Linear time invariant systems

Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
http://www.seas.upenn.edu/users/~aribeiro/

February 25, 2016

Linear time invariant systems

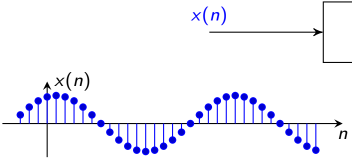
Finite impulse response filter design

Fourier transform and convolution

- Fourier transform enables signal and information processing
  - ⇒ Patterns and properties easier to discern on frequency domain
- Also enables analysis and deign of linear time invariant (LTI) systems
  - ⇒ Not altogether unrelated to pattern discernibility
- Two properties of LTI systems
  - ⇒ Characterized by their (impulse) response to a delta input
  - ⇒ Responses to other inputs are convolutions with impulse response
- Equivalent properties in the frequency domain
  - ⇒ Characterized by frequency response $= \mathcal{F}$(impulse response)
  - ⇒ Output spectrum $=$ input spectrum $\times$ frequency response

Systems

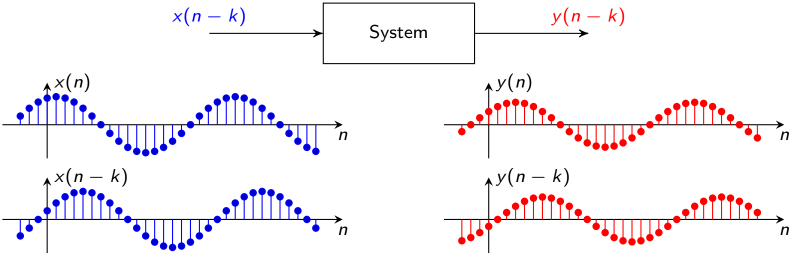
- A system is characterized by an input ($x(n)$) output ($y(n)$) relation
- This relation is between functions, not values
- Each output value $y(n)$ depends on all input values $x(n)$



- We can, alternatively, consider continuous time systems. The same.

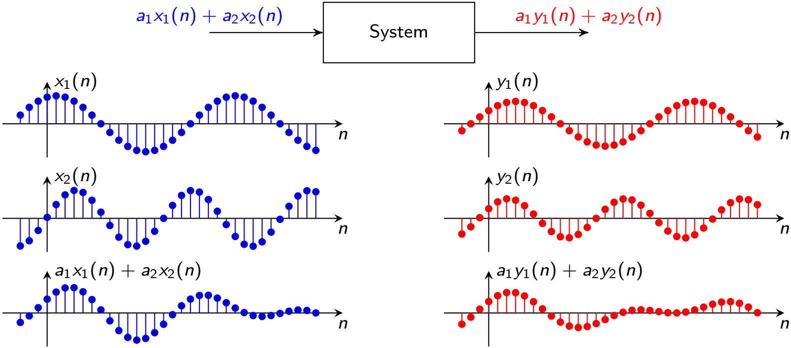
Time invariant systems

- A system is time invariant if a delayed input yields a delayed output
- If input $x(n)$ yields output $y(n)$ then input $x(n-k)$ yields $y(n-k)$
- Think of output when input is applied k time units later

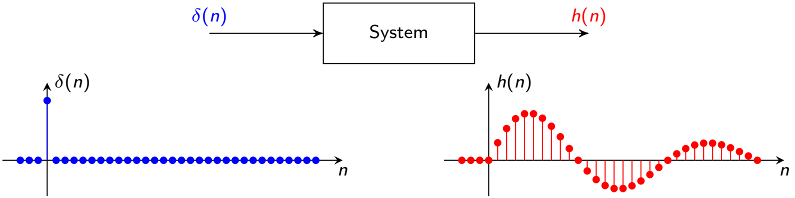


Linear systems

- In a linear system $\Rightarrow$ input a linear combination of inputs
  $\Rightarrow$ Output the same linear combination of the respective outputs
- I.e., if input $x_1(n)$ yields output $y_1(n)$ and $x_2(n)$ yields $y_2(n)$
  $\Rightarrow$ Input $a_1x_1(n) + a_2x_2(n)$ yields output $a_1y_1(n) + a_2y_2(n)$



