2. Discrete Time Signal Operations

EE 274/COE 197E

Rhandley Cajote, Ph. D.
Crisron Lucas, MSc.

Today's Lesson:

DT Signal Manipulation

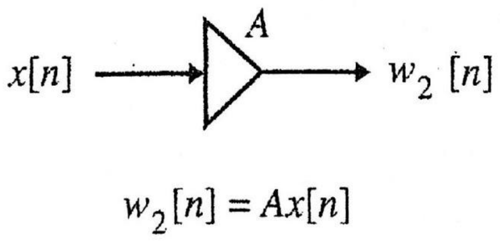
a. Amplitude scaling
   i. Amplify/Boost
   ii. Attenuate/Cut
   iii. Invert
b. Time shifting
   i. Delay
   ii. Advance
c. Time scaling
   i. Upsample
   ii. Downsample
   iii. Reversal

DT Signal Operation

a. Discrete Math Operations
   i. Addition/Subtraction
   ii. Modulation
b. Signal Decomposition
   i. Even component
   ii. Odd component
c. Complex Operations
   i. DTS Convolution
d. Signal Reconstruction
   i. Interpolation (D2A)

Amplitude scaling & inversion

$$\boxed{x[n] \rightarrow Ax[n]}$$



$|A| > 0 \rightarrow$ amplify/boost

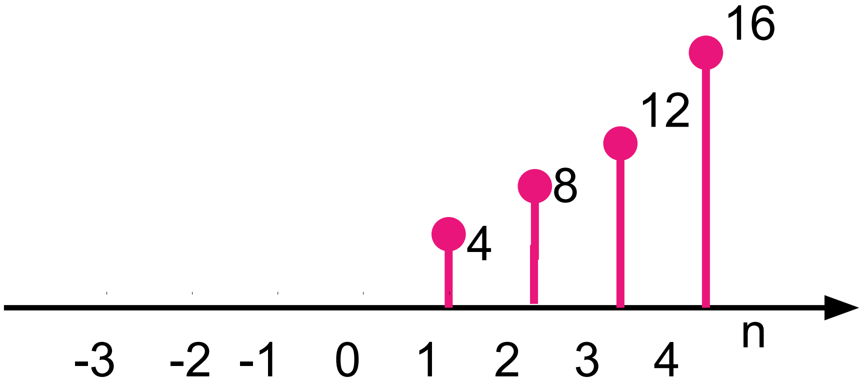
$0 < |A| < 1 \rightarrow$ gain reduction/attenuate

$A < 0 \rightarrow$ invert

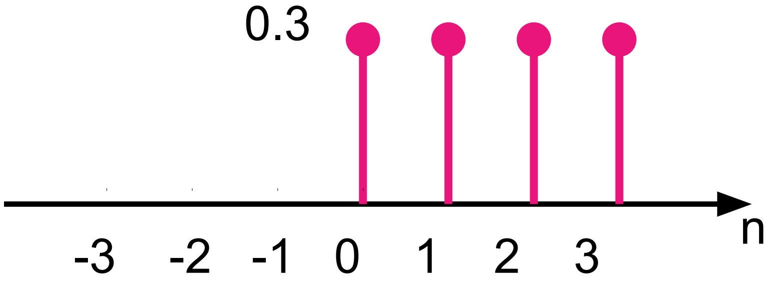
Discrete Time Signal Manipulation

Amplitude scaling

Amplified ramp, $4r(n)$



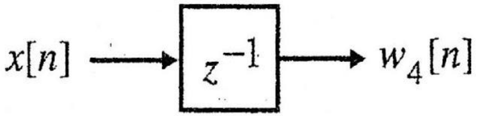
Attenuated step, $0.3u(n)$



Discrete Time Signal Manipulation

Time shifting

$$\boxed{x[n] \rightarrow x[n-n_0]}$$



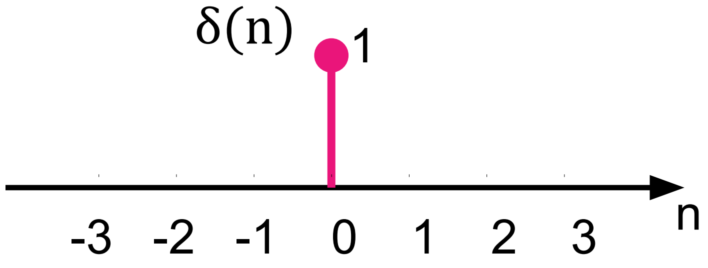
$w_4[n] = x[n-1]$

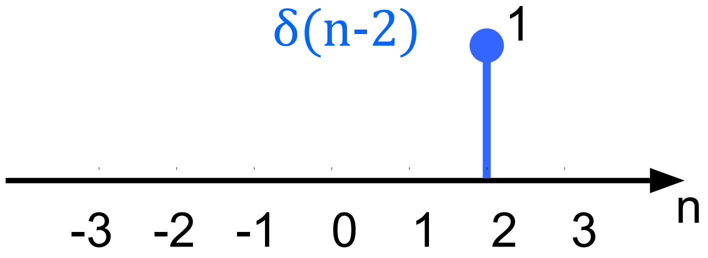
$n_0 > 0 \rightarrow$ time delay

$n_0 < 0 \rightarrow$ time advance

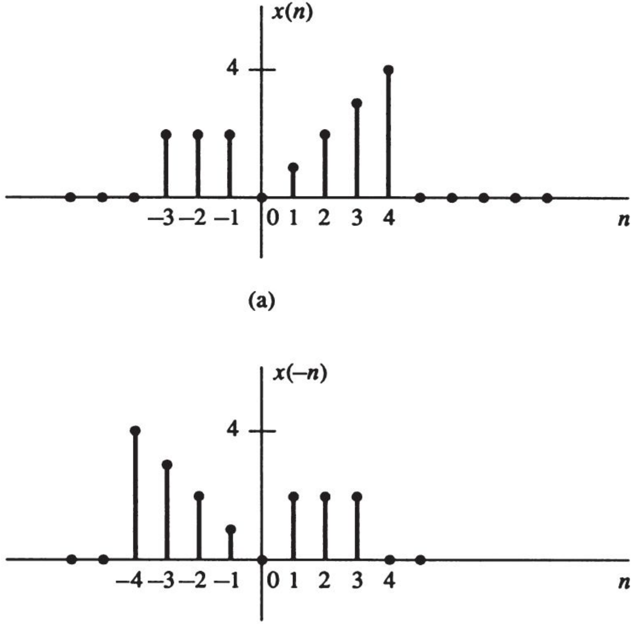
Discrete Time Signal Manipulation

Time shifting





Time scaling & reversal



(a)

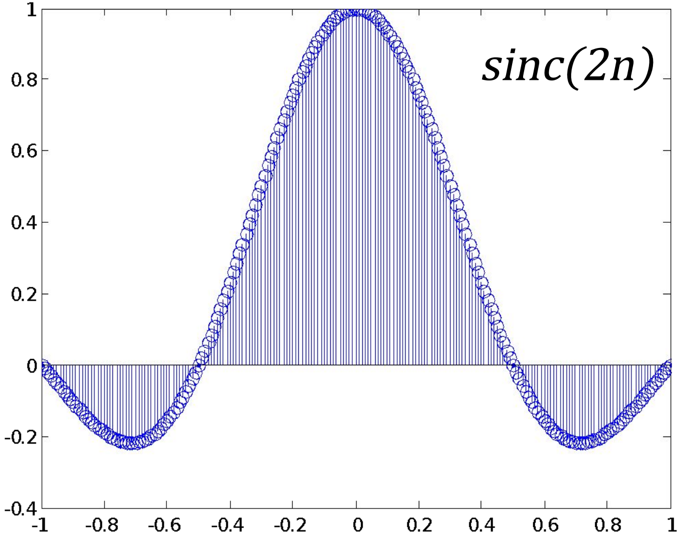
(b)

