Assignment 1

Sachinkumar Dubey
EE20MTECH11009

08 September 2020

1 Introduction

In this assignment, we have performed the basic signal processing operations, which are as follow:

1. Convolution
2. Correlation
3. Down-sampling
4. Up-sampling

Each of the above operation has been demonstrated by writing C programs. For ease of calculation, discrete time signals have been considered as the inputs to these operators. However, these operations can also be performed on continuous time signals. The following sections contain information regarding each of these operations.

2 Convolution

In simple terms, Convolution is a mathematical operation, which take two signals and produces a third signal which is output. In linear systems, It defines the relation between the input and the output signal. The output signal is said to be convolution of input signal and the impulse response of the system.

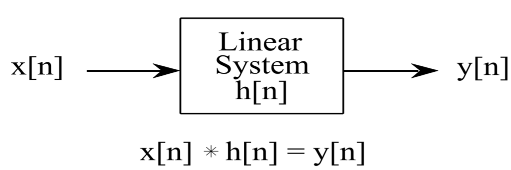

Figure 1: Convolution

Convolution for two discrete time signals is defined as:

$$y[n] = \sum_{k=0}^{\infty} x(k)h(n-k) \tag{1}$$

Graphically, one of the signals say $h[n]$ is reflected and then slided along the second signal $x[n]$ by adding discrete time offsets. In the overlap region, The corresponding samples are multiplied with each other and sum of those products is taken to find the output signal value at that particular time offset.

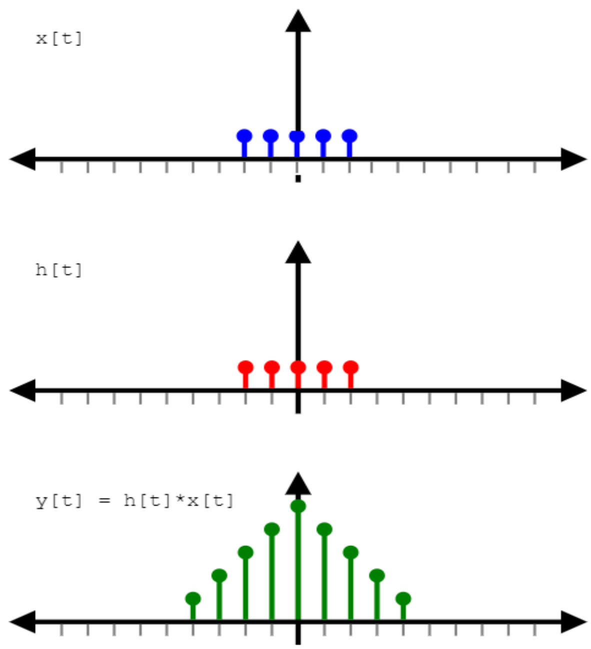

Figure 2: Convolution of two discrete time signals

Convolution of two continuous time signals $x(t)$ and $h(t)$ is given by:

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) + h(t-\tau)d\tau \tag{2}$$

Properties of Convolution are as follow:

1. Commutative Property : $x_1(t) * x_2(t) = x_2(t) * x_1(t)$
2. Distributive Property : $x_1(t)*[x_2(t)+x_3(t)] = [x_1(t)*x_2(t)]+[x_1(t)*x_3(t)]$
3. Associative Property : $x_1(t) * [x_2(t) * x_3(t)] = [x_1(t) * x_2(t)] * x_3(t)$

4. Shifting property :
$x_1(t) * x_2(t) = y(t)$
$x_1(t) * x_2(t - t_0) = y(t - t_0)$
$x_1(t - t_0) * x_2(t) = y(t - t_0)$
$x_1(t - t_0) * x_2(t - t_1) = y(t - t_0 - t_1)$

5. Convolution with Impulse : $x_1(t) * \delta(t - t_0) = x(t - t_0)$

6. Scaling Property : If $x(t) * h(t) = y(t)$ then $x(at) * h(at) = \frac{1}{|a|} y(at)$

Few applications of Convolution are as follow:

1. Finding output of LTI systems

2. Image and signal Processing

3. Probability and statistics

3 Correlation

Correlation of two signals gives the similarity between the signals when one of the signal is slided over the other.Graphically, one of the signal is slided along the other signal without reflecting the first signal (unlike convolution). The comparison of cross-correlation with convolution is shown in below figure (Image Source : Wikipedia) :

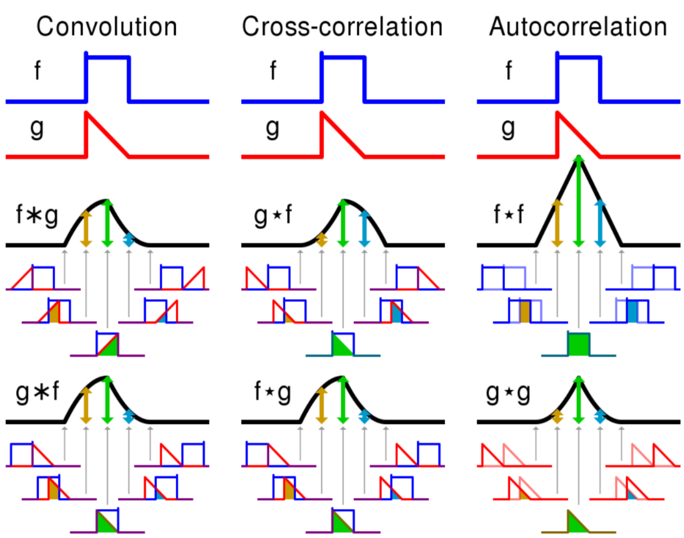

Figure 3: Convolution v/s Correlation

The cross-correlation of two discrete time signals is given by equation (3). However, signal can also be correlated with itself, which is known as auto-correlation. The auto-correlation is a measure of similarity between a signal and its delayed version as shown in equation (4)

