GCT535: Sound Technology for Multimedia

Fundamentals of DSP - Part 1

KAIST Graduate School of Culture Technology

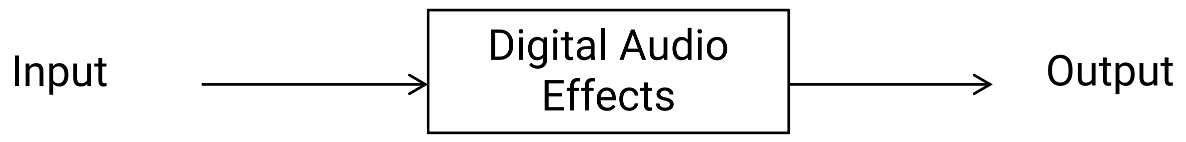
Juhan Nam

Goals

- Introducing digital audio effects
- The fundamentals of digital signal processing
    - Linear time-invariant (LTI) system
    - FIR / IIR Filters
    - Convolution
    - Impulse response
    - Frequency response

Introduction

- Once we record or synthesize sound, we modify the acoustic characteristics using various **digital audio effects**
    - Loudness: volume, compressor, limiter, noise gate
    - Timbre: filters, equalizer, distortion, modulation, chorus, flanger, vocoder
    - Pitch: pitch correction, harmonizer, vibrato
    - Duration: time scale modification, resampling
    - Spatial effect: panning, delay, reverberation, binaural (HRTF)

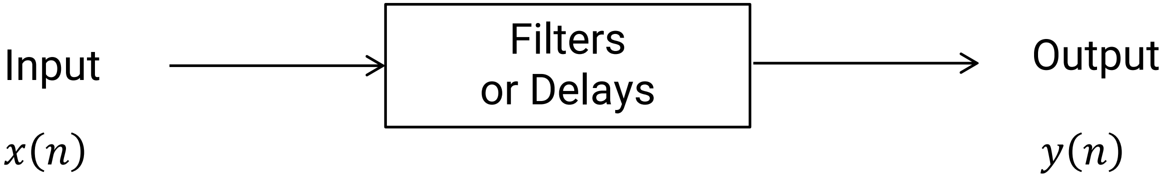


Introduction

- Filters and delays
  - Bi-quads filters: lowpass, bandpass, highpass, equalizers
  - Comb filter (delayline): delay/echo, chorus, flanger, reverb
  - Convolution: resampling, HRTF, reverberation (measured impulse responses)
- Non-linear processing
  - Modulation: amplitude, frequency
  - Distortion and dynamic range control: compressor, limiter, noise gate
- Block-based processing
  - Pitch shifting and Time-scale modification

Filters and Delays

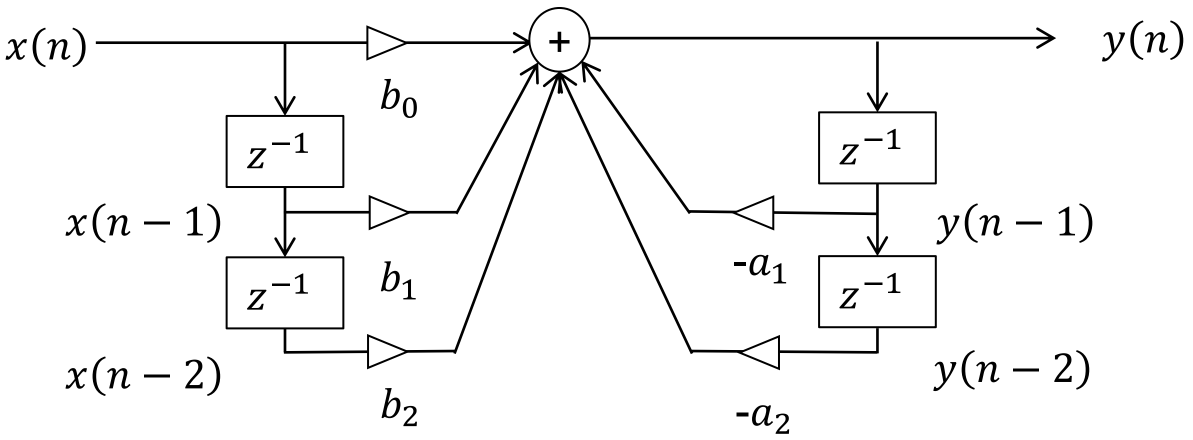
- Take the input signal $x(n)$ as a sequence of numbers and returns the output signal $y(n)$ as another sequence of numbers
- Perform a combination of three mathematic operations upon the input
  - **Multiplication**: $y(n) = b \cdot x(n)$
  - **Delay**: $y(n) = x(n - M)$
  - **Summation**: $y(n) = x(n) + a \cdot y(n - M)$



Filters and Delays

- Bi-quad filter
  - Two delay elements for input or output
  - The delayed output (“feedback”) causes **resonance**
  - Rooted from analog circuits (R-L-C)

$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + b_2 \cdot x(n-2) - a_1 \cdot y(n-1) - a_2 \cdot y(n-2)$$

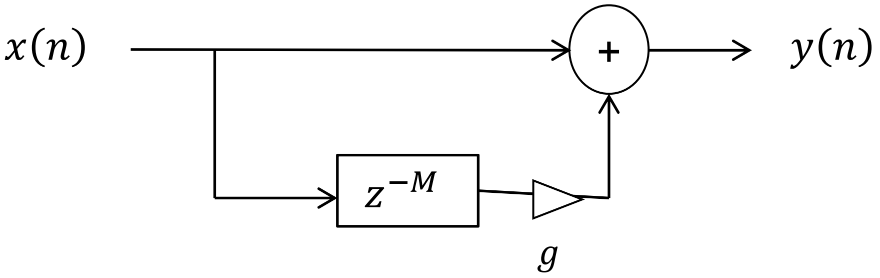


Filters and Delays

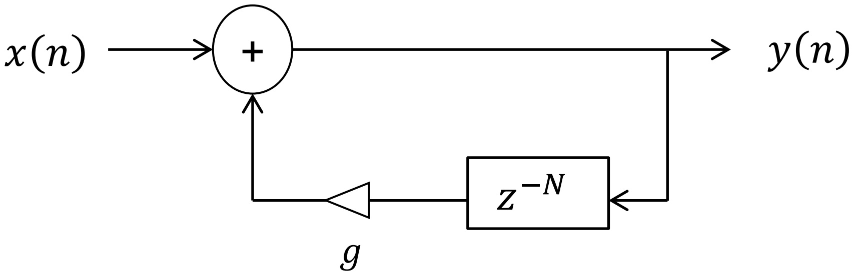
- Comb filter
  - Use a delayline
  - Feedforward is used for chorus/flanger and feedback is for echo/reverb
  - Rooted from magnetic tape recording

