BME 50500: Image and Signal Processing in Biomedicine

Lecture 3: Linear Systems

Lucas C. Parra
Biomedical Engineering Department
City College of New York

CCNY

Linear Time Invariant System (LTI)

A transformation L: $\boldsymbol{y} = L[\boldsymbol{x}]$ is called **linear** if:

$$\boldsymbol{y} = L[a\,\boldsymbol{x}_1 + b\,\boldsymbol{x}_2] = a\,L[\boldsymbol{x}_1] + b\,L[\boldsymbol{x}_2]$$

A **linear system** is a functional transformation of time functions L: $y(t)$=L[$x(t)$] such that:

$$y(t) = L[a\,x_1(t) + b\,x_2(t)] = a\,L[x_1(t)] + b\,L[x_2(t)]$$

Note that in a linear system the current output at time t may be influenced by past or future inputs $x(t')$.

A linear system is called **time invariant** if:

$$y(t) = L[x(t)] \quad \Rightarrow \quad y(t+\tau) = L[x(t+\tau)]$$

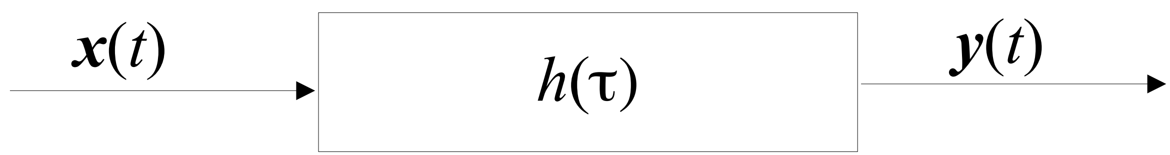
(**shift invariant** in the discrete time case)

Impulse response

A LTI system is fully characterized by the **impulse response** $h(\tau)$

$$y(t)=\int_{-\infty}^{\infty} d\tau\, h(\tau)\, x(t-\tau)$$

A LTI is represented as:

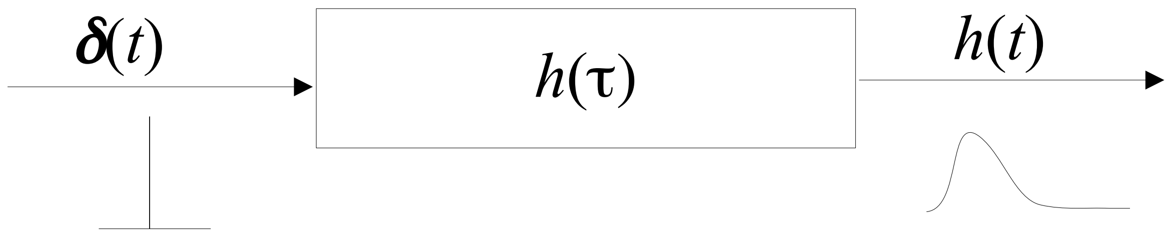


$h(\tau)$ is called impulse response because it is the system response to an input impulse:

$$x(t)=\delta(t)$$
$$y(t)=\int_{-\infty}^{\infty} d\tau\, h(\tau)\,\delta(t-\tau)=h(t)$$

Impulse response

Impulse response $h(\tau)$ can be measured using an unit impulse:

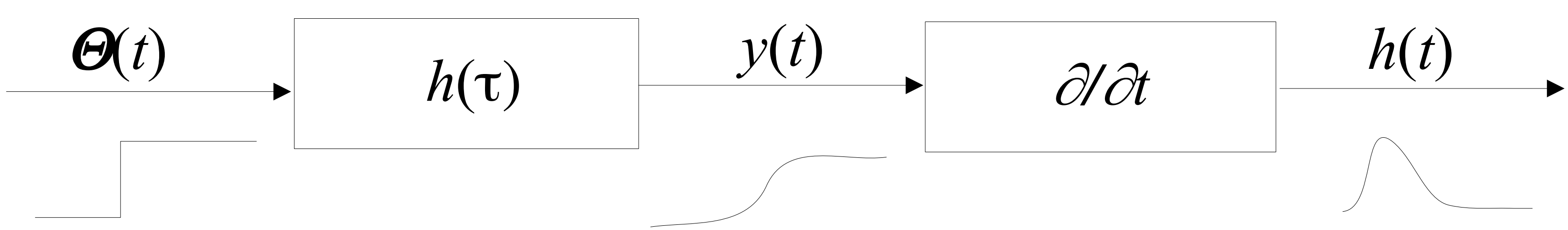


Also by differentiating the output to a step input:

$$x(t)=\Theta(t) \qquad \text{where} \qquad \delta(t)=\partial/\partial t\,\Theta(t)$$

$$\frac{\partial}{\partial t}y(t)=\int_{-\infty}^{\infty} d\tau\, h(\tau)\frac{\partial}{\partial t}\Theta(t-\tau)$$

$$=\int_{-\infty}^{\infty} d\tau\, h(\tau)\delta(t-\tau)=h(t)$$



Impulse response -discrete, causal, finite

In practice when implementing this digitally we have to make the following simplifications:

1. **Discrete:** *Approximate* integral with sum at discrete lags $\tau = k\,\Delta t$
Sample input and output at times $t = n\,\Delta t$:

$$\int d\tau\, h(\tau)\, x(t-\tau) = \sum_l h[l]\, x[n-l]$$

2. Assume **Causal:** Depends only on the past $h[l]=0,\ l < 0$:
3. Assume **Finite Impulse Response (FIR)**: $h[l]=0,\ P<l,\ P < \infty$

$$y[n] = \sum_{l=0}^{P-1} h[l]\, x[n-l]$$

Impulse response

Assignment 3A: Experiment with impulse response h

1. Record your voice using sd.rec() (import sounddevice as sd) and save it as a wav file (from scipy.io import wavfile as wav). Do this separately. Send with your homework only the subsequent code along with your recorded wav file.
2. Load a sound file using wav.read()
3. Display the sound signal (subplot #1).
4. Display the absolute value of its Fourier transform (subplot #2).
5. Select coefficients for h – the impulse response, a.k.a filter.
6. Filter the sound with this impulse response using np.convolve() (import numpy as np)
7. Display the absolute value of the Fourier transform (amplitude) of this filtered signal (subplot #3).
8. Go back to 4 and select new values until you achieve either a low-pass or high-pass filter, as judged by how much the Fourier transform of the filtered signal changed relative to the unfiltered signal. Show the amplitude of the Fourier transform for this filter. (subplot #4). You could also listen to the filtered signals to make a judgment on what your filter did (import sounddevice as sd; sd.play(x, fs)). Indicate whether your goal was a low-pass or a high-pass filter.

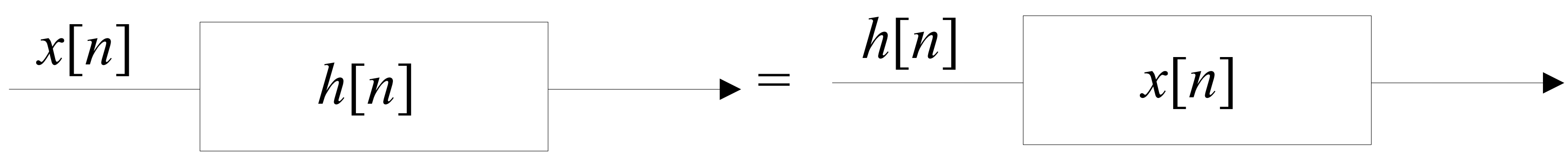
   In total there should be 4 plots on your figure. Put the 3 spectra next to each other and use the same vertical axes so they are easier to compare to each other. Be sure to properly label the time and frequency axes in seconds and Hertz respectively. Only show the positive frequencies and show amplitude on logarithmic scale. Submit your program and sound file.

