Laboratory

9

Sampling and Convolution

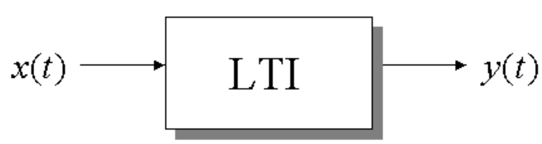
Objective

The main goal of this experiment is how to use MATLAB to compute the output of the Linear Time Invariant systems using the built-in functions in Matlab, also to write a Matlab code to sampling a given function and then convolute it with other sampled function.

Introduction

- **Linear Time Invariant Systems**

  If we have a linear time invariant system and we know the impulse response of it (h(t) we can know the output of any other input using convolution. Convolution is an operation by which the output of an linear time-invariant (LTI) system with a known impulse response can be determined which is shown in Figure 1, given an arbitrary input signal. Observe the system to the left, with continuous-time input x(t) and output y(t). Convolution is simply the process that determines the output given the input.



$$
\begin{aligned}
x(t) &= \delta(t), \\
y(t) &= \int_{-\infty}^{t} x(\tau)\,d\tau \\
&= \int_{-\infty}^{t} \delta(\tau)\,d\tau \\
&= u(t)
\end{aligned}
$$

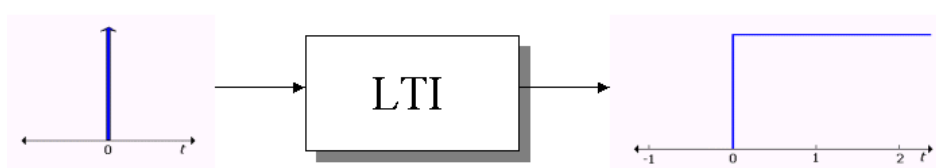


Figure 1 : LTI systems when input is Impulse

The unit impulse is used as an example input for the system shown above. When the input to any LTI system is a unit impulse, the output is called the impulse response and is denoted by h(t). So, in this example h(t)=u(t) The system in this particular example is known as an integrator because it produces a unit step signal as the output, according to the system's corresponding mathematical relationship shown to the left of it. In general, however, any relationship which is linear and time-invariant, with unit impulse as input qualifies as a valid impulse response for an LTI system.

- **Continues Convolution**

The convolution, y(t) of two signals, s(t) and h(t) is expressed by the convolution integral

$$y(t) = \int_{-\infty}^{+\infty} s(\tau)\, h(t-\tau)\, d\tau \quad = s(t) * h(t)$$

Here, the shorthand notation is included that replaces the explicit integration operation by an asterisk between the convolved functions.

Dissecting the convolution integral, we clearly see that there is some kind of a multiplication by s(t) and h(t) in the integrand and there's an integration ("summation") over the variable τ. But what does t-τ mean?

First off, s(τ) and h(τ) are the same as s(t) and h(t); the variable symbol makes no difference. h(-τ) is h(τ) that is flipped or reversed in time and h(t-τ) is the function h(-τ) time shifted along the τ axis by an amount t (Figure 2). In Figure 2. the shift is to the right, i.e., t>0; if t<0 the shift would be to the left. The time shift, t is the time at which the output, y is determined. To illustrate these concepts in Figure 2, the signals s(t) and h(t) are taken to be rectangle functions of unit length with heights of 1 and 0.5, respectively.