$$y(n) = x(n) + g \cdot x(n - M)$$

$$y(n) = x(n) + g \cdot y(n - N)$$



Feedforward Comb Filter

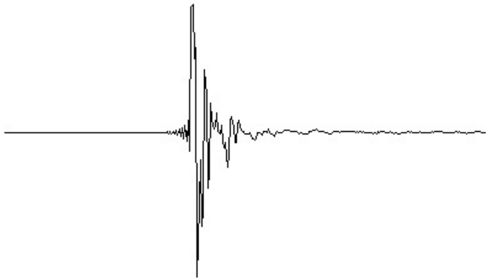


Feedback Comb Filter

Filters and Delays

- Convolution filter
  - Conduct the convolution operation with an **impulse response** of a system
  - The impulse response is often measured from natural objects
    - HRTF: human ears
    - Reverberation: rooms

$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + b_2 \cdot x(n-2) + \cdots + b_M \cdot x(n-M)$$



$h(n) = [b_0, b_1, b_2 \ldots, b_M]$

Linear Time-Invariant Filters

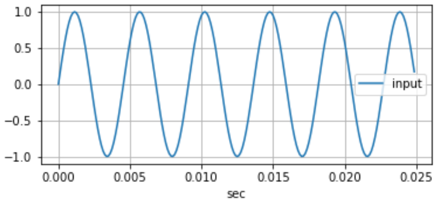
- They are called Linear Time-Invariant (LTI) filters in the context of digital signal processing
- Linearity
  - Scaling: if $x(n) \rightarrow y(n)$, then $\mathrm{a} \cdot x(n) \rightarrow \mathrm{a} \cdot y(n)$
  - Superposition: if $x_1(n) \rightarrow y_1(n)$ and $x_2(n) \rightarrow y_2(n)$, then $x_1(n) + x_2(n) \rightarrow y_1(n) + y_2(n)$
- Time-Invariance
  - If $x(n) \rightarrow y(n)$, then $x(n-N) \rightarrow y(n-N)$ for any N
  - This means that the system does not change its behavior over time

Linear Time-Invariant Filters

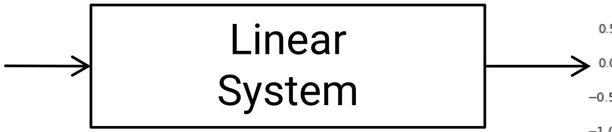
- Non-linear filters
  - $y(n) = x(n)^2$
- Time-variant filters
  - $y(n) = n \cdot x(n) + x(n-1)$
- Modulation
  - Amplitude or frequency modulation generates new sinusoidal components
- Dynamic range compression
  - Input gain is a function of the input: $y(n) = f(x(n)) \cdot x(n)$

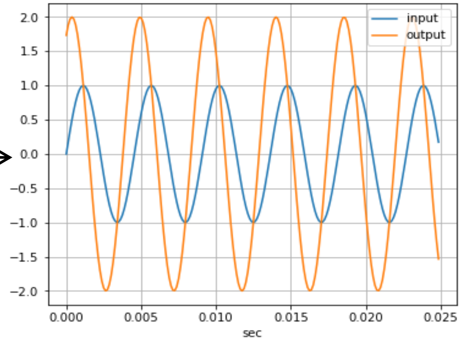
Linear Time-Invariant System

- Remember that sinusoids are eigenfunctions of linear system
  - The input sinusoids changes in amplitude and phase while preserving the same frequency
  - No new sinusoidal components are created



