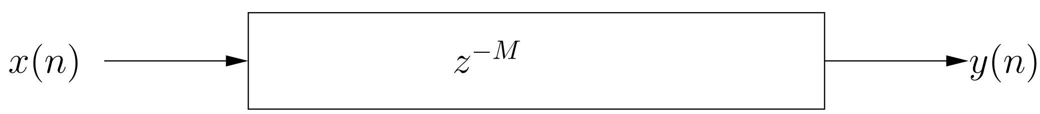

Figure 13: The M-sample delay line.

- The function of a delay line is to introduce a time delay, corresponding to M samples between its input and output

$$y(n) = x(n - M), \quad n = 0, 1, 2, \ldots$$

- It is linear phase, with a phase delay of M samples (it delays all frequencies by this amount).
- This can be seen by the symmetry of the impulse response about the M^{th} sample.

The Simple Comb Filter

- What happens when we multiply the output of a delay line by a gain factor g then feed it back to the input?

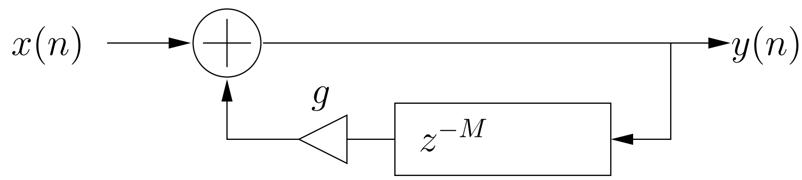

Figure 14: The signal flow diagram of a comb filter.

- The difference equation for this filter is

$$y(n) = x(n) + gy(n - M),$$

- If the input to the filter is an impulse

$$x(n) = \{1, 0, 0, \ldots\}$$

the output (impulse response) will be ...

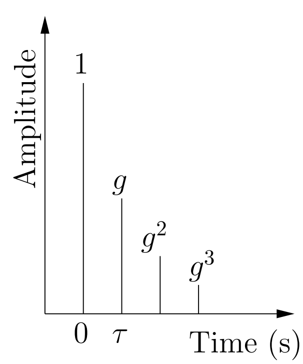

Figure 15: Impulse response for filter $y(n) = x(n) + gy(n - M)$.

A Simple Comb Filter

- Since the pulses are equally spaced in time at an interval equal to the loop time $\tau = M/f_s$ seconds, it is periodic and will sound at the frequency $f_0 = 1/\tau$.

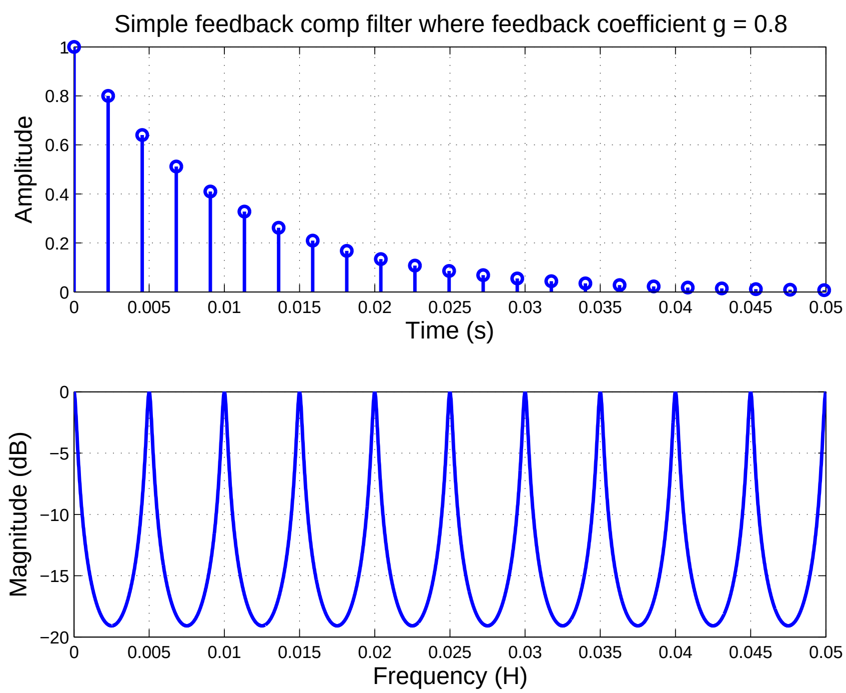

Figure 16: Impulse and magnitude response of a comb filter with feedback g = 0.8.