- Linear + time invariant system = linear time invariant system (LTI)
- Also called a LTI filter, or a linear filter, or simply a filter
- The impulse response is the output when input is a delta function
  - $\Rightarrow$ Input is $x(n) = \delta(n)$ (discrete time, $\delta(0) = 1$)
  - $\Rightarrow$ Output is $y(n) = h(n)$ = impulse response



Scale and shifted impulse responses

- Since the system is time invariant (shift)

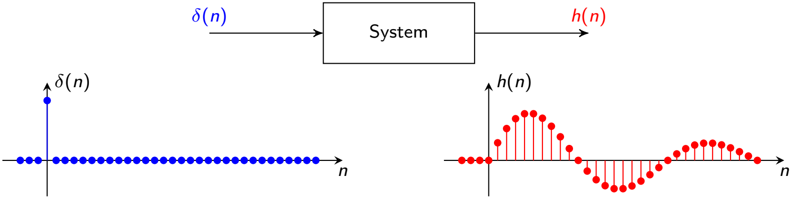
  $\Rightarrow$ Input $\delta(n-k)$ $\Rightarrow$ Induces output response $h(n-k)$

- Since the system is linear (scale)

  $\Rightarrow$ input $x(k)\delta(n-k)$ $\Rightarrow$ Output $x(k)h(n-k)$

- Since the system is linear (sum)

  $\Rightarrow$ $x(k_1)\delta(n-k_1)+x(k_2)\delta(n-k_2)$ $\Rightarrow$ $x(k_1)h(n-k_1)+x(k_2)h(n-k_2)$



Scale and shifted impulse responses

- Since the system is time invariant (shift)

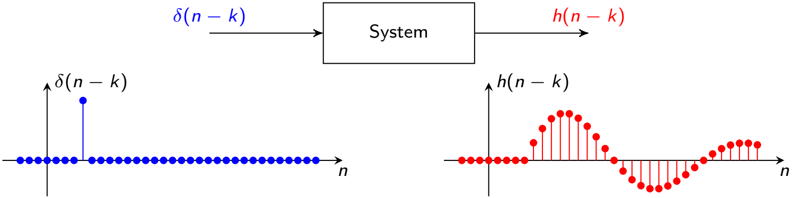
  $\Rightarrow$ Input $\delta(n-k)$ $\Rightarrow$ Induces output response $h(n-k)$

- Since the system is linear (scale)

  $\Rightarrow$ input $x(k)\delta(n-k)$ $\Rightarrow$ Output $x(k)h(n-k)$

- Since the system is linear (sum)

  $\Rightarrow x(k_1)\delta(n-k_1)+x(k_2)\delta(n-k_2)$ $\Rightarrow x(k_1)h(n-k_1)+x(k_2)h(n-k_2)$



Scale and shifted impulse responses

- Since the system is time invariant (shift)

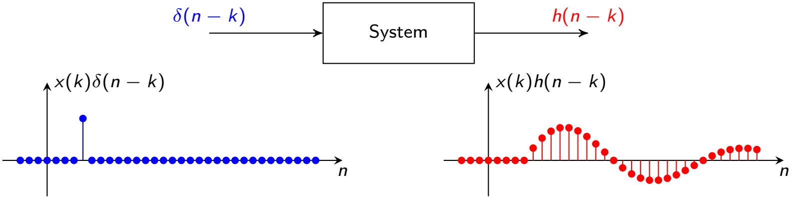
  $\Rightarrow$ Input $\delta(n-k)$ $\Rightarrow$ Induces output response $h(n-k)$

- Since the system is linear (scale)

  $\Rightarrow$ input $x(k)\delta(n-k)$ $\Rightarrow$ Output $x(k)h(n-k)$

- Since the system is linear (sum)

  $\Rightarrow x(k_1)\delta(n-k_1) + x(k_2)\delta(n-k_2)$ $\Rightarrow x(k_1)h(n-k_1) + x(k_2)h(n-k_2)$



Scale and shifted impulse responses

- Since the system is time invariant (shift)

