Aim ↬ To understand and implement the convolution of signals in order to analyze the output of a linear time-invariant (LTI) system using MATLAB.

Software Required ↬ MATLAB

Theory ↬

Convolution is a mathematical operation that combines two signals to produce a third signal. It is a fundamental tool in signal processing, especially in the analysis of linear time-invariant (LTI) systems. The convolution of two signals provides a way to determine how the shape of one signal is modified by another.

Linear Time-Invariant (LTI) System ↴

A Linear Time-Invariant (LTI) system is a system that satisfies two key properties:

1. **Linearity**: The principle of superposition applies. If the system's response to $x_1(t)$ is $y_1(t)$ and the response to $x_2(t)$ is $y_2(t)$, then the response to $a \cdot x_1(t)+b \cdot x_2(t)$ is $a \cdot y_1(t)+b \cdot y_2(t)$, where a and b are constants.
2. **Time-Invariance**: The system's behavior does not change over time. If the response to x(t) is y(t), then the response to $x(t-t_0)$ is $y(t-t_0)$ for any time shift t_0.

Continuous-Time Convolution ↴

For continuous-time signals, the convolution of two signals x(t) and h(t) is defined as:

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

Here:

- x(t) is the input signal.
- h(t) is the impulse response of the system.

Discrete-Time Convolution ↴

For discrete-time signals, the convolution of two signals x[n] and h[n] is defined as:

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

Here:

- x[n] is the input signal.
- h[n] is the impulse response of the system.

Properties of Convolution ↴

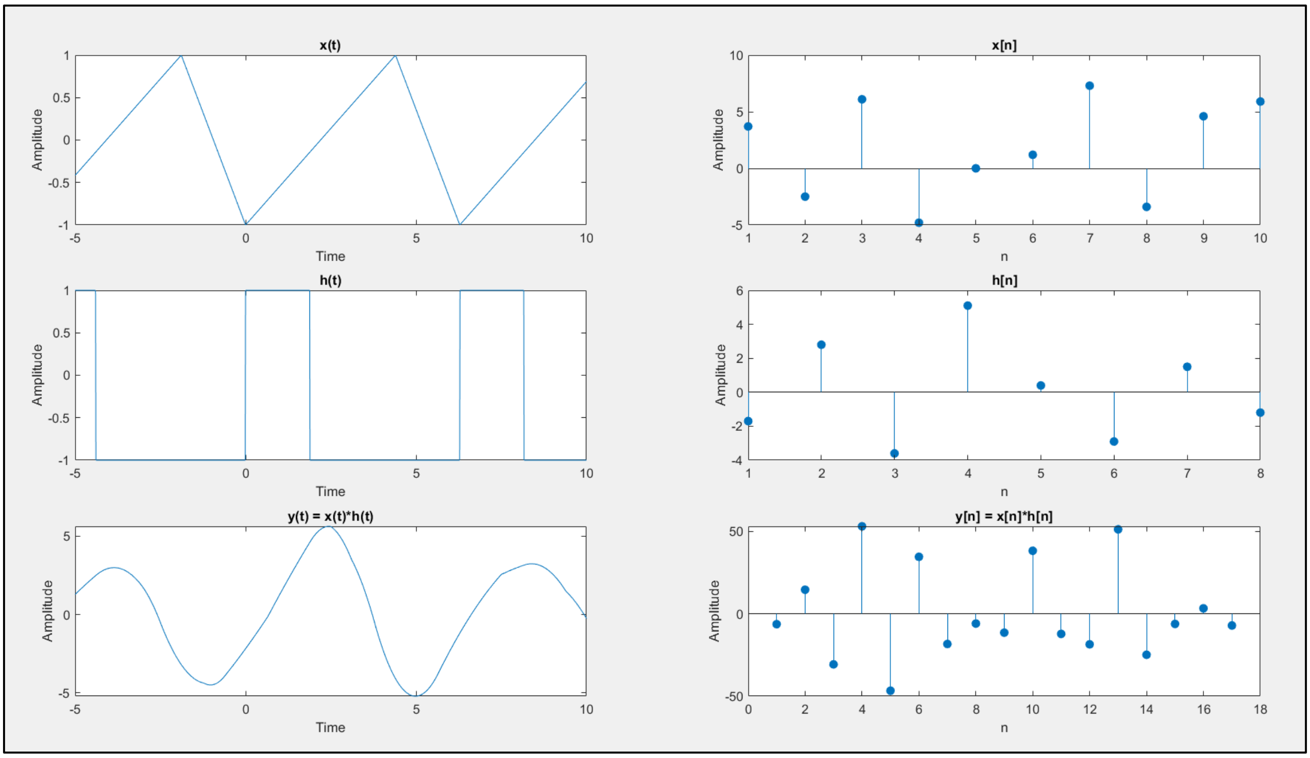
1. Commutativity: $x(t) * h(t) = h(t) * x(t)$
2. Associativity: $x(t) * [h(t) * g(t)] = [x(t) * h(t)] * g(t)$
3. Distributivity: $x(t) * [h(t) + g(t)] = x(t) * h(t) + x(t) * g(t)$
4. Identity: Convolution with a delta function δ(t) or δ[n] yields the original signal: $x(t) * \delta(t) = x(t)$ or $x[n] * \delta[n] = x[n]$.

Code ↬

```
%Convolution
clc;
clear;
t = -5:0.01:10;
x_t = sawtooth(t, 0.7);
h_t = square(t,30);
y_t = conv(x_t, h_t, 'same') * 0.01;

subplot(3,2,1);
plot(t,x_t);
xlabel('Time');
ylabel('Amplitude');
title('x(t)');

subplot(3,2,3);
plot(t,h_t);
xlabel('Time');
ylabel('Amplitude');
title('h(t)');

subplot(3,2,5);
plot(t, y_t);
xlabel('Time');
ylabel('Amplitude');
title('y(t) = x(t)*h(t)');
```

```
x_n = [3.7,-2.5,6.1,-4.8,0,1.2,7.3,-3.4,4.6,5.9];
h_n = [-1.7,2.8,-3.6,5.1,0.4,-2.9,1.5,-1.2];

y_n = conv(x_n, h_n);

subplot(3,2,2);
stem(x_n, 'filled');
xlabel('n');
ylabel('Amplitude');
title('x[n]');

subplot(3,2,4);
stem(h_n, 'filled');
xlabel('n');
ylabel('Amplitude');
title('h[n]');

subplot(3,2,6);
stem(y_n, 'filled');
xlabel('n');
ylabel('Amplitude');
title('y[n] = x[n]*h[n]');
```

Output ↬



Result ↬

In the continuous-time convolution, the sawtooth wave x(t) and square wave h(t) were convolved to produce the output y(t), demonstrating how the shape of x(t) is modified by h(t). In discrete-time convolution, the sequences x[n] and h[n] were convolved, showing the impact of h[n] on x[n] and resulting in the output y[n]. The plots of both continuous and discrete convolutions illustrate the effects of these operations on the signals.

Conclusion ↬

The continuous-time and discrete-time convolutions effectively demonstrate how an input signal interacts with an impulse response. The continuous convolution approximates the convolution integral using discrete sampling, while the discrete convolution directly combines sequences. Both methods highlight the principles of signal processing, including linearity and time invariance.

Precautions ↬

- Verify sequence lengths are suitable for convolution to avoid artifacts.
- Ensure correct interpretation of the convolution output in relation to input sequences.
- Double-check input parameters for accuracy.
- Confirm that plots accurately represent the convolution results.