Convolution

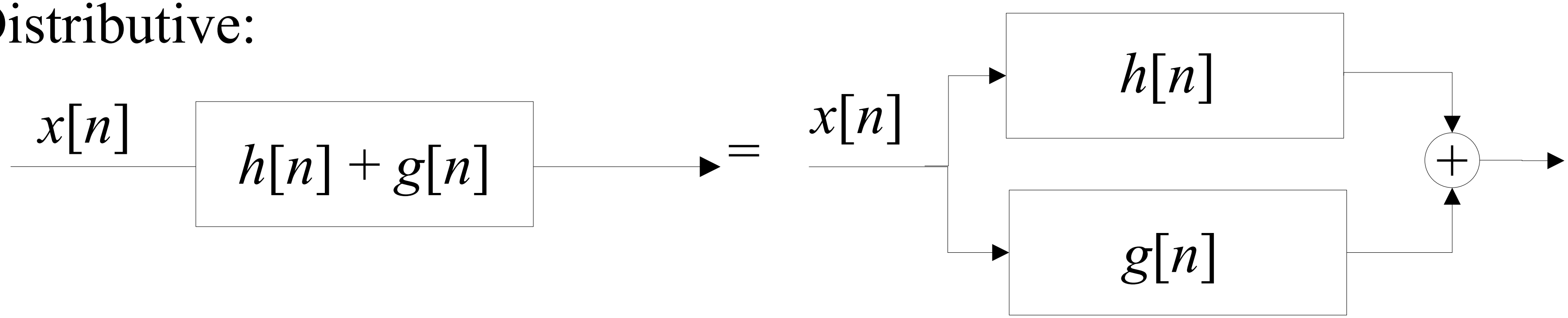
$$h[n] * x[n] = \sum_{l=-\infty}^{\infty} h[l]\,x[n-l]$$

Using this definition one can show the following properties:

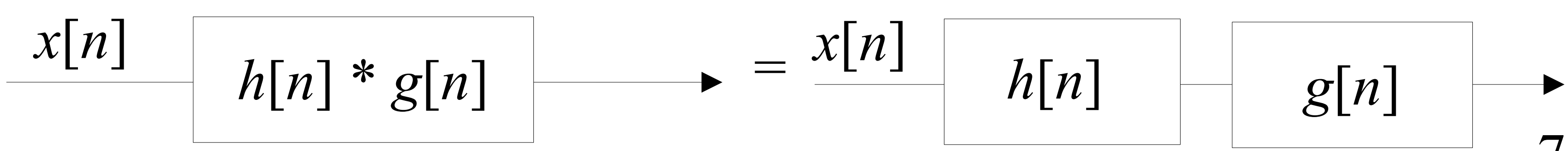
Commutative:



Distributive:



Associative:



Fourier Transform – Convolution Theorem

$$y(t)=h(t)*x(t) \quad \Leftrightarrow \quad Y(\nu)=H(\nu)X(\nu)$$

Because the Fourier transform of the convolution ...

$$Y(\nu)=FT[h(t)*x(t)]=\int_{-\infty}^{\infty} dt\, h(t)*x(t)\, e^{-i2\pi\nu t}=$$

$$=\int_{-\infty}^{\infty} dt \int_{-\infty}^{\infty} dt'\, h(t')\, x(t-t')\, e^{-i2\pi\nu t}$$

$$=\int_{-\infty}^{\infty} dt \int_{-\infty}^{\infty} dt'\, h(t')\, x(t)\, e^{-i2\pi\nu(t+t')}$$

$$=\int_{-\infty}^{\infty} dt'\, h(t')\, e^{-i2\pi\nu t'} \int_{-\infty}^{\infty} dt\, x(t)\, e^{-i2\pi\nu t}$$

$$=H(\nu)\,X(\nu)$$

Fourier Transform – Convolution Theorem

Note that with the convolution theorem we can implement convolution as a multiplication in the frequency domain.

$$y(t) = h(t) * x(t) \Leftrightarrow Y(\nu) = H(\nu) X(\nu)$$

$$\begin{tikzcd}
h(t) \arrow[r, "FT"] & H(\nu) \arrow[dr] & & & \\
& & \otimes \arrow[r] & Y(\nu) \arrow[r, "FT^{-1}"] & y(t) \\
x(t) \arrow[r, "FT"] & X(\nu) \arrow[ur] & & &
\end{tikzcd}$$

Impulse response

Assignment: Measure the impulse response of an RC circuit.

- Build an RC circuit with any resistor and capacitor. Apply at rectangle waveform at the input and measure the response at the output. Also measure the input. Make sure you pick a reasonable sampling rate and a frequency for the waveform so that you observe the step response.
- Measures the step response (at least 10 repeats) and save it to a CSV file.
- Load the data into python. You can use pandas read_csv().
- Calculate the temporal derivative to obtain the impulse response.
- Arrange the 10 repeats into a matrix and then average across the 10 repeats. You may use np.diff(), np.nonzero(), “>” and astype(int) to convert the boolean into interger numbers.
- Display your results on a graph showing the impulse response as a function of time. Include axis labels, and units.
- Repeat for output measures across the resistor and across the capacitor.