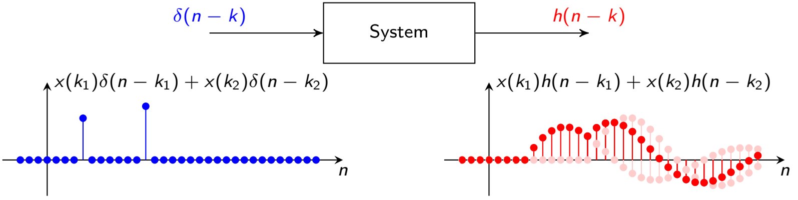
  $\Rightarrow$ Input $\delta(n-k)$ $\Rightarrow$ Induces output response $h(n-k)$

- Since the system is linear (scale)

  $\Rightarrow$ input $x(k)\delta(n-k)$ $\Rightarrow$ Output $x(k)h(n-k)$

- Since the system is linear (sum)

  $\Rightarrow$ $x(k_1)\delta(n-k_1) + x(k_2)\delta(n-k_2)$ $\Rightarrow$ $x(k_1)h(n-k_1) + x(k_2)h(n-k_2)$



Output of a linear time invariant system

- Shift, Scale, and Sum $\Rightarrow$ Is this a Convolution? $\Rightarrow$ Of course
- Can write any signal x as $\Rightarrow x(n) = \sum_{k=-\infty}^{+\infty} x(k)\delta(n-k)$
- Thus, output of LTI with impulse response h to input x is given by

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k)$$

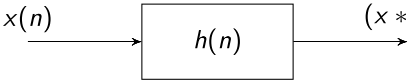
- The above sum is the convolution of x and h $\Rightarrow y = x * h$

Output of a linear time invariant system

Theorem

A linear time invariant system is completely determined by its impulse response h. In particular, the response to input x is the signal $y = x * h$.

- Innocent looking restrictions $\Rightarrow$ Linearity + time invariance
  $\Rightarrow$ Induce very strong structure (anything but innocent)



$$(x * h)(n) = \sum_{-\infty}^{\infty} x(k)h(t-k)$$

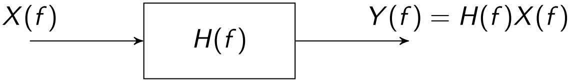
- Can derive exact same result for continuous time systems

Frequency response

- Frequency response = transform of impulse response $\Rightarrow H = \mathcal{F}(h)$

Corollary

A linear time invariant system is completely determined by its frequency response H. In particular, the response to input X is the signal $Y = HX$.



- Design in frequency $\Rightarrow$ Implement in time
  - $\Rightarrow$ Have done this already, but now we know its true for any LTI

Causality

- A causal filter is one with $h(n) = 0$ for all negative $n < 0$
  - $\Rightarrow$ Otherwise, we would respond to spike before seeing spike
- In general $\Rightarrow y(n) = \displaystyle\sum_{k=-\infty}^{+\infty} x(k)h(n-k) = \sum_{k=-\infty}^{n} x(k)h(n-k)$
- The value $y(n)$ is only affected by past inputs $x(k)$, with $k \leq n$
- If filter is not causal but $h(n) = 0$ for all $n < N$
  - $\Rightarrow$ Make it causal with a delay $\Rightarrow \tilde{h}(n) = h(n-N)$
- Frequency response of delayed filter $\Rightarrow \tilde{H}(f) = H(f)e^{j2\pi fN}$
  - $\Rightarrow$ Qualitatively the same filter

Finite impulse response

- A causal finite impulse response filter (FIR) is one for which

$$h(n) = 0 \qquad \text{for all } n \geq N$$

- We say the filter is of length N; only N values in $h(n)$ are not null
- Can write output at time n as

$$y(n) = h(0)x(n) + h(1)x(n-1) + \ldots h(N-1)x(n-N+1)$$

- Running input vector $\mathbf{x}_N(n) = [x(n); x(n-1); \ldots; x(n-N+1)]$
- FIR filter vector response $\mathbf{h} = [h(0), h(1), \ldots, h(N-1)]$
- Can then write output at time n as $\Rightarrow y(n) = \mathbf{h}^T \mathbf{x}_N$