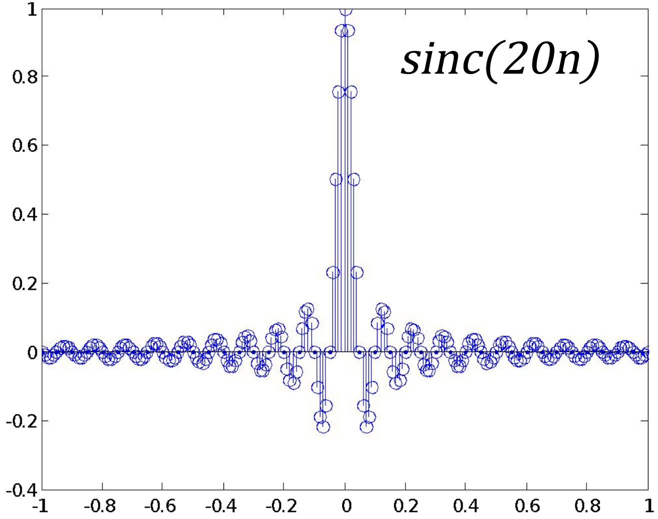
$$\boxed{x[n] \rightarrow x[kn]}$$

$$|k| > 1 \quad \rightarrow \text{compress}$$
$$0 < |k| < 1 \quad \rightarrow \text{stretch}$$
$$k < 0 \quad \rightarrow \text{time reversal/fold}$$

