

Convolution and some of its applications

DSP Lab 08

Table of contents

1	Objective	1
2	Theoretical aspects	1
2.1	Convolution operation	1
2.2	Convolution in Matlab	2
2.3	Echo (reverb) in sound processing	2
3	Exercises	3
4	Final questions	4

1 Objective

Students should understand the convolution equation, and how to use convolution in Matlab.

2 Theoretical aspects

2.1 Convolution operation

For two signals $x[n]$ and $h[n]$, the **convolution** operation is defined as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

When one signal is the input to a LTI system, and the other signal is the impulse response of the system, **the output of the system is the convolution between the input and the impulse response.**

The order of the two signals does not matter. One of them must have the indices $[k]$ and the other one $[n - k]$

Properties of convolution:

- it is commutative
- it is associative
- convolution with $\delta[n]$ leaves the signal unchanged

LTI systems interconnection:

- systems connected in series \Leftrightarrow convolution of their impulse responses
- systems connected in parallel \Leftrightarrow sum of their impulse responses

2.2 Convolution in Matlab

Convolution of two vectors in Matlab is achieved with the function `conv()`:

```
a = [1, 2, 3, 4, 5];  
b = [2, 1, 0, 3];  
y = conv(a,b)           % Convolution of vectors a and b
```

2.3 Echo (reverb) in sound processing

One of the prominent applications of convolution is in music/sound effects. As an example, consider an **echo** (or **reverberation** or **reverb** effect).

In acoustics, an echo is similar to a system in the following way: - you emit an impulse, e.g. a short sound (this is the input impulse) - you receive back multiple reflections of your sound, with various delays and decaying amplitudes, as the signal is reflected and bounces back (this is the impulse response, i.e. the response of the system to your impulse)

There are techniques to actually measure the **impulse response of a room** (a cathedral, a stadium etc), which characterizes the reverberation effect of that space.

Then one can transplant reverberation effect over a sound signal by performing a convolution between the sound signal and the impulse response which characterizes that particular room or open space.

3 Exercises

1. Load an audio signal and extract an 100000-long sequence of it. Convolve the sequence with the impulse response $\{1/6, 1/6, 1/6, 1/6, 1/6, 1/6\}$, using the `conv()` function from Matlab.
 - a. Play the resulting sequence and compare with the original.
 - b. What type of filter do we implement here? (discussion)
2. Download the “IM Reverbs Pack” archive from <http://www.voxengo.com/impulses/>. It contains impulse responses that create a reverberation effect.
 - a. Unzip and play the file “Scala Milan Opera Hall.wav”.
 - b. Load file “Scala Milan Opera Hall.wav” in Matlab (use `audioread()`). Restrict the data to about 1 second length. Call the resulting vector **h**.
 - c. Load the first 4 seconds of “Kalimba.mp3” (use `audioread()`), convolve with **h** and play the result (use `sound()` or `audioplayer()`).

How does the signal sound? What audio effect did we implement here?

Other sites for impulses responses:

- <http://pcfarina.eng.unipr.it/Public/IMP-RESP/>
 - <https://www.kvraudio.com/forum/viewtopic.php?t=107337&start=645>
3. Load the first 4 seconds of “Kalimba.mp3” (use `audioread()`), convolve first with one impulse response, and then convolve the result with a second impulse response from the pack.
 - a. Play the resulting signal. How does the signal sound?
 - b. Compute and display the equivalent impulse response of the complete system
 4. Implement a simpler echo scheme which just replicates the same sound a few times, with different delays and attenuations: (<https://www.dsprelated.com/showcode/41.php>). The same sound should be repeated three times:
 - after 0.05 seconds, with attenuation 1
 - after 0.1 seconds, with attenuation 0.9
 - after 0.14 seconds, with attenuation 0.75

What is the corresponding impulse response **h**? Find **h** and then apply the echo with `conv()`.

5. Implement the same scheme in Simulink

6. Check the length of the convolution result vector, and deduce the general rule: what is length of the convolution of two signals of lengths L_1 and L_2 ?
7. Implement a Matlab function `y = myconv(x,h)` which implements convolution. The function is given two input vectors and outputs the resulting vector.

4 Final questions

1. TBD