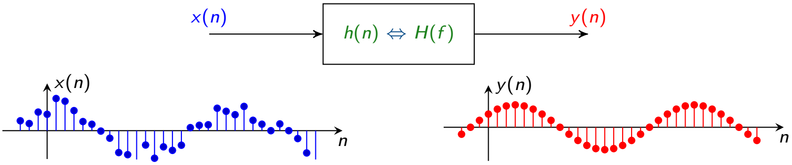
Finite impulse response filter design

Linear time invariant systems

Finite impulse response filter design

Filter design and implementation

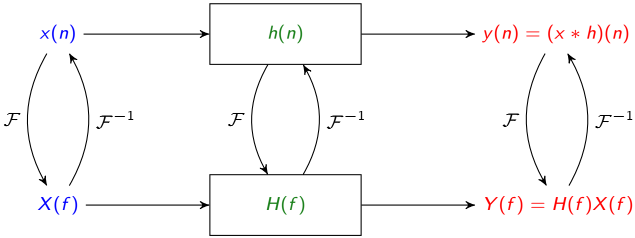
- We want to utilize a LTI system to process discrete time signal $x(n)$
  $\Rightarrow$ E.g., to smooth out the signal $x(n)$ shown below



- All LTIs are completely determined by their impulse responses h
  $\Rightarrow$ Design h and implement filter as time convolution $\Rightarrow y = x * h$
- All LTIs are completely determined by their frequency responses h
  $\Rightarrow$ Design H and implement filter as spectral product $\Rightarrow Y = HX$

Frequency design and time implementation

- Time and frequency representations are equivalent

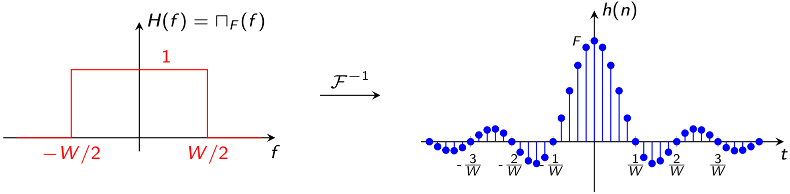


- Identify pattern transformation in frequency domain $\Rightarrow$ Design H
- Use inverse DTFT to compute impulse response $\Rightarrow h = \mathcal{F}^{-1}(H)$
- Implement convolution in time $\Rightarrow y(n) = (x * h)(n)$

Causality and infinite response

- Impulse response $h = \mathcal{F}^{-1}(H)$ is typically not causal and infinite
  - $\Rightarrow$ E.g., Low pass filter with cutoff freq. $W/2$ $\Rightarrow$ $H(f) = \sqcap_W(f)$

$$h(n) = \int_{-f_s/2}^{f_s/2} H(f) e^{j2\pi f n T_s}\, df = W \text{sinc}(\pi W n T_s)$$



- Multiply by window (chop) for finite response with N nonzero coeffs.
- Delay $h(n)$ to obtain a causal filter with $h(n) = 0$ for $n \leq 0$

- Transform $h(n)$ into finite impulse response

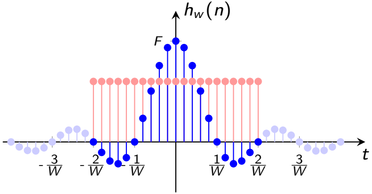
$$h_w(n) = h(n)w(n)$$

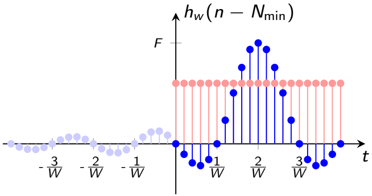
- Window $w(n) = 0$ for $n \notin [N_{\min}, N_{\max}]$
- Filter length $N = N_{\max} - N_{\min} + 1$

- Transform $h_w(n)$ into causal response

$$h_w(n) \implies h_w(n - N_{\min})$$

- Choose borders $N_{\min}$ and $N_{\max}$ to retain highest values of $h(n)$
- Often, around $n = 0$. But not always