- The comb filter is so called because its amplitude response resembles the teeth of a comb.
- The spacing between the maxima of the "teeth" is equal to the natural frequency.

Effect of the Feedback coefficient

- The depth of the minima and height of the maxima are set by the choice of g, where values closer to 1 yield more extreme maxima and minima.

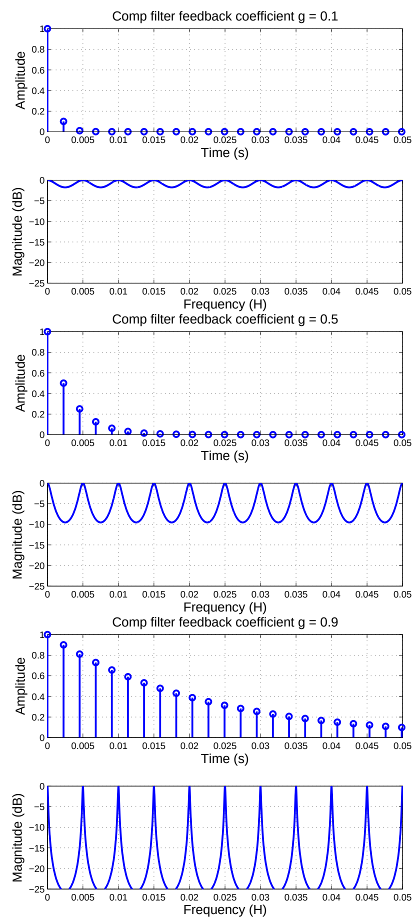

Figure 17: Impulse and Magnitude Response with increasing feedback coefficient.

Comb Filter Decay Rate

- The response decays exponentially as determined by the loop time and gain factor g.
- Values of g nearest 1 yield the longest decay times.
- To obtain a desired decay time, g may be approximated by

$$g = 0.001^{\tau/T}$$

where

$$\begin{aligned} \tau &= \text{the loop time} \\ T60 &= \text{the time to decay by 60dB} \end{aligned}$$

and 0.001 is the level of the signal at 60dB down.