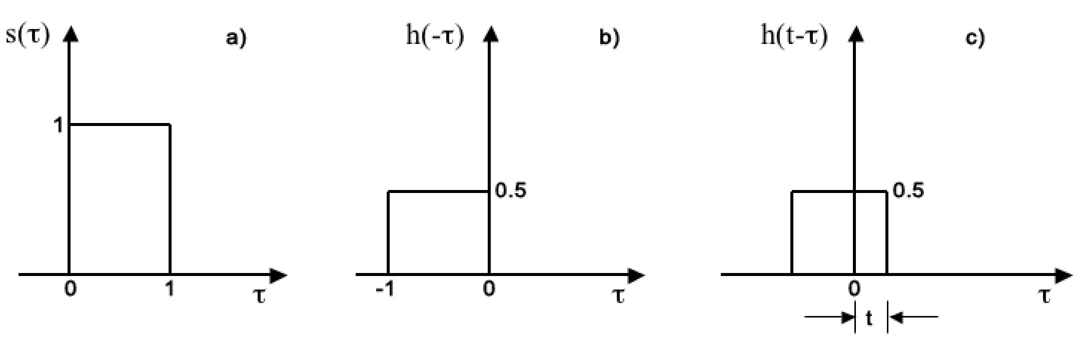


Figure 2 Graphical description of flipping and shifting operations used in convolution: a) input signal, s(t), b) reversed impulse response function, h(-τ), and c) shifted h(-τ) by t yields h(t-τ).

The convolution integral is continuously evaluated at each time shift t by multiplication and integration of s(τ) times h(t-τ) for all values of t possibly running from -infinity to + infinity.

- **Discrete Convolution**

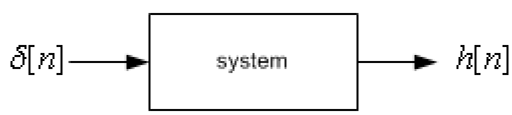
  Analog to Discrete conversion produces digital signals sampled at a particular sampling interval, Δt. Assuming both x(n) and h(n) are digital functions with a sampling interval of unity, the convolution operation is defined as

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

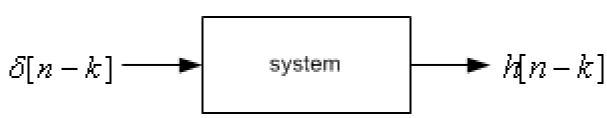
where x[n] is input signal, h[n] is impulse response, and y[n] is output. * denotes convolution. Notice that we multiply the terms of x[k] by the terms of a time-shifted h[n] and add them up.

Impulse Response

Impulse response is the output of a system resulting from an impulse function as input. And it is denoted as h[n].

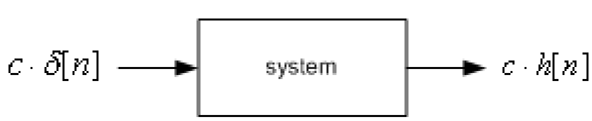


If the system is time-invariant, the response of a time-shifted impulse function is also shifted as same amount of time.



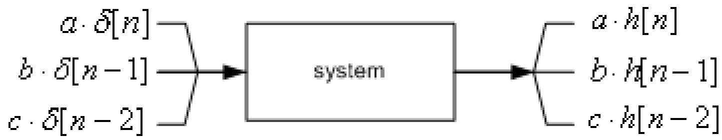
For example, the impulse response of δ[n-1] is h[n-1]. If we know the impulse response h[n], then we can immediately get the impulse response h[n-1] by shifting h[n] by +1. Consequently, h[n-2] results from shifting h[n] by +2.

If the system is linear (especially scalar rule), a scaled in input signal causes an identical scaling in the output signal.



For instance, the impulse response of 3·δ[n] is just multiplying by 3 to h[n].

If the input has 3 components, for example, a·δ[n] + b·δ[n-1] + c·δ[n-2], then the output is simply a·h[n] + b·h[n-1] + c·h[n-2]. This is called additive property of linear system, thus, it is valid only on the linear system.



Procedure

- **Sampling Signals**

  In this section , you will be able to sampling any given function in time domain for a given sampling frequency **fs** . The following code sampling two signals in MATLAB which generate a two time vectors at the appropriate rate, and use the generated vectors to generate the signals and then plot the sampled signals using **stem** function.

```
f = 2000;
T = 1/f;
tmin = 0;
tmax = 5*T;
dt = T/100;
dt1 = 1/10000;
dt2 = 1/3000;
t = tmin:dt:tmax;
t1 = tmin:dt1:tmax;
t2 = tmin:dt2:tmax;
x = sin(2*pi*f*t);
x1 = sin(2*pi*f*t1);
x2 = sin(2*pi*f*t2);
subplot(211)
plot(t,x,'r');
hold on
stem(t1,x1);
subplot(212)
plot(t,x,'r');
hold on
stem(t2,x2);
```

Figure 3 shows the sampled signals at different sampling frequencies for sinusoidal function with frequency 2000 Hz .