$e^{j\omega t}$





$A(\omega)e^{j\phi(\omega)}e^{j\omega t}$

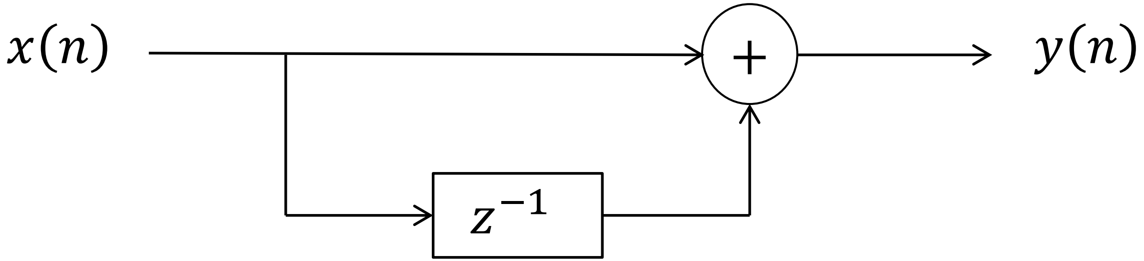
Amplitude Response

Phase Response

The Simplest Lowpass Filter

- Difference equation: $y(n) = x(n) + x(n-1)$

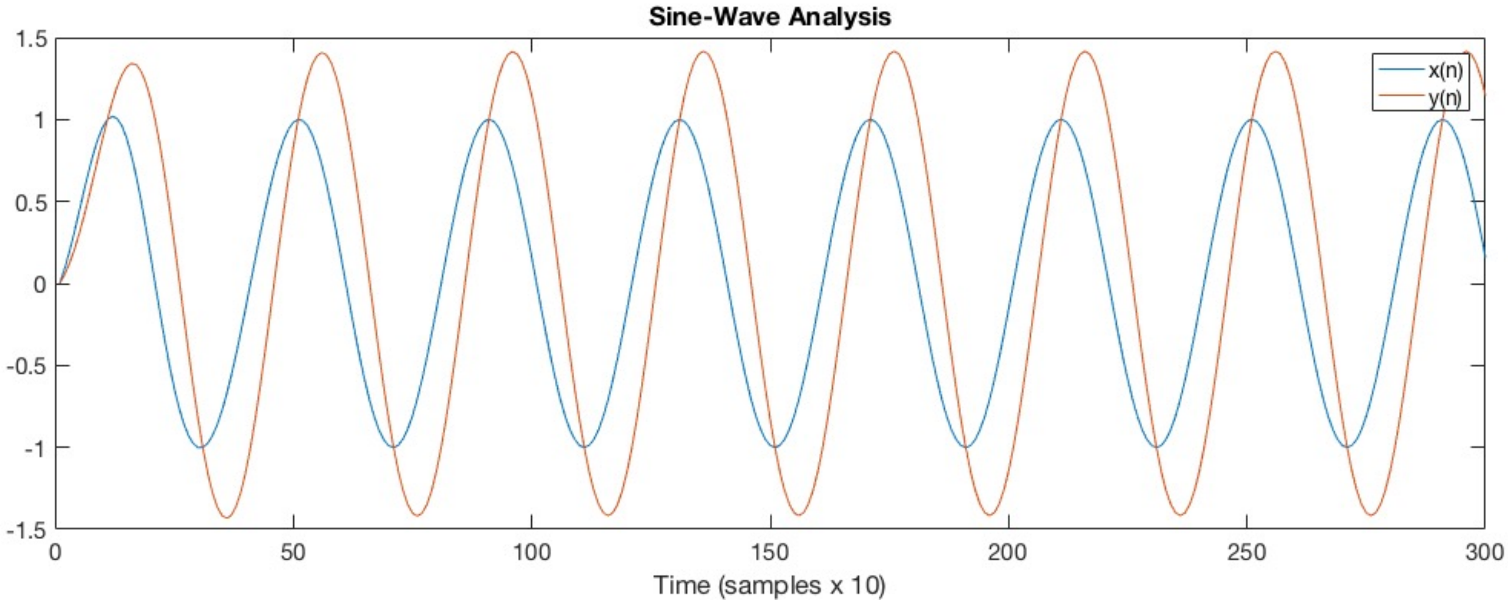
- Signal flow graph



"Delay Operator"

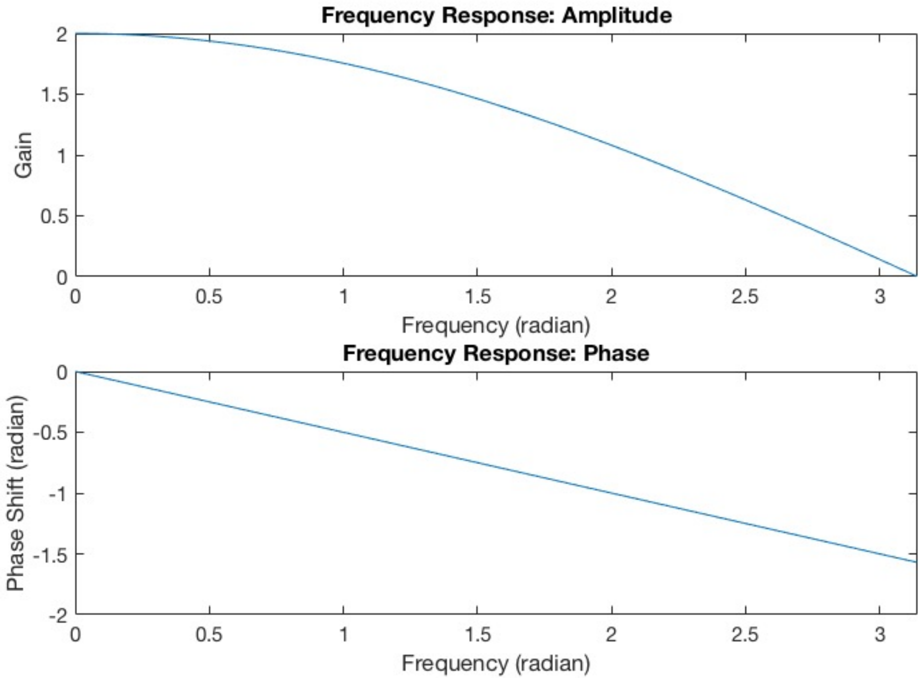
The Simplest Lowpass Filter: Sine-Wave Analysis

- Measure the amplitude and phase changes given a sinusoidal signal input



The Simplest Lowpass Filter: Frequency Response

- Plot the amplitude and phase change over different frequency
  - The frequency sweeps from 0 to the Nyquist rate



The Simplest Lowpass Filter: Frequency Response

- Mathematical approach
  - Use complex sinusoid as input: $x(n) = e^{j\omega n}$
  - Then, the output is:

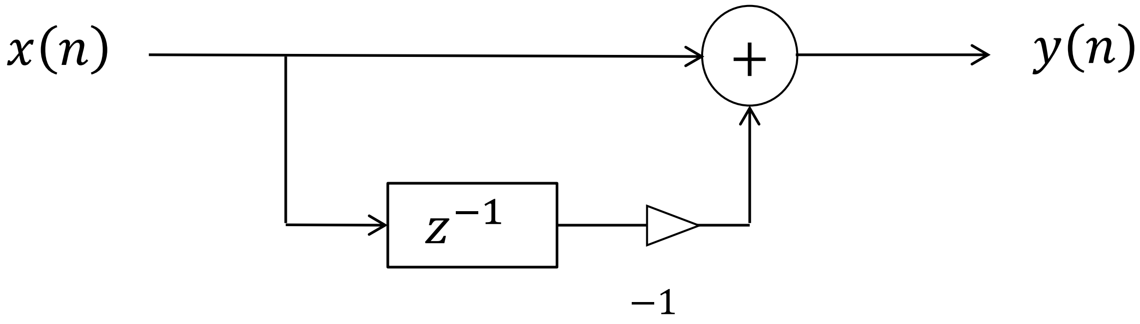
    $y(n) = x(n) + x(n-1) = e^{j\omega n} + e^{j\omega(n-1)} = \left(1 + e^{-j\omega}\right) \cdot e^{j\omega n}$