Impulse response

Assignment 3C: Measure the impulse response of an acoustic system.

- Use sd.playrec() function (import sounddevice as sd) to play and record sound at the same time. Be sure that speaker and microphones are close enough (loud enough) so that you record the sound coming out of the speakers.
- Make repeated measures of the impulse response (at least 20). This can be done by playing 20 impulses with enough spacing between them so the recorded signals generated by neighboring pulses does not overlap.
- Take the average of the repeated impulse responses measures.
- Estimate the amount of noise of this impulse response for each sample.
- Display your results on a graph showing the impulse response as a function of time. Include error bars, axis labels, and time units.
- Save the figure as png image, and send a copy of the figure and code.

Fourier Transform - Inverse Filter

With the Convolution Theorem we can derive the inverse convolution (or inverse filter)

$$y(t)=h(t)*x(t) \Leftrightarrow Y(\nu)=H(\nu)X(\nu)$$

Therefore

$$X(\nu)=\frac{Y(\nu)}{H(\nu)}$$

And the inverse filter is given by the inverse FT of $H^{-1}(\nu)$:

$$x(t)=FT^{-1}\left[\frac{1}{H(\nu)}\right]*y(t)=h_{inv}(t)*y(t)$$

Discrete Fourier Transform - FFT

In signal processing we always work with the DFT since we can compute Frourier transform only for discrete frequencies.

Important result on computational cost: While computing DFT values $X[k]$, $k=1 \ldots N$, would seem to take N^2 operations there is an efficient method called **Fast Fourier Transform** (FFT) of order:

$$\boldsymbol{N \log_2 N}$$

```
>> X =  fft(x);
>> x = ifft(X);
```

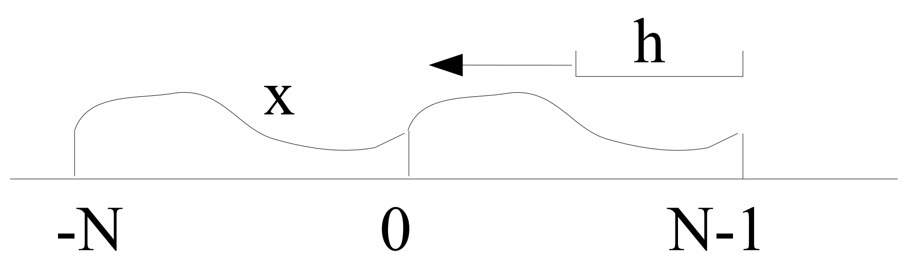
With this one can implement convolution in $\mathbf{log_2(\boldsymbol{P})}$ operations per sample rather than $\boldsymbol{P}$!

DFT - circular convolution

Because $X[k]$ corresponds to a periodic $x[n]$ with period N the convolution of two signals is equivalent to a **circular convolution**:

$$h[n] \circ x[n] = \sum_{k=0}^{N-1} h[k]\, x[(n-k) \bmod N]$$

That is, the circular convolution "wraps around"



It can be implemented with a circular Toeplitz matrix:

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[N-1] \end{bmatrix} = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ h[2] & h[1] & h[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix}$$

DFT - circular convolution

The convolution theorem for the DFT corresponds now to a **circular convolution:**

$$y[n] = h[n] \circ x[n] \quad \Leftrightarrow \quad Y[k] = H[k] X[k]$$

We can use this for a fast implement the linear convolution

$$y[n] = h[n] * x[n]$$

Fourier Transform - Inverse Filter

Assignment 4:

- Generate a random signal $x[n]$ with n=1...N. (N is a power of 2)
- Filter it with h=[1; -0.8;0.5] to generate $y = h*x$;
- Recover the signal from y with the inverse filter implemented in the Fourier domain. Show the original and recovered x in a single graph.
- Show the impulse response of the inverse filter.
- Recover the signal by convolving with the inverse filter in the time domain. Again compare the results in a single graph.