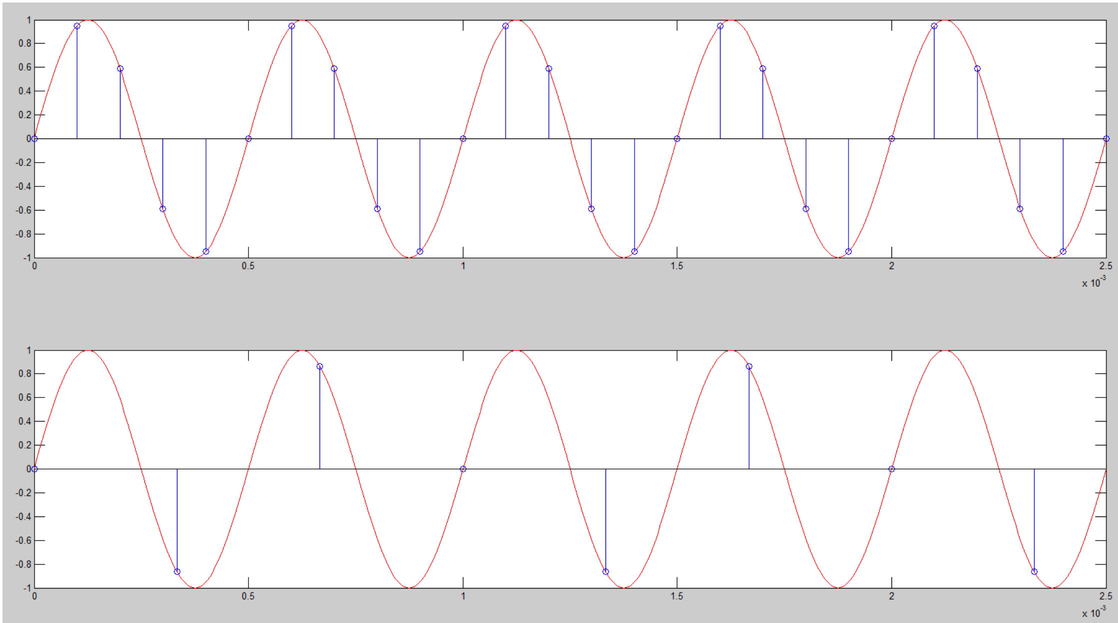


Figure 3: Sampling Signals at different sampling frequencies.

According to above code and generated sampling signals, answer the following questions :

1. Calculate the sampling frequency for each sampled signal.
2. Calculate the number of samples in one Cycle for each sampled signal.
3. Using **Nyquist** Sampling Theorem ($F_s >= 2F_o$), which of the sampled signals can be reconstructed without Aliasing problem?

- **Continues Signal Convolution**

  Now you will be able to do continuous convolution between two different continues signals x(t) and h(t), where x(t) is the input signal and h(t) is the impulse response of the LTI system. In MATLAB, there is a built-in function called **conv** which takes two arguments : input function as a vector and impulse response function as a vector.