    $= \left(1 + e^{-j\omega}\right) \cdot x(n)$
  - Frequency response: $H(\omega) = \left(1 + e^{-j\omega}\right) = \left(e^{j\frac{\omega}{2}} + e^{-j\frac{\omega}{2}}\right) e^{-j\frac{\omega}{2}} = 2\cos(\frac{\omega}{2})e^{-j\frac{\omega}{2}}$
  - Amplitude response: $|H(\omega)| = 2\cos\left(\frac{\omega}{2}\right)$
  - Phase response: $\angle H(\omega) = -\frac{\omega}{2}$

The Simplest Highpass Filter

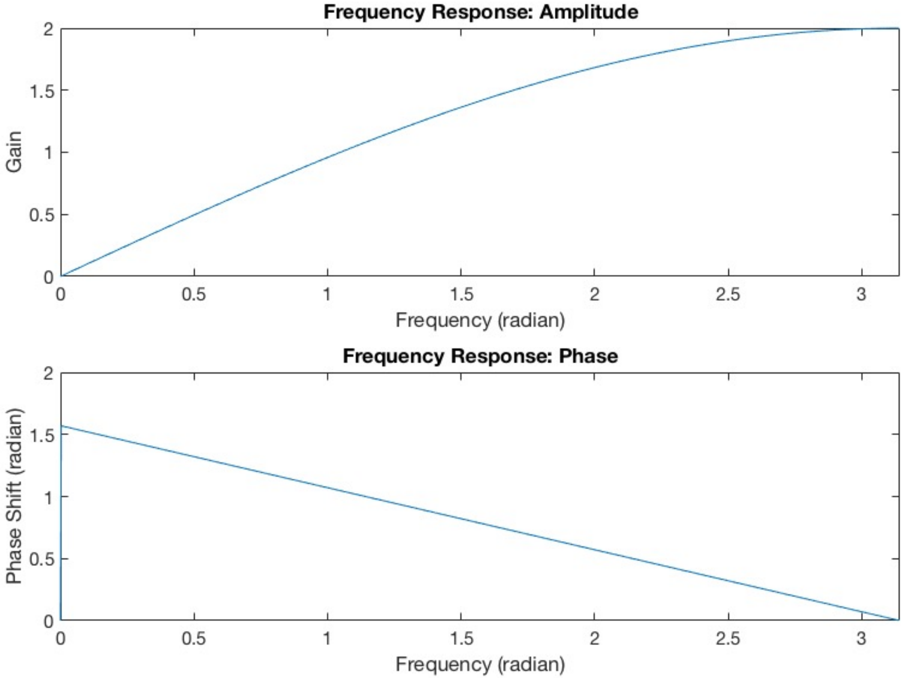
- Difference equation: $y(n) = x(n) - x(n-1)$

- Signal flow graph



The Simplest Highpass Filter: Frequency Response

- Plot the amplitude and phase change over different frequency
  - The frequency sweeps from 0 to the Nyquist rate

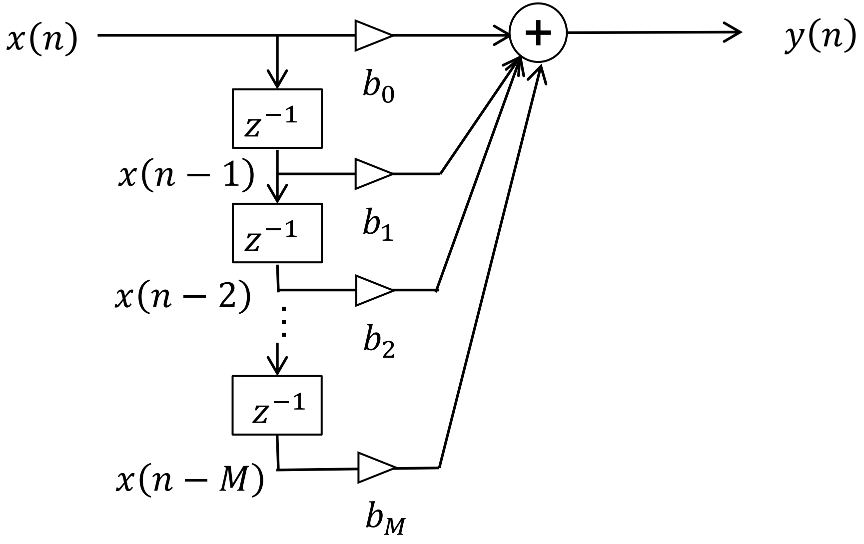


Finite Impulse Response (FIR) System

- Difference equation

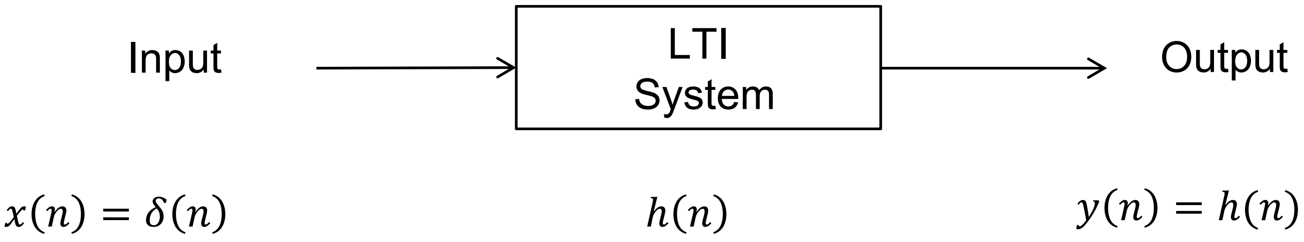
$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + b_2 \cdot x(n-2) + \cdots + b_M \cdot x(n-M)$$

- Signal flow graph



Impulse Response

- The system output when the input is a unit impulse
  - $x(n) = \delta(n) = [1, 0, 0, 0, \ldots] \rightarrow y(n) = h(n) = [b_0, b_1, b_2 \ldots, b_M]$ (for FIR system)
- Characterizes the digital system **as a sequence of numbers**
  - A system is represented just like audio samples!