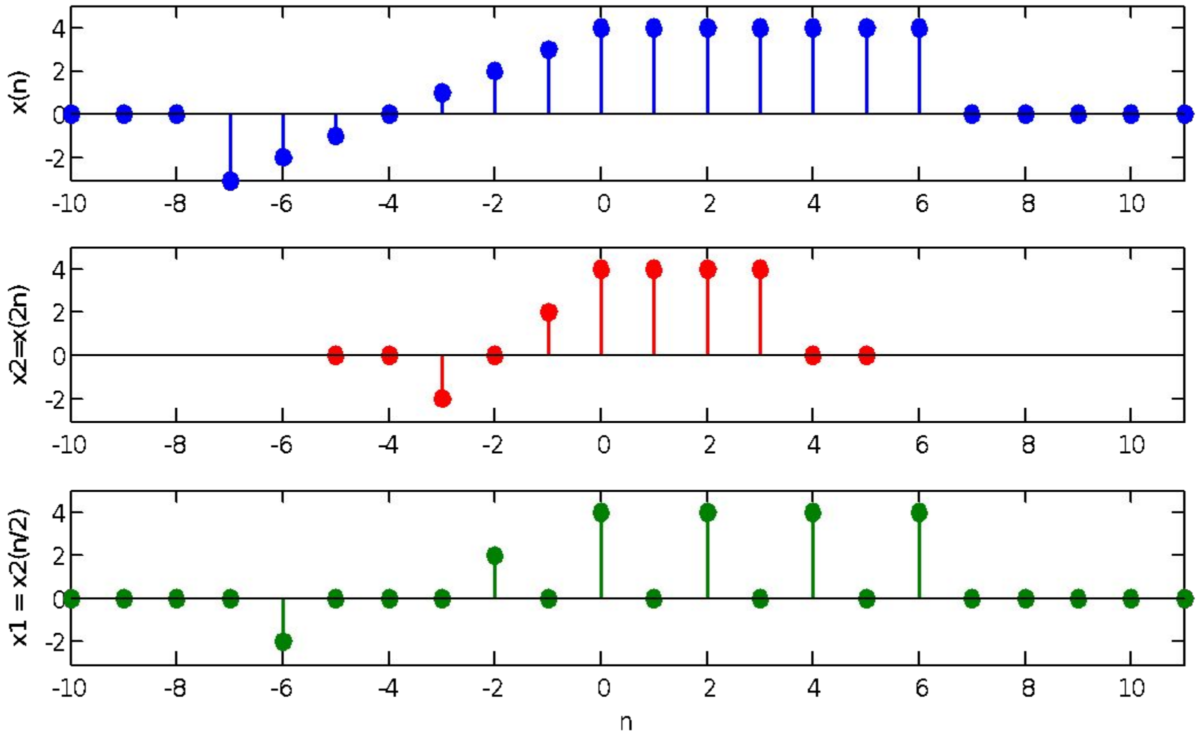
Discrete Time Signal Manipulation

Time scaling





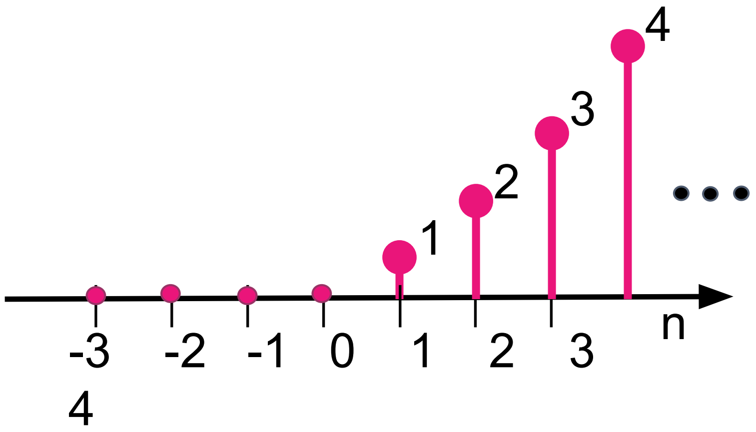
Up-sampling & Down-sampling



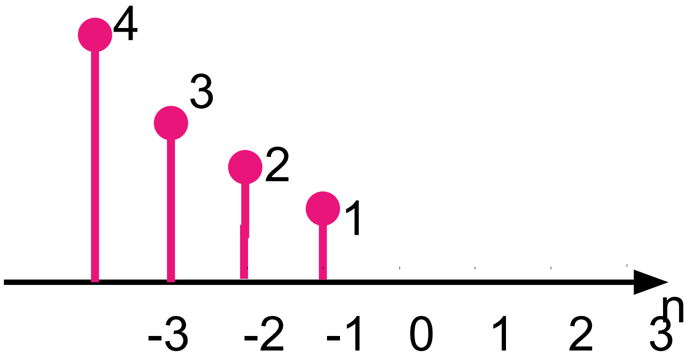
Discrete Time Signal Manipulation

Time reversal

Unit ramp, $r(n)$



Time-reversed unit ramp, $r(-n)$



Notes on Signal Manipulation

- Time Delay and Folding are **NOT** commutative

$$TD[x(n)] = x(n-k), \quad k > 0$$
$$FD[x(n)] = x(-n)$$

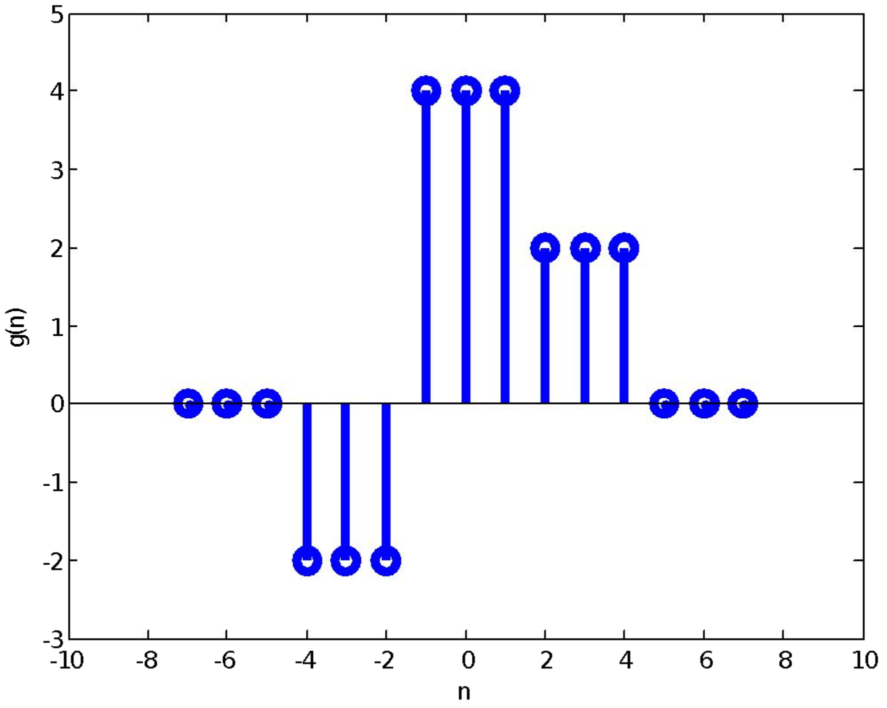
$$TD\,\{FD[x(n)]\} = TD[x(-n)] = x(-n+k)$$

$$FD\,\{TD[x(n)]\} = FD[x(n-k)] = x(-n-k)$$

Example:

- Describe the following DTS signal using elementary functions

$$g(n)=\begin{cases}-2, \text{for } -4\le n\le -2\\ 4, \text{for } -1\le n\le 1\\ 2, \text{for } 2\le n\le 4\\ 0, \quad \text{elsewhere}\end{cases}$$



Example:

- **Answer:**

$$-2\delta[n+4] - 2\delta[n+3] - 2\delta[n+2]$$

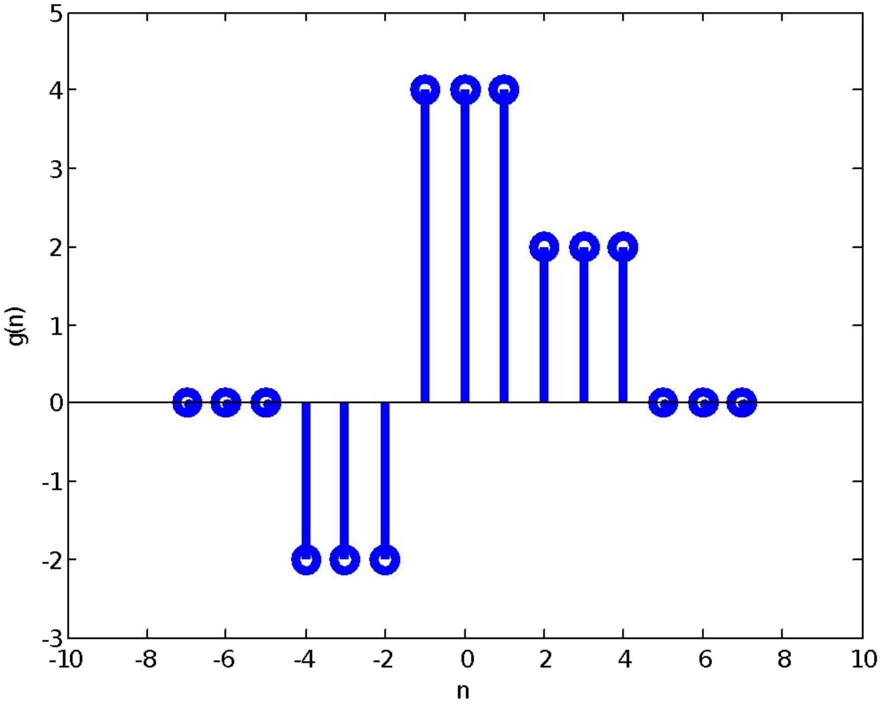
$$+4\delta[n+1] + 4\delta[n] + 4\delta[n-1] +$$

$$2\delta[n-2] + 2\delta[n-3] + 2\delta[n-4]$$

or

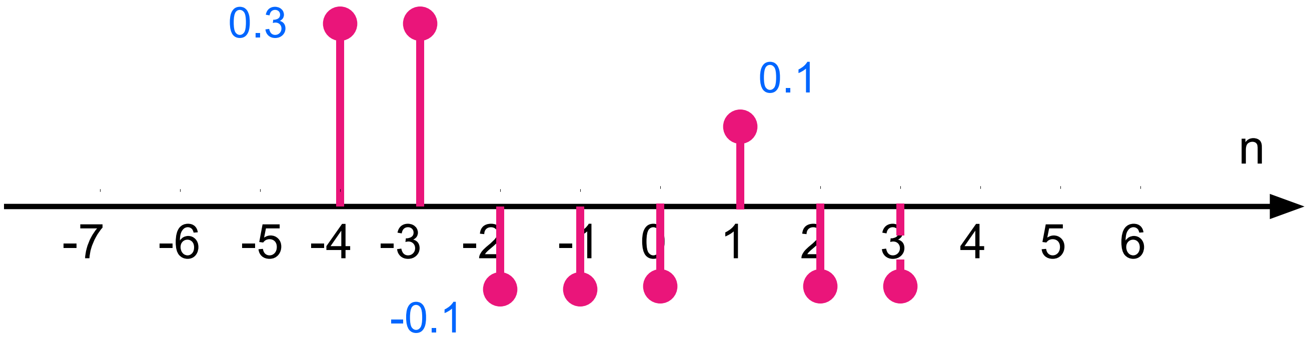
$$-2u[n+4] + 6u[n+1] - 2u[n-2]$$

$$-2u[n-5]$$



Example:

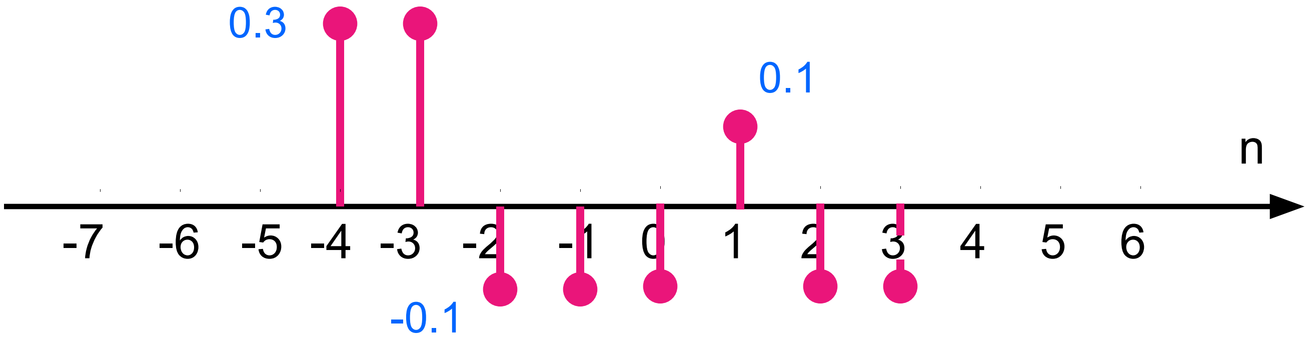
- Describe the following DTS signal using elementary functions