$$R_{xy}[k] = \sum_{n=0}^{\infty} x(n)y(n-k) \tag{3}$$

$$R_{xx}[k] = \sum_{n=0}^{\infty} x(n)x(n-k) \tag{4}$$

The cross-correlation of two continuous time signals is given by:

$$x(t) \star y(t) = \int_{-\infty}^{\infty} x(t)y(t-\tau)dt \tag{5}$$

Properties of Correlation:

1. Correlation is Conjugate symmetry : $R_{12}(\tau) = R_{21}^{*}(-\tau)$
2. It is not commutative like convolution : $R_{12}(\tau) \neq R_{21}(-\tau)$
3. Auto-correlation is maximum at 0 : $|R_{xx}(\tau)| \leq R_{xx}(0)$

4 Down-Sampling

The down sampling operation (with factor M) is implemented by discharging $M-1$ consecutive samples and retaining every Mth sample. When down-sampling is applied to a discrete signal $x[n]$, it produces a downsampled output $y[n]$:

$$y[n] = x[nM] \tag{6}$$

Figure 4: Block diagram representation of a down-sampler

The purpose of performing down-sampling is to reduce the amount of data to match with the capacity of further processing blocks. For example, If you have a processing blocks which can only process an audio signal of 180KHz or below, then a 320KHz signal can be down-sampled to 180KHz signal to match the requirement.

In the figure (5), An example of down-sampling is demonstrated. We can observe that the down-sampling reduces the sampling frequency of the original signal by a factor of M.

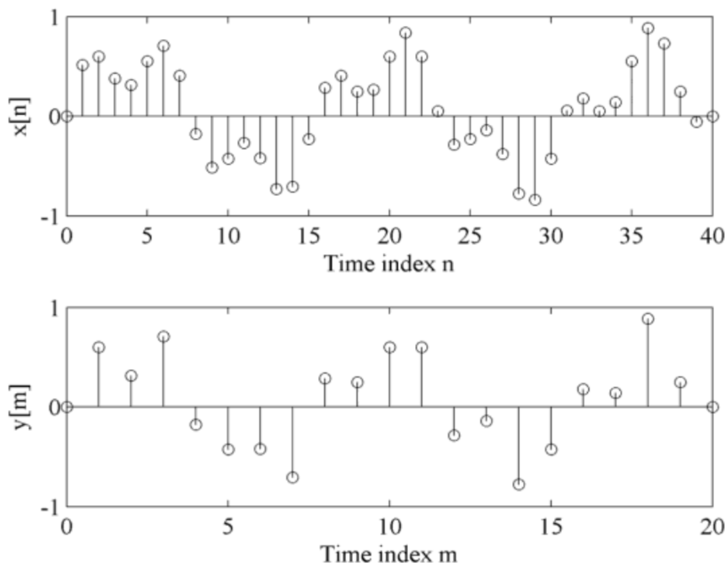

Figure 5: Example of dowm-sampling with factor M=2

5 Up-Sampling

The up-scaling with a factor of L is implemented by inserting $L-1$ zeros between two consecutive samples. When up-sampling is applied to a discrete signal $x[n]$, it produces a up-sampled output $y[n]$:

$y[n] = = x[m/L], m = 0, \pm L, \pm 2L, \cdot\cdot$

$y[n] = 0, Otherwise$

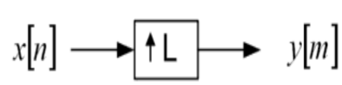

Figure 6: Block diagram representation of a up-sampler

The purpose of performing up-sampling is for rate matching by adding samples by interpolation. For example, If you want to mix two samples with different sampling rate, then low frequency signal can be up-sampled to match the higher frequency signal and the mixing can be performed easily sample-wise.

In the figure (7), An example of up-sampling is demonstrated. We can observe that the up-sampling increases the sampling frequency of the original signal by a factor of L.

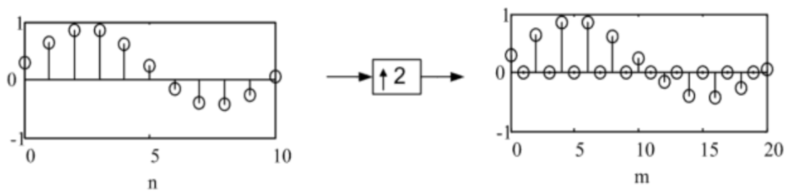

Figure 7: Example of up-sampling with factor L=2

6 conclusion

The effect of these basic signal processing algorithms on the number of samples, frequency and amplitude of the input signals can be clearly observed from the output of the C programs. (refer output file)