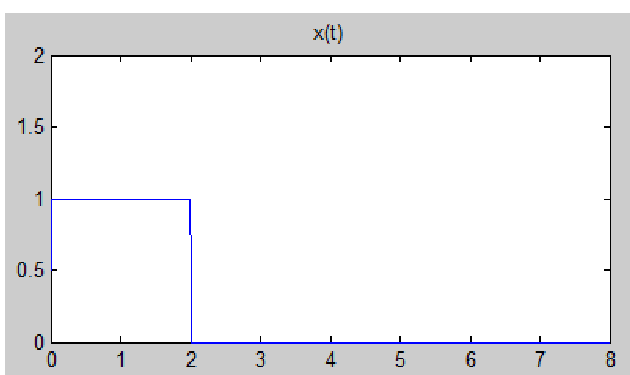


Figure 4: X(t) input signal

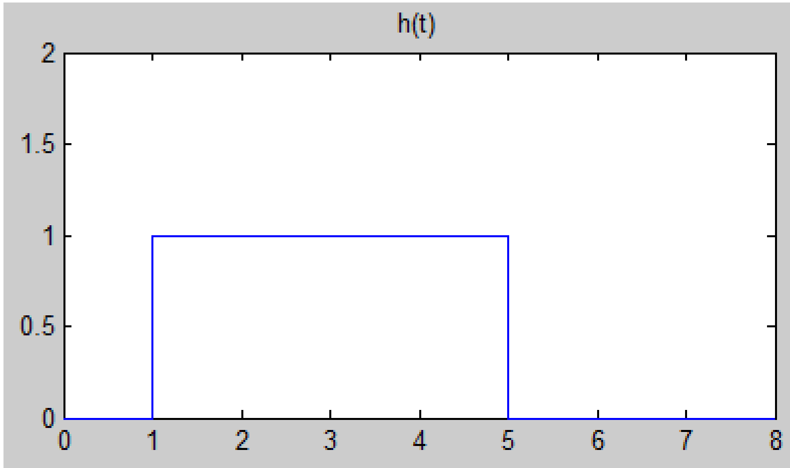


Figure 5: H(t) impulse response signal

The following MATLAB code shows how Figure 4 and 5 are generated and plotted and also shows how the output of the system can be produced by using convolution.

```
t=0:.01:8;
T=0:.01:16;
x= heaviside(t)-heaviside(t-2);
subplot(2,2,1)
plot(t,x)
title('x(t)')
h=heaviside(t-1)-heaviside(t-5);
subplot(2,2,2)
plot(t,h)
title('h(t)')
y=conv(x,h).*.01;
subplot(2,1,2)
plot(T,y)
title('y(t)')
```

Figure 6 shows the output of the system. It is important to note that in the code, the **conv** is multiplied by 0.01 which is the time step since compute the integral by multiply the signal high with the time width which is the time step.

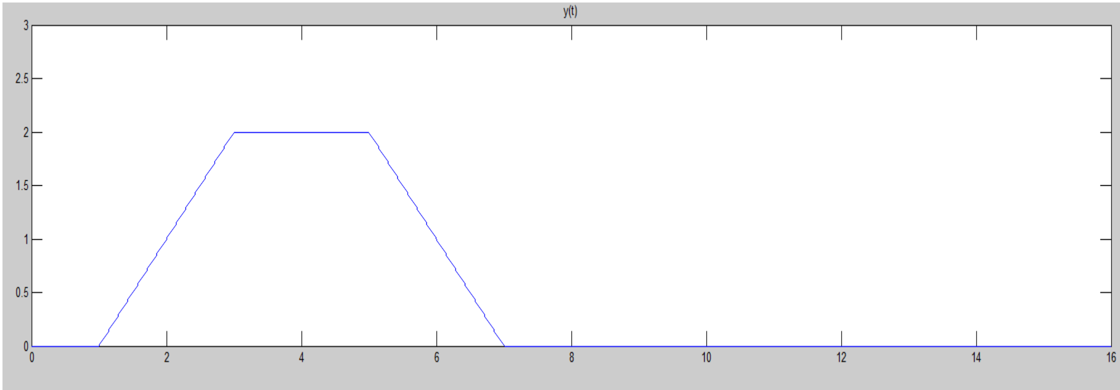


Figure 6: Y(t) system output , Y(t)=X(t)*H(t)