Examples: Impulse Response

- The simplest lowpass filter
  - $\mathrm{h}(n) = [1, 1]$
- The simplest highpass filter
  - $\mathrm{h}(n) = [1, -1]$
- Moving-average filter (order=5)
  - $\mathrm{h}(n) = \left[\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right]$

Convolution

- The output of LTI digital filters is represented by the convolution operation between $x(n)$ and $h(n)$

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{\infty} x(i) \cdot h(n-i) = \boxed{\sum_{i=-\infty}^{\infty} h(i) \cdot x(n-i)}$$

This is more practical expression
when the input is an audio streaming

- Examples
  - The simplest lowpass filter
  - $y(n) = [1, 1] * x(n) = 1 \cdot x(n) + 1 \cdot x(n-1) = x(n) + x(n-1)$

Proof: Convolution

- Method 1
  - The input is represented as the sum of weighted and delayed impulses units
    - $x(n) = [x_0, x_1, x_2 \dots, x_M] = x_0 \cdot \delta(n) + x_1 \cdot \delta(n-1) + x_2 \cdot \delta(n-2) + \cdots + x_M \cdot \delta(n-M)$
  - By the linearity and time-invariance
    - $y(n) = x_0 \cdot h(n) + x_1 \cdot h(n-1) + x_2 \cdot h(n-2) + \cdots + x_M \cdot h(n-M) = \sum_{i=0}^{M} x(i) \cdot h(n-i)$

Proof: Convolution

- Method 2
  - The impulse response can be represented as a set of weighted impulses
    - $h(n) = [b_0, b_1, b_2 \dots, b_M] = b_0 \cdot \delta(n) + b_1 \cdot \delta(n-1) + b_2 \cdot \delta(n-2) + \cdots + b_M \cdot \delta(n-M)$
  - By the linearity, the distributive property and $x(n) * \delta(n-k) = x(n-k)$
    - $y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + b_2 \cdot x(n-2) + \cdots + b_M \cdot x(n-M) = \sum_{i=0}^{M} h(i) \cdot x(n-i)$

Properties of Convolution

- Commutative: $x(n) * h_1(n) * h_2(n) = x(n) * h_2(n) * h_1(n)$
- Associative: $(x(n) * h_1(n)) * h_2(n) = x(n) * (h_1(n) * h_2(n))$
- Distributive: $x(n) * \left(h_1(n) + h_2(n)\right) = x(n) * h_1(n) + x(n) * h_2(n)$

Example: Convolution

- Given $x(n) = [x_0, x_1, x_2, \dots, x_5]$ and $h(n) = [h_0, h_1, h_2\ ]$
  - $y(n) = \sum_{i=0}^{M} h(i) \cdot x(n-i)$
  - $y(0) = h(0) \cdot x(0)$
  - $y(1) = h(0) \cdot x(1) + h(1) \cdot x(0)$
  - $y(2) = h(0) \cdot x(2) + h(1) \cdot x(1) + h(2) \cdot x(0)$
  - $y(3) = h(0) \cdot x(3) + h(1) \cdot x(2) + h(2) \cdot x(1)$
  - $y(4) = h(0) \cdot x(4) + h(1) \cdot x(3) + h(2) \cdot x(2)$
  - $y(5) = h(0) \cdot x(5) + h(1) \cdot x(4) + h(2) \cdot x(3)$
  - $y(6) = \qquad h(1) \cdot x(5) + h(2) \cdot x(4)$
  - $y(7) = \qquad h(2) \cdot x(5)$

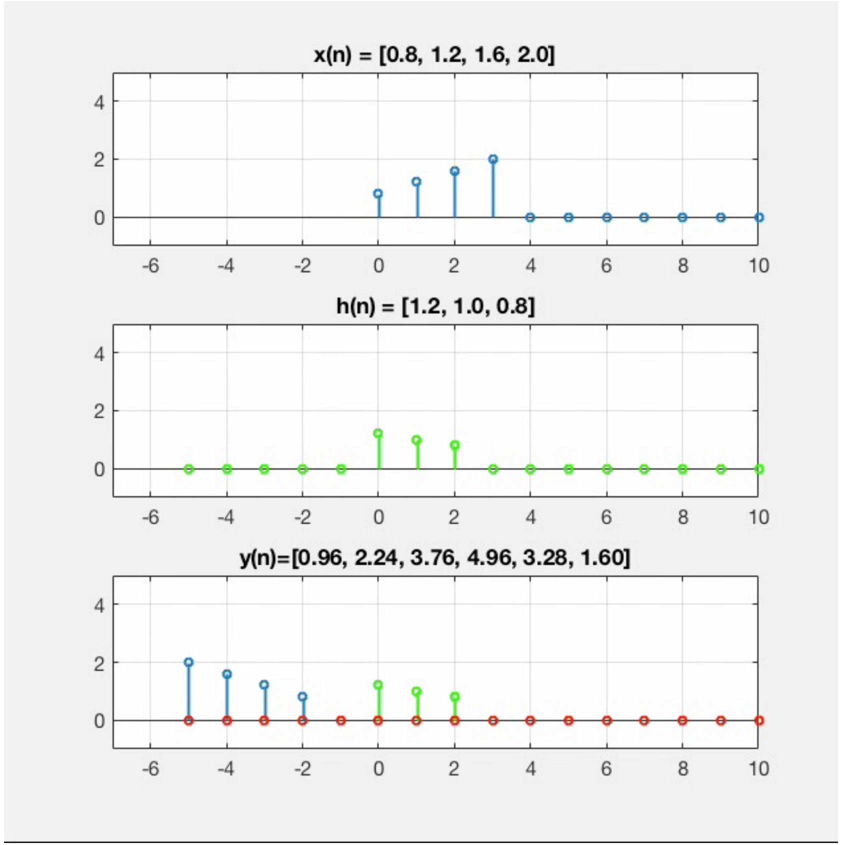
Transient state ($y(0)$–$y(1)$)

Steady state ($y(2)$–$y(5)$)

Transient state ($y(6)$–$y(7)$)

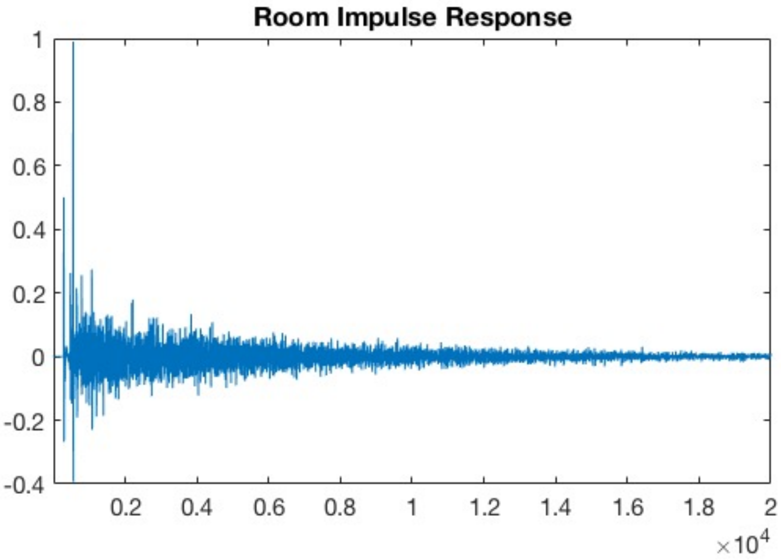
- The size of transient region is equal to the number of delay operators
- If the length of $x(n)$ is M and the length of $h(n)$ is N, then the length of $y(n)$is $M + N - 1$.

Demo: Convolution



Example: Convolution Reverb

- Convolution Reverb



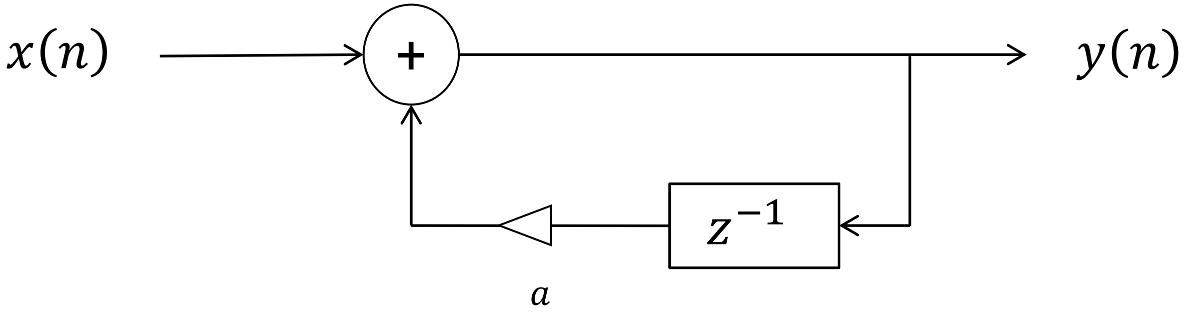
Room Impulse Response

Lobby

Church

A Simple Feedback Lowpass Filter

- Difference equation: $y(n) = x(n) + a \cdot y(n-1)$
- Signal flow graph
  - When a is slightly less than 1, it is called “Leaky Integrator”



A Simple Feedback Lowpass Filter: Impulse Response

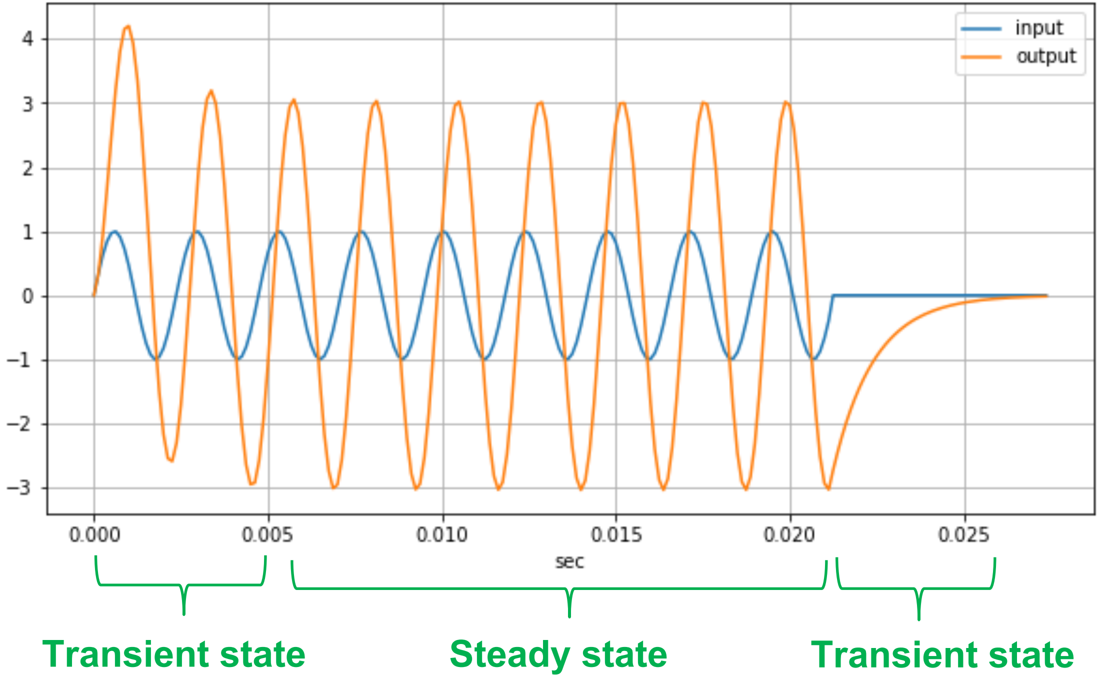
- Impulse response: exponential decays
  - $y(0) = x(0) = 1$
  - $y(1) = x(1) + a \cdot y(0) = a$
  - $y(2) = x(2) + a \cdot y(1) = a^2$
  - $y(3) = x(3) + a \cdot y(2) = a^2$
  - …

$\longrightarrow$ $h(n)$=$[1, a, a^2, a^3, \dots]$

- **Stability Issue!**
  - If $a < 1$, the filter output converges (stable)
  - If $a = 1$, the filter output oscillates (critical)
  - If $a > 1$, the filter output diverges (unstable)

A Simple Feedback Lowpass Filter: Sine-Wave Analysis

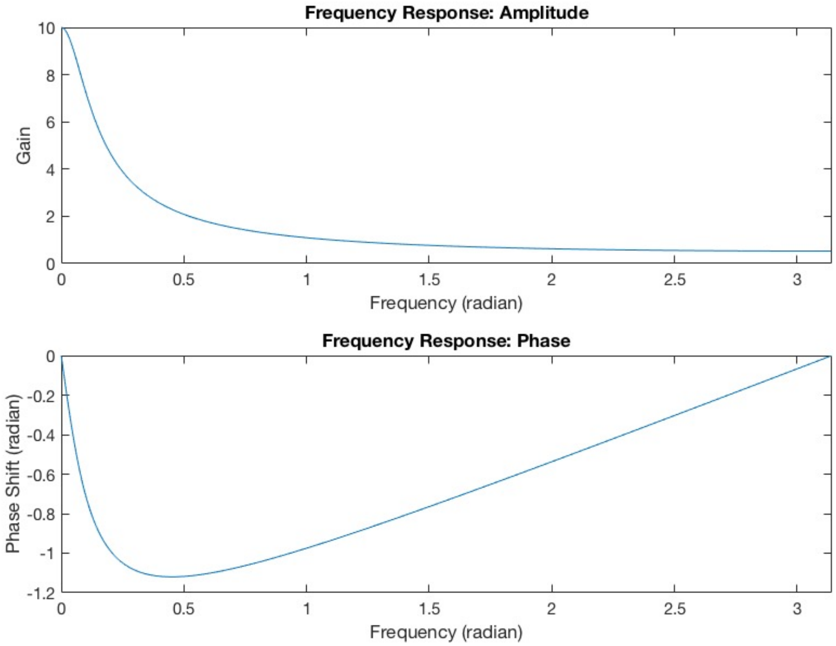
- Measure the amplitude and phase changes given a sinusoidal signal input



$a = 0.9$

A Simple Feedback Lowpass Filter: Frequency Response

- More dramatic change than the simplest lowpass filter (FIR)
  - Phase response is not linear



$$y(n) = x(n) + 0.9 \cdot y(n-1)$$

A Simple Feedback Lowpass Filter: Frequency Response

- Mathematical approach
  - Use complex sinusoid as input: $x(n) = e^{j\omega n}$
  - $y(n) = G(\omega)e^{j(\omega n + \theta(\omega))} \rightarrow y(n-m) = e^{-j\omega m}y(n)$ for any m
  - The output is: $y(n) = x(n) + a \cdot y(n-1)$

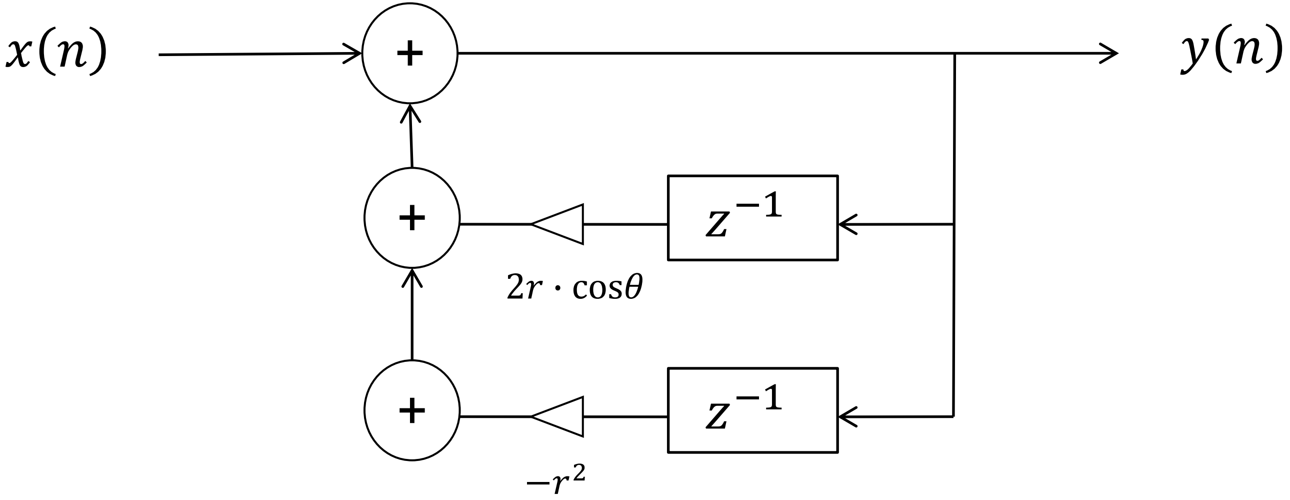
$$y(n) = x(n) + a \cdot e^{j\omega} y(n)$$

  - Frequency response: $H(\omega) = \frac{1}{(1 - a \cdot e^{-j\omega})} = \frac{1}{(1 - a \cdot \cos(\omega) + a \cdot j \cdot \sin(\omega))}$
  - Amplitude response: $|H(\omega)| = \frac{1}{(1 - a \cdot \cos(\omega))^2 + (a \cdot \sin(\omega))^2}$
  - Phase response: $\angle H(\omega) = -\operatorname{atan}(\frac{a \cdot \sin(\omega)}{1 - a \cdot \cos(\omega)})$
- Note that the this approach is getting complicated