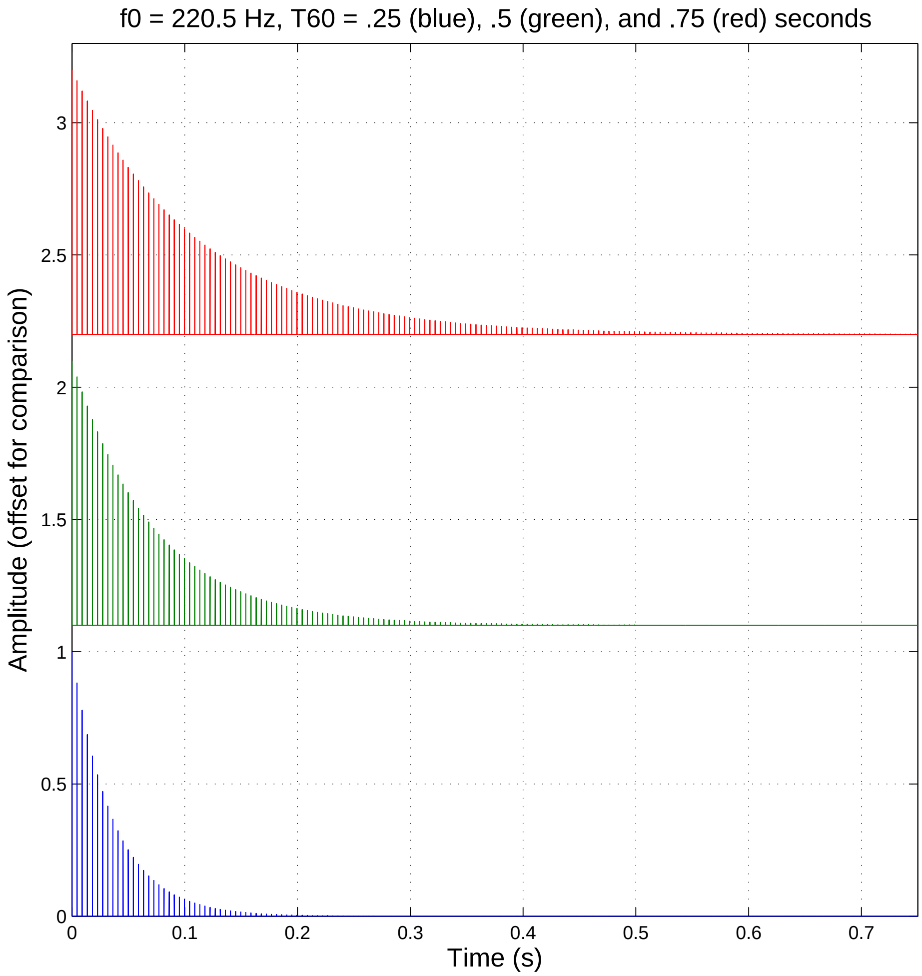

Figure 18: Comb filter impulse responses with a changing the decay rate.

General Comb Filter

- Consider now, adding to the filter a delay element which delays the input by M_1 samples, with some gain g_1.
- The general comb filter is given by the difference equation

$$y(n) = x(n) + g_1 x(n - M_1) - g_2 y(n - M_2)$$

where g_1 and g_2 are the feedforward and feedback coefficients, respectively.

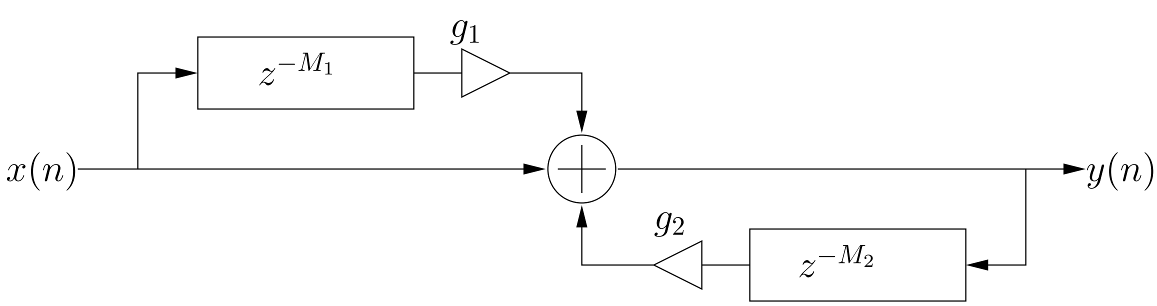

Figure 19: Signal flow diagram for digital comb filters.

A very simple string model

- A very simple string model can be implemented using a single delay line and our simple first-order low pass filter to model frequency-dependent loss.

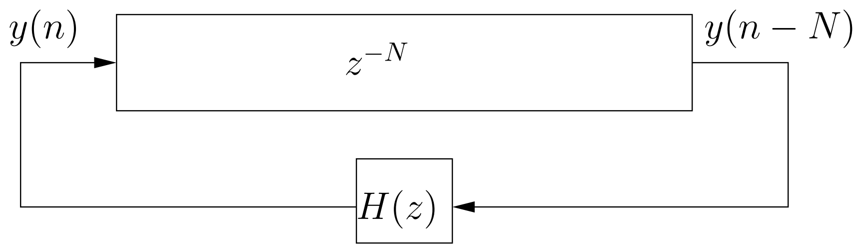

Figure 20: A very simple model of a rigidly terminated string.

- All losses have been *lumped* to a single observation point in the delay line, and approximated with our first-order simple low-pass filter

$$y(n) = x(n) + x(n-1)$$

- Different sounds can be created by changing this filter.
- The Karplus-Strong Algorithm may be interpreted as a **feedback comb filter** (with lowpassed feedback) or a simplified **digital waveguide** model.
- How do you *pluck* the string? (Start off with a noise burst.)