Spectral effects of windowing and delaying

- Multiplication in time domain $\Rightarrow$ Convolution in frequency domain
- As a result, instead of filtering with $H(f)$, we filter with

$$H_w = H * W$$

- Choose windows with spectrum $W = \mathcal{F}(w)$ close to delta function
- Time delay $\Rightarrow$ Multiplication with complex exponential in frequency

$$H_w(f) \quad \Longrightarrow \quad H_w(f)e^{j2\pi f N_{\min} T_s}$$

- Irrelevant, as it should, we just shifted the response

FIR filter design methodology

- Procedure to design time coefficients of a FIR filter

(1) Spectral analysis to determine filter frequency response $H(f)$

(2) Inverse DFT (not DTFT) to determine impulse response $h(n)$

(3) Determine nr. of coefficients N and coefficient range $[N_{\text{min}}, N_{\text{max}}]$

(4) Select window $w(n)$ $\Rightarrow$ Alters spectrum to $H_w = H * W$

(5) Shift impulse response by N_{min} time steps to make filter causal

- How to we use FIR filter coefficients $h(n)$ to implement the filter?

FIR implementation

- The output $y(n)$ of the FIR filter is given by the convolution value

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

- Since h is finite and causal, only N nonzero terms. Make $k = n - l$

$$y(n) = \sum_{k=n-(N-1)}^{n} x(k)h(n-k) = \sum_{l=0}^{N-1} h(l)x(n-l)$$

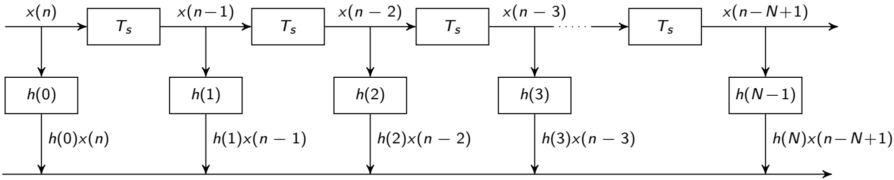
- Easier to visualize when written in expanded form

$$y(n) = h(0)\,x(n) + h(1)\,x(n-1) + \ldots + h(N-1)\,x(n-N+1)$$

- The expression above can be implemented with a shift register

Shift registers

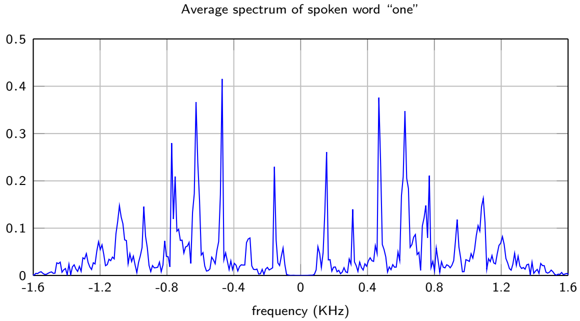
- Upon arrival of signal value $x(n)$ we compute output value $y(n)$ by
  - $\Rightarrow$ Delay (shift) units to shift elements of signal x
  - $\Rightarrow$ Product (scale) units to multiply with filter coefficients $x(n)$
  - $\Rightarrow$ Sum units to aggregate the products $h(k)x(n-k)$



- Shift register can be implemented in hardware (or software)

Voice recognition $\Rightarrow$ Spectral design

- For a given word to be recognized we compare the spectra $\bar{X}$ and X
  - $\Rightarrow \bar{X} \Rightarrow$ Average spectrum magnitude of word to be recognized
  - $\Rightarrow X \Rightarrow$ Recorded spectrum during execution time



- Made coparison with inner product $\Rightarrow X^T\bar{X}$
- Equivalent to using $\bar{X}$ to filter X $\Rightarrow Y(f) = H(f)X(f)$ with $H(f) = \bar{X}$

(2) Impulse response $h(n)$ $\Rightarrow$ Inverse DFT of $\bar{X}$

(4) Window to keep $N = 1,000$ largest consecutive taps