Reson Filter

- Difference equation
  - $y(n) = x(n) + 2r \cdot \cos(\theta) \cdot y(n-1) - r^2 \cdot y(n-2)$

- Signal flow graph



Reson Filter

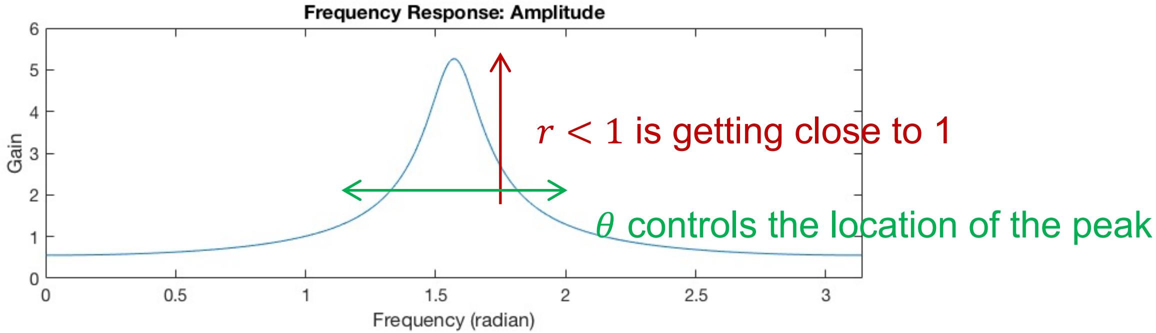
- Mathematical approach
  - Use complex sinusoid as input: $x(n) = e^{j\omega n}$
  - $y(n) = G(\omega)e^{j(\omega n+\theta(\omega))}$ → $y(n-m) = e^{-jm\omega}y(n)$ for any m
  - The output is: $y(n) = x(n) + 2r \cdot \cos(\theta) \cdot y(n-1) - r^2 \cdot y(n-2)$

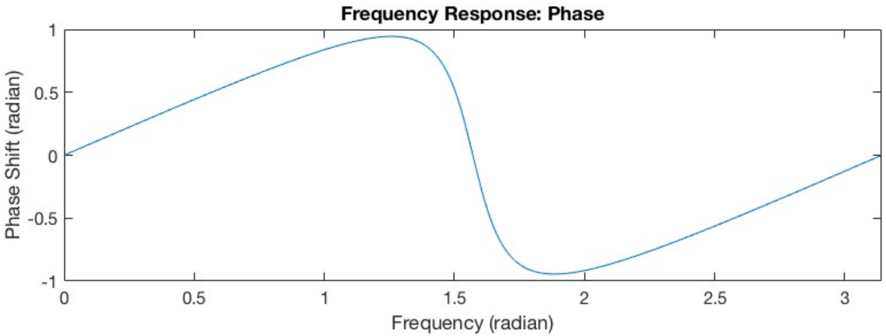
    $$y(n) = x(n) + 2r \cdot \cos(\theta) \cdot e^{j\omega}y(n) - r^2 \cdot e^{j2\omega}y(n-2)$$
  - Frequency response
    - $H(\omega) = \frac{1}{(1-2r\cdot\cos(\theta)\cdot e^{j\omega}+r^2\cdot e^{j2\omega})}$

      $$= \frac{1}{(1-r(\cos(\theta)+j\cdot\sin(\theta))e^{j\omega})(1-r(\cos(\theta)-j\cdot\sin(\theta))e^{j\omega})}$$
- Now you see that the this approach is getting even more complicated
  - We will introduce more intuitive method to obtain the frequency response

Reson Filter: Frequency Response

- Generate resonance at a particular frequency
  - Control the peak height by r and the peak frequency by θ



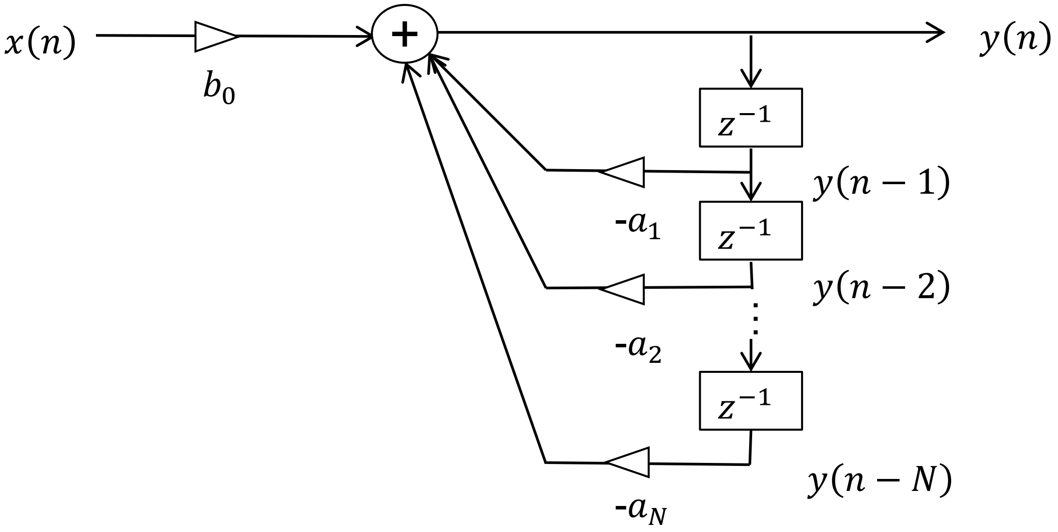


For stability: $r < 1$

Infinite Impulse Response (IIR) Filters

- Difference equation
    - $y(n) = b_0 \cdot x(n) - a_1 \cdot y(n-1) - a_2 \cdot y(n-2) - \cdots - a_N \cdot y(n-N)$

- Signal flow graph



General Form of LTI Filters

- Difference equation
  - $y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + b_2 \cdot x(n-2) + \ldots + \; b_M \cdot x(n-M)$
    $-a_1 \cdot y(n-1) - a_2 \cdot y(n-2) - \cdots - a_N \cdot y(n-N)$
- Signal flow graph