Example:

- **Answer:**

$$x(n) = 0.3u(n+4) - 0.4u(n+2) + 0.2\delta(n-1) + 0.1u(n-4)$$



Even & Odd Signal Decomposition

A discrete time signal can be decomposed into its even and odd components:

$$x_{even}[n] = \frac{x[n]+x[-n]}{2} \qquad x_{odd}[n] = \frac{x[n]-x[-n]}{2}$$

<u>Example</u>:

$x[n] = \{1,\underline{2},3,4,5\}$

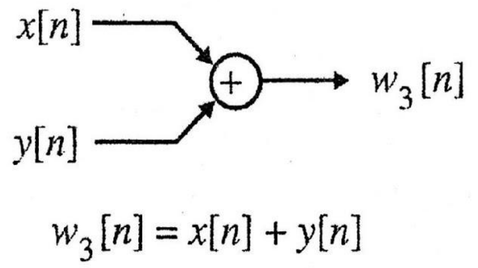
$x_{even}[n] = \{2.5,2,2,\underline{2},2,2,2.5\}$

$x_{odd}[n] = \{-2.5,-2,-1,\underline{0},1,2,2.5\}$

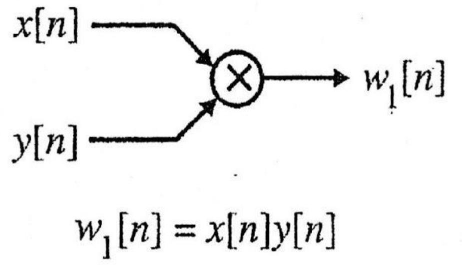
```
>> x = [0 0 1 2 3 4 5];
>> x_fold = fliplr(x)
>> x_even = 0.5*(x+x_fold)
>> x_odd = 0.5*(x-x_fold)
```

Mathematical Operations on DTS

Addition/Subtraction: $x[n] \pm y[n]$



Modulation: $x[n] \, . \, y[n]$



How about Division, Integration and Differentiation?

Notes on Addition and symmetry

Sum of two even DTS:

$$z[n] = x[n] + y[n]$$
$$= x[-n] + y[-n]$$
$$= z[-n] \quad \rightarrow \textbf{even}$$

Sum of two odd DTS:

$$z[n] = x[n] + y[n]$$
$$= -x[-n] - y[-n]$$
$$= -z[-n] \rightarrow \textbf{odd}$$

Sum of even + odd? → NO SYMMETRY

DTS Convolution

Convolutional Sum of 2 DTS, x[n] and h[n]:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

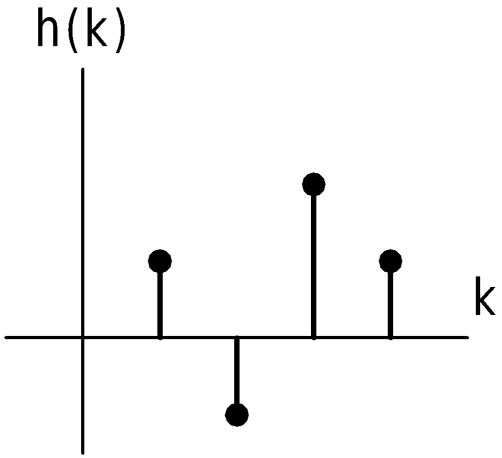
$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

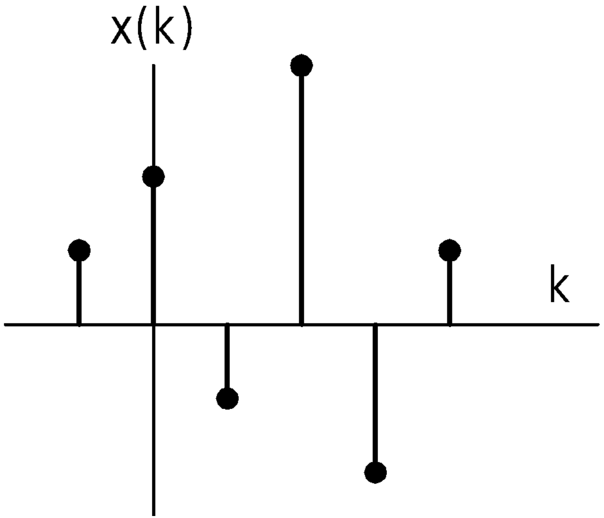
- Denoted as **y[n] = x[n]*h[n]** or **y[n] = h[n]*x[n]**
- Involves 1) Folding, 2)Shifting, 3) Multiplication, and 4) Addition
- Convolution is commutative as well as associative

Example:

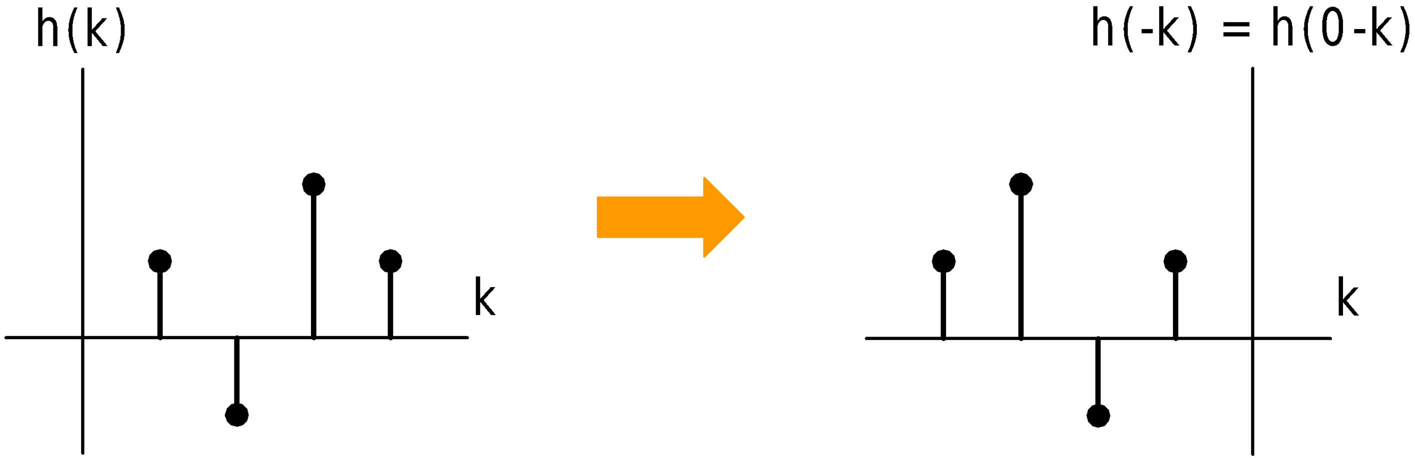
Convolve the following signals:

$h[n] = \{\underline{0}, 1, -1, 2, 1\}$ and $x[n] = \{1, \underline{2}, -1, 3, -2, 1\}$



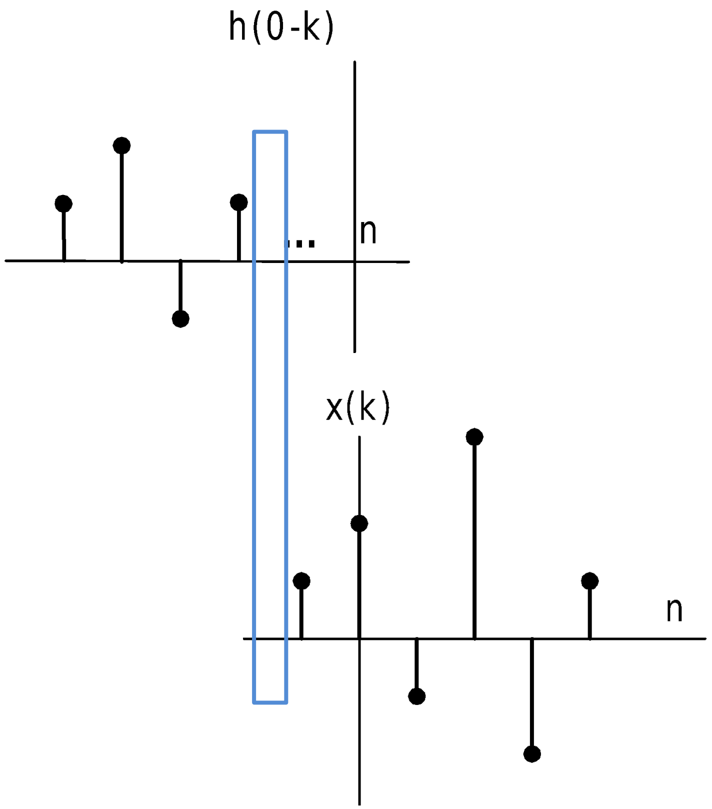


DTS Convolution



Flip/Fold

DTS Convolution

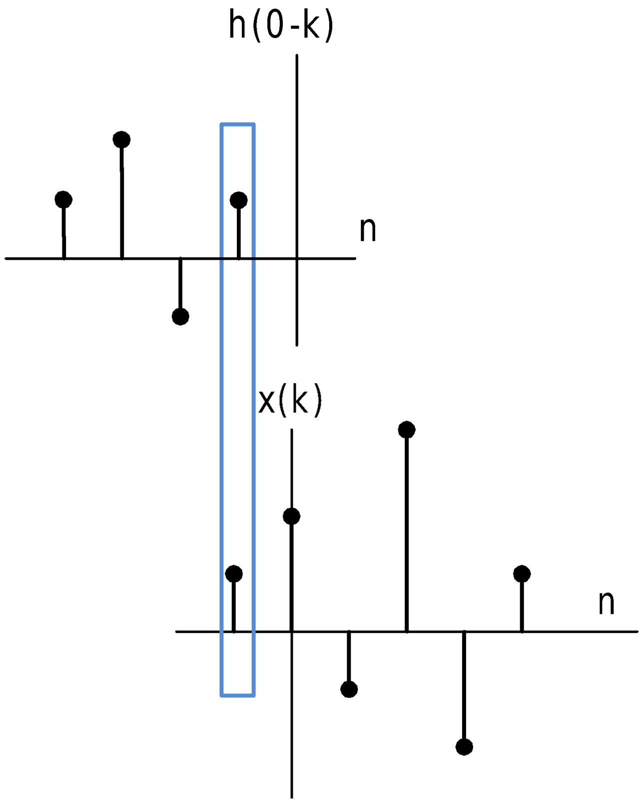


Shift:

Note that when $n_0 < 0$,

there are no overlapping signals: y(-1), y(-2),... = 0

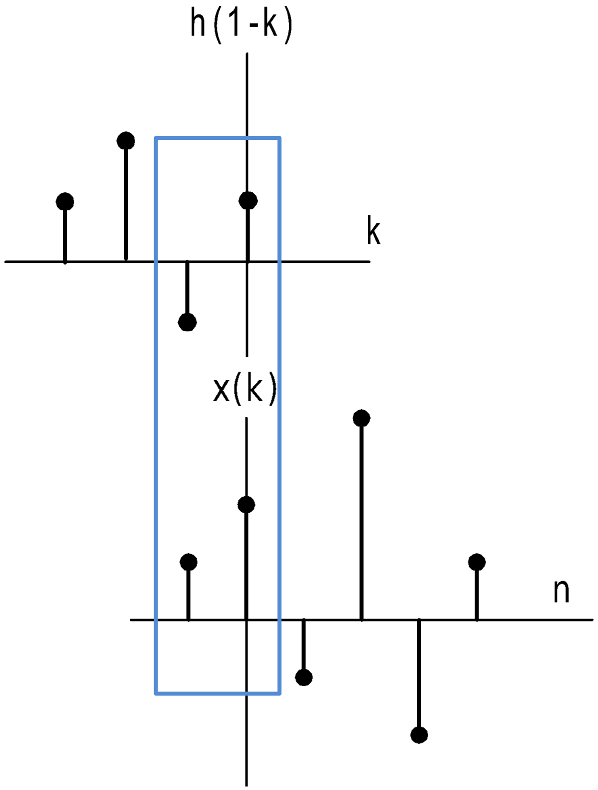
DTS Convolution



Shift:
Note that when $n_0 = 0$,

$$y(0) = 1$$

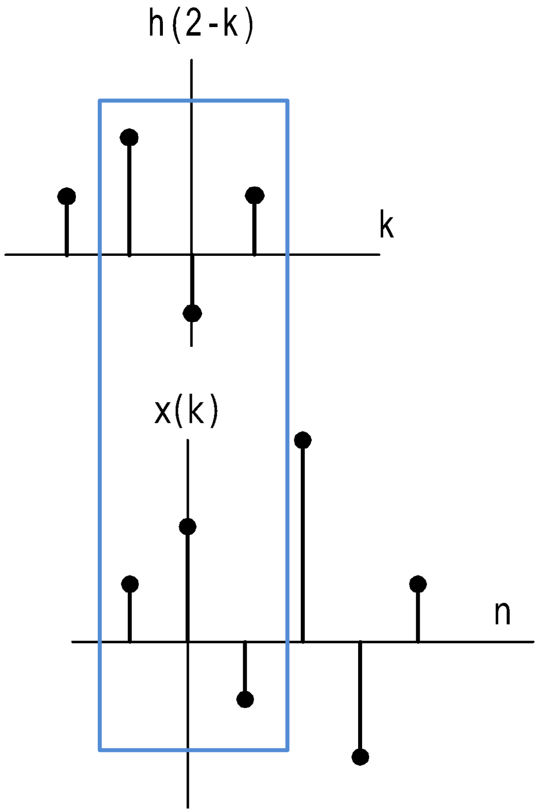
DTS Convolution



Shift:
Note that when $n_0 = 1$,

$$y(1) = -1(1) + 1(2) = 1$$

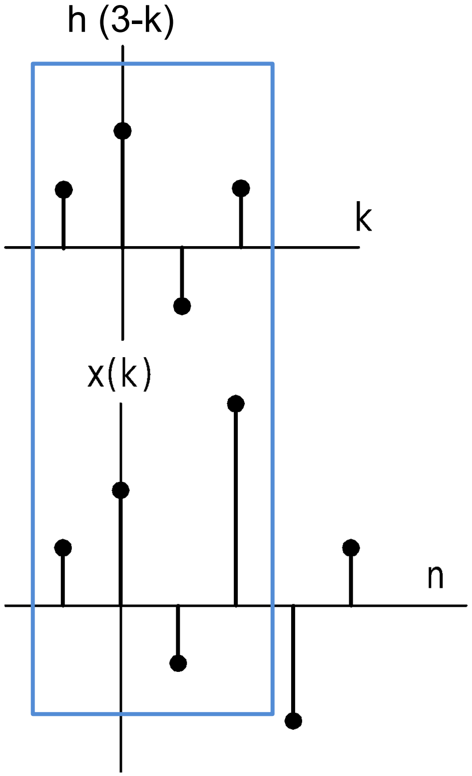
DTS Convolution



Shift:
Note that when $n_0 = 2$,

$$y(2) = 2(1) -1(2) +1(-1)= -1$$

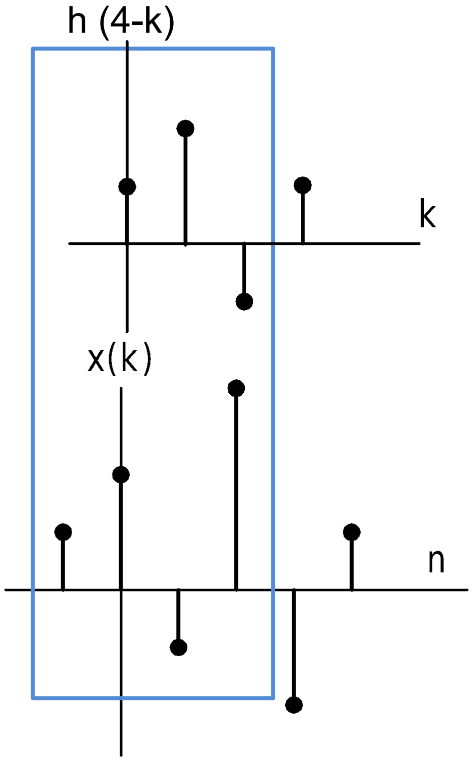
DTS Convolution



Shift:
Note that when $n_0 = 3$,

$$y(3) = 1(1) + 2(2) + -1(-1) + 1(3) = 9$$

DTS Convolution

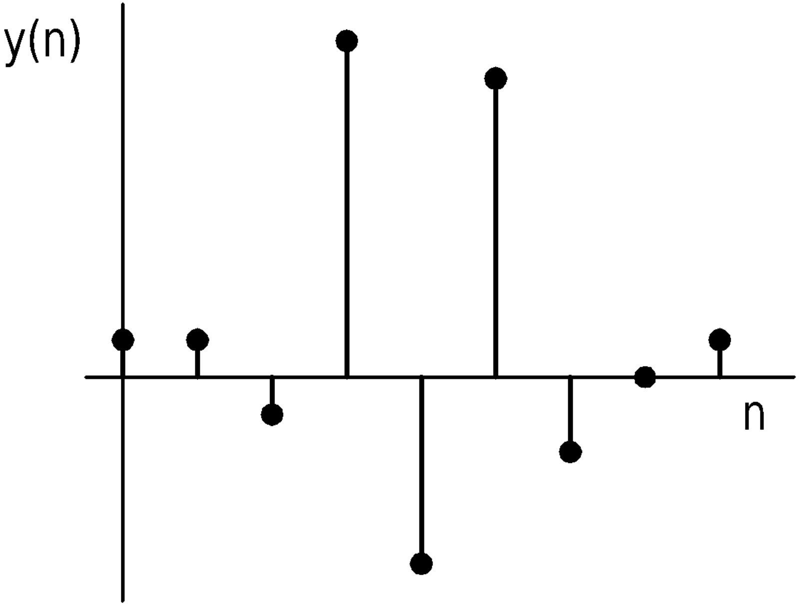


Shift:

Note that when $n_0 = 4$,

$y(4) = 1(2) + 2(-1) + -1(3) + 1(-2) = -5$

DTS Convolution



$y(n) = \{\, \underline{1}, 1, -1, 9, -5, 8, -2, 0, 1 \,\}$

```
>> h = [1 -1 2 1];
        % 1<n<4
>> x = [1 2 -1 3 -2 1];
        %-1<n<4
>> y = conv(h,x)
        % 0<n<8
```

DTS Convolution

Table method:

n	-1	0	1	2	3	4	5	6	7	8
x[n]	1	2	-1	3	-2	1				
h[n]		0	1	-1	2	1				
		1	2	-1	3	-2	1			
			-1	-2	1	-3	2	-1		
				2	4	-2	6	-4	2	
					1	2	-1	3	-2	1
y[n]		**<u>1</u>**	**1**	**-1**	**9**	**-5**	**8**	**-2**	**0**	**1**

Signal Reconstruction

x_a(t) can be exactly recovered from its sample values **x(n)** using the **interpolation function**:

$$g(t) = \frac{\sin 2\pi Bt}{2\pi Bt}$$

- **B = bandwidth**

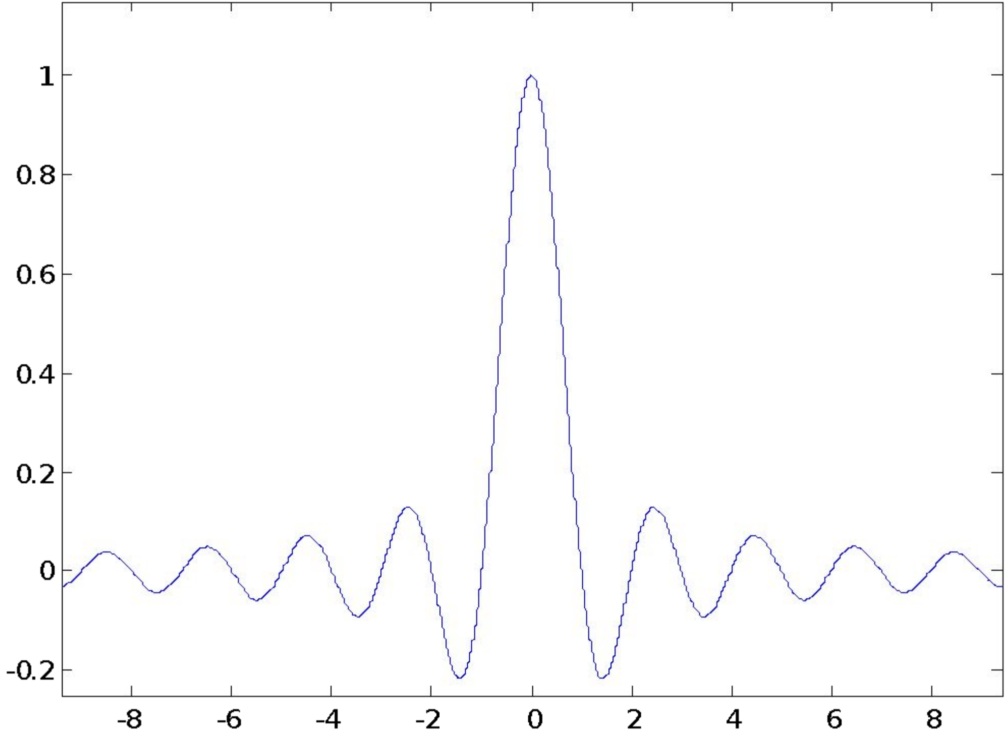
Let x(n) be samples of x_a(t)

$$x_a(t) = \sum_{n=-\infty}^{\infty} x(n) * g\left(t - \frac{n}{F_s}\right)$$

Signal Reconstruction

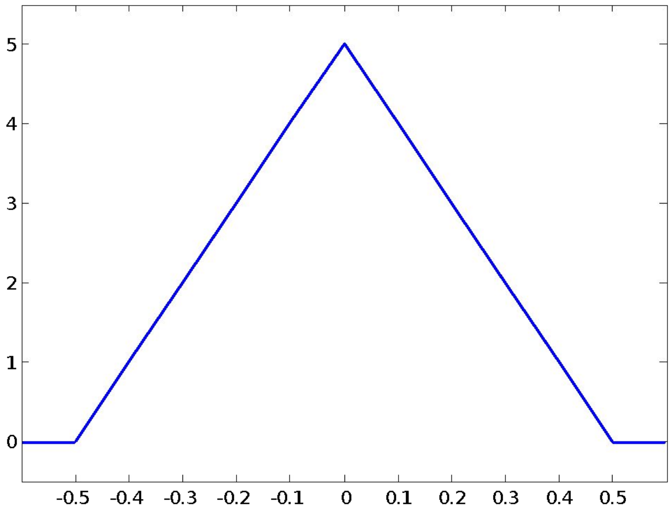
Interpolation function
(Reconstruction filter)

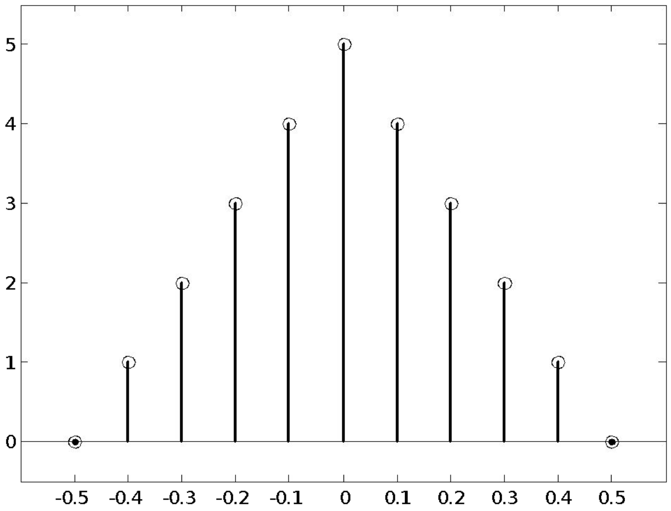
$$\boxed{g(t) = \frac{\sin 2\pi Bt}{2\pi Bt}}$$



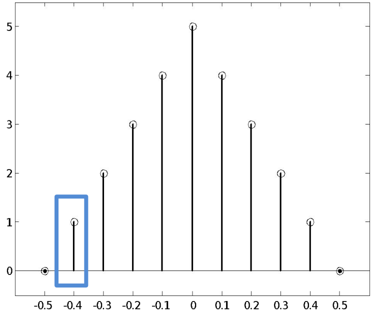
Example

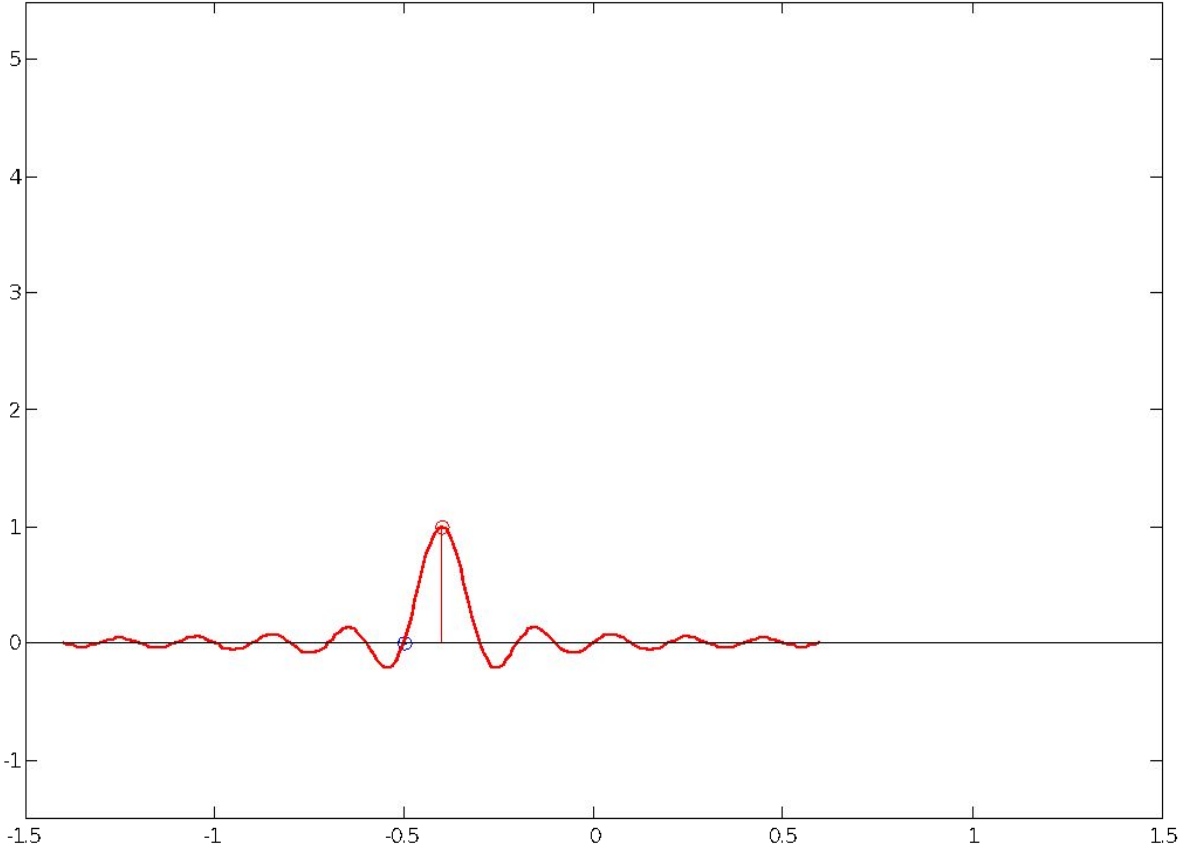
Reconstruct the sampled signal on the right



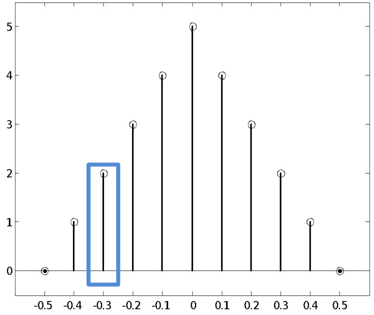


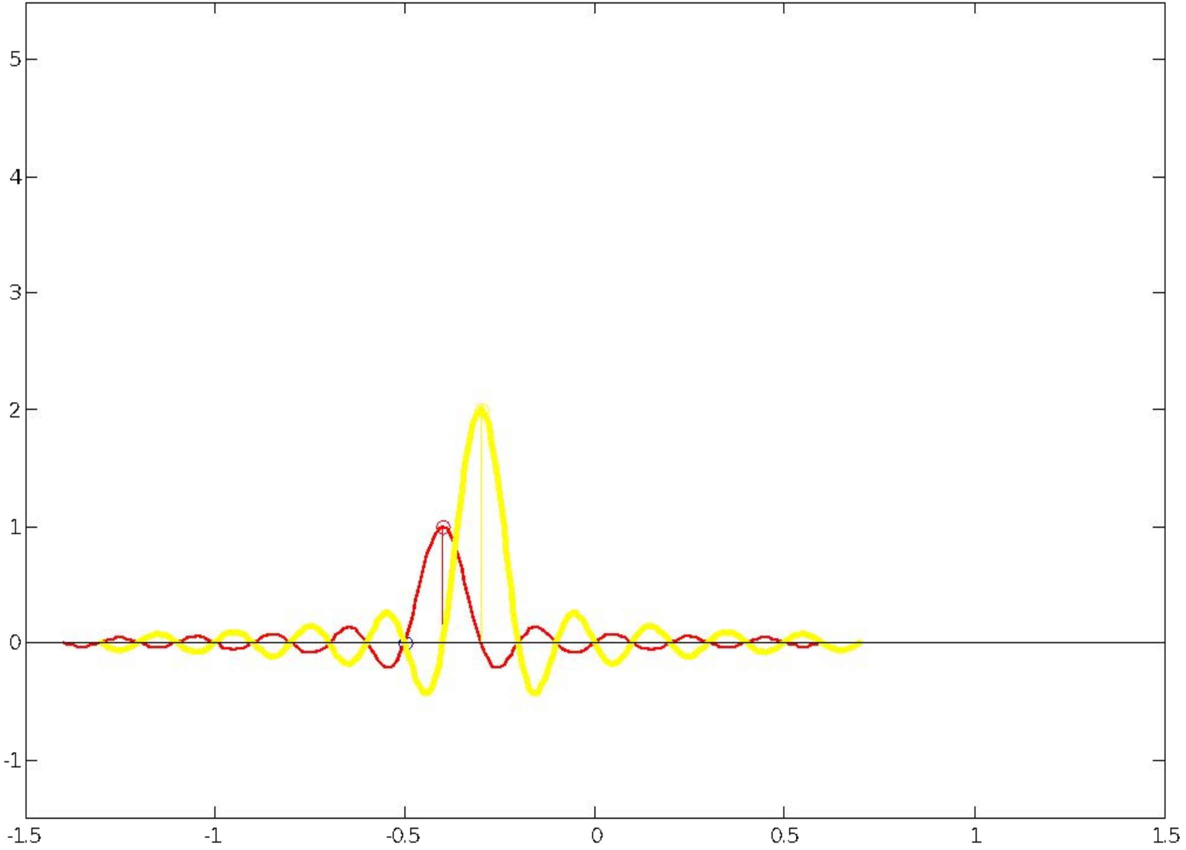
Example



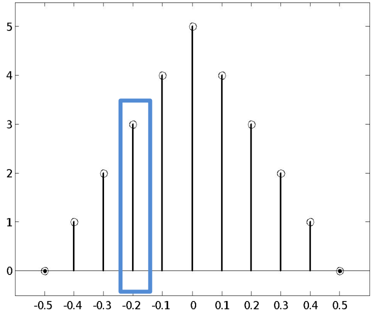


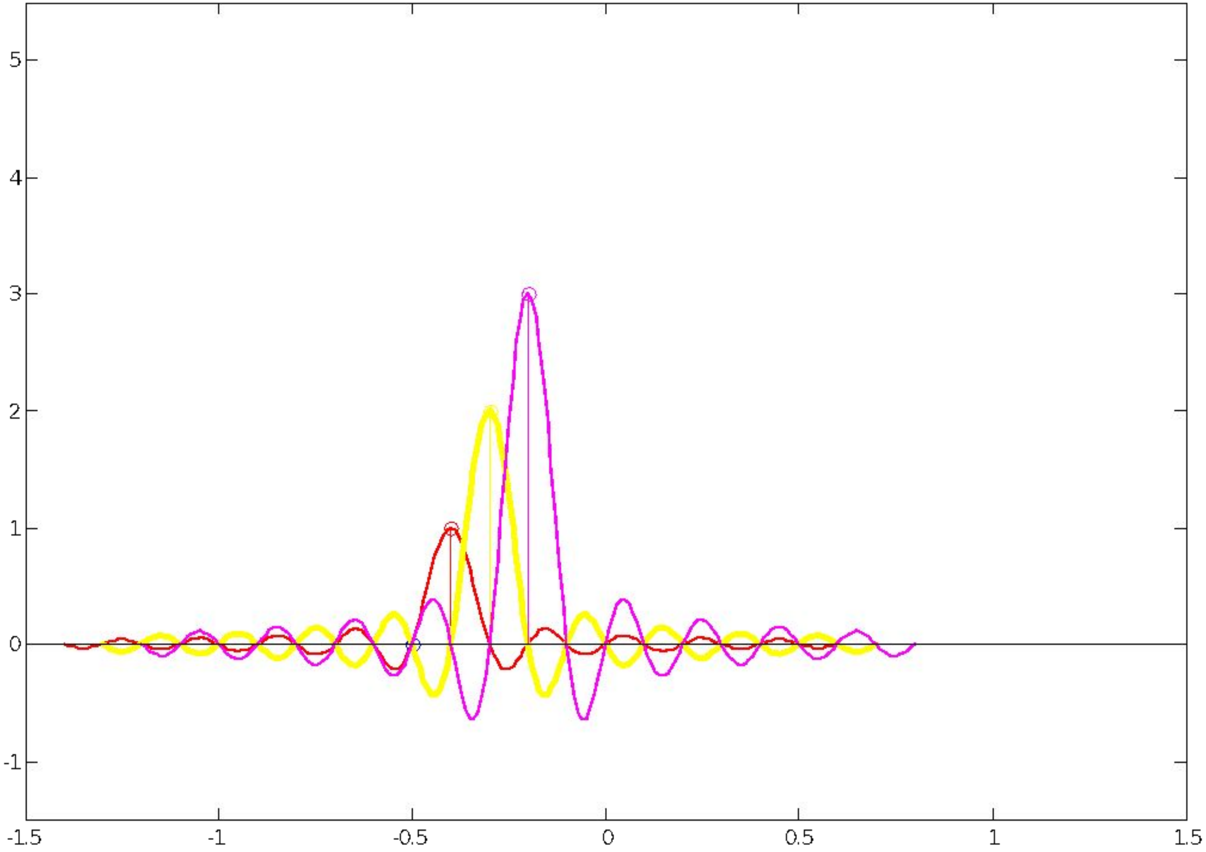
Example



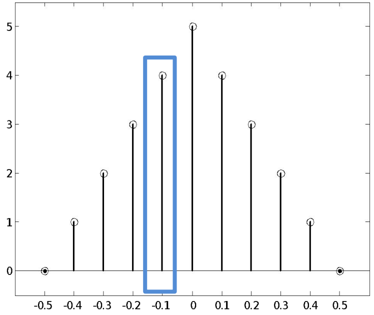