- **Discrete Convolution**

  If arbitrary sequences are of infinite duration, then MATLAB cannot be used directly to compute the convolution. MATLAB does provide a built-in function called **conv** that computes the convolution between two finite-duration sequences. The **conv** function assumes that the two sequences begin at n = 0 and is invoked by :

  y = conv(x,h);

  For example, given the following two sequences:

  x(n) = [3,11,7, 0,-1,4,2], $0 \le n \le 6$; h(n) = [2,3,0,-5,2,1], $0 \le n \le 5$

  Determine the convolution y(n) = x(n) * h(n). The output result Y[n] is shown in Figure 7.

```
x = [3 11 7 0 -1 4 2];
h = [2 3 0 -5 2 1];
y = conv(x,h)
stem(0:1:11,y)
title('y[n]')
y = 6    31    47    6   -51    -5    41    18   -22    -3    8    2
```

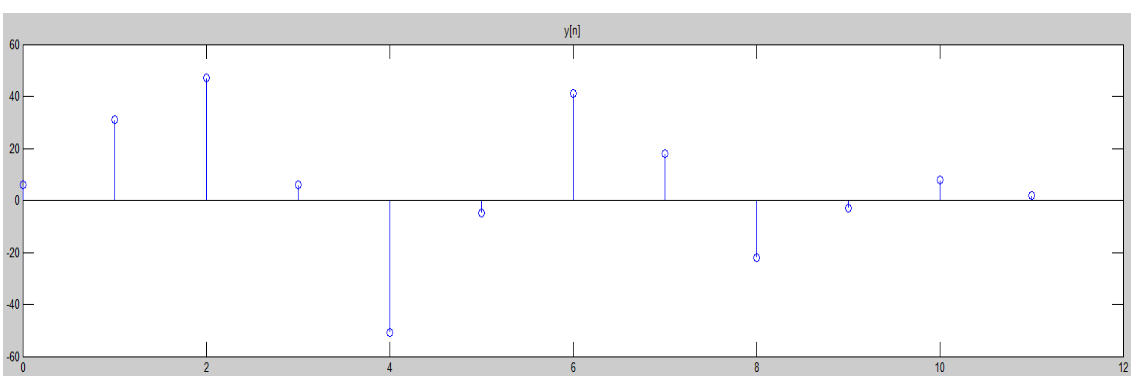
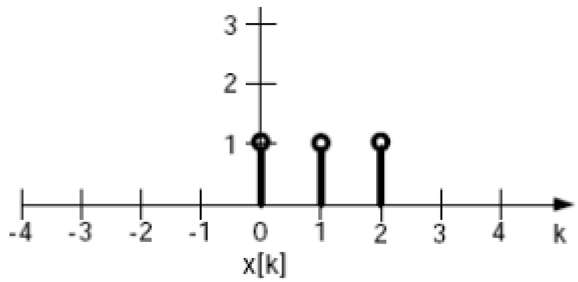
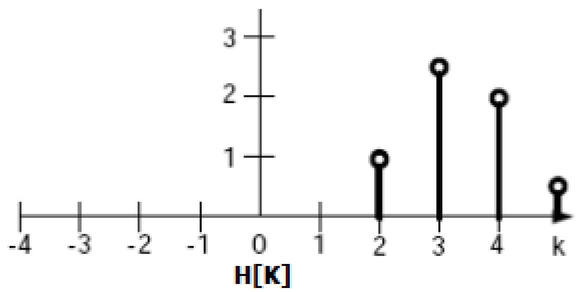


Figure 7: convolution output Y[n]=X[n]*H[n]

TODO:

1. Compute and plot the convolution M(k)= X(k)*h(k) for the following X(k) and h(k).





2. Compute and plot the convolution C(t)=X(t)*H(t) and then sample the output convolution for two different values of sampling rate.

   X(t) = sin(2*Pi*1000*t) , H(t)= cos(s*Pi*1000*t).

3. Once the M[k] and C[k] are computed and plotted , compute and plot the convolution Y[k]= M[k] * C[k].