BME 50500: Image and Signal Processing in Biomedicine

Lecture 3: Linear Systems

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

http://bme.ccny.cuny.edu/faculty/parra/teaching/signal-and-image/
parra@ccny.cuny.edu
Linear Time Invariant System (LTI)

A transformation $L: y = L[x]$ is called **linear** if:

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

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:

![Diagram](image)

$h(\tau)$ 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:

\[
\delta(t) \rightarrow h(\tau) \rightarrow h(t)
\]

Also by differentiating the output to a step input:

\[
x(t) = \Theta(t) \quad \text{where} \quad \delta(t) = \partial / \partial t \Theta(t)
\]

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

\[
= \int_{-\infty}^{\infty} d\tau h(\tau) \delta(t-\tau) = h(t)
\]
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 $t = n \Delta t$:
   
   $$\int d\tau h(\tau) x(t-\tau) = \sum h[l] x[n-l]$$

2. Assume **Causal:** Depends only on the past $h[l]=0$, $l < 0$:

3. Assume **Finite Impulse Response (FIR):** $h[l]=0$, $l > P < \infty$

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

Assignment 3A: Experiment with impulse response $h$

1. Load a sound file using `wavread()`
2. Display the sound signal.
3. Display the absolute value of its Fourier.
4. Select coefficients for filter $h$.
5. Filter the sound with this impulse response using `conv()` or `filter()`
6. Display the absolute value of the Fourier transform (amplitude) of this filtered signal.
7. 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.

In total there should be 4 plots on your figure. 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. Use 'grid on' on that frequency display. 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:

\[ x[n] \quad h[n] \quad = \quad h[n] \quad x[n] \]

Distributive:

\[ x[n] \quad h[n] + g[n] \quad = \quad x[n] \quad h[n] \quad g[n] \]

Associative:

\[ x[n] \quad h[n] * g[n] \quad = \quad x[n] \quad h[n] \quad g[n] \]
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 ...

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

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

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

\[ = \int dt' h(t') e^{-i2\pi \nu t'} \int_{-\infty}^{\infty} dt \ x(t) e^{-i2\pi \nu t} \]

\[ = H(\nu) X(\nu) \]
Fourier Transform – Convolution Theorem

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

\[ y(t) = h(t) \ast x(t) \iff Y(\nu) = H(\nu) \cdot X(\nu) \]
Impulse response

Assignment 3B: Measure the impulse response of a real system.

• Chose any electronic circuit that contains at the very least a capacitor or inductor and a resistor. Preferably use a circuit that performs some form of filtering, and use a function generator as input. For instance you can measure data at the input and output of the ECG device that you developed in the design course. Work in teams so that everyone has access to data. Each student should use different data recordings. Make sure you pick a reasonable sampling rate.
• Make repeated measures of the impulse response (at least 10) to estimate the amount of noise and obtain the average impulse response over those repeated measures.
• Display your results on a graph showing the impulse response as a function of time. Include error bars, axis labels, and units.
Fourier Transform - Inverse Filter

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

\[ y(t) = h(t) * x(t) \iff Y(\nu) = H(\nu) X(\nu) \]

Therefore

\[ X(\nu) = \frac{Y(\nu)}{H(\nu)} \]

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

\[ x(t) = FT^{-1}\left[ \frac{1}{H(\nu)} \right] \ast y(t) = h_{inv}(t) \ast y(t) \]
Discrete Fourier Transform - FFT

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

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

$$N \log_2 N$$

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

With this one can implement convolution in $\log_2(P)$ operations per sample rather than $P$!
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**:

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

That is, the circular convolution "wraps around"

![Circular convolution diagram]

It can be implemented with a circular Toeplitz matrix:

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

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

\[ y[n] = h[n] \circ x[n] \iff Y[k] = H[k] X[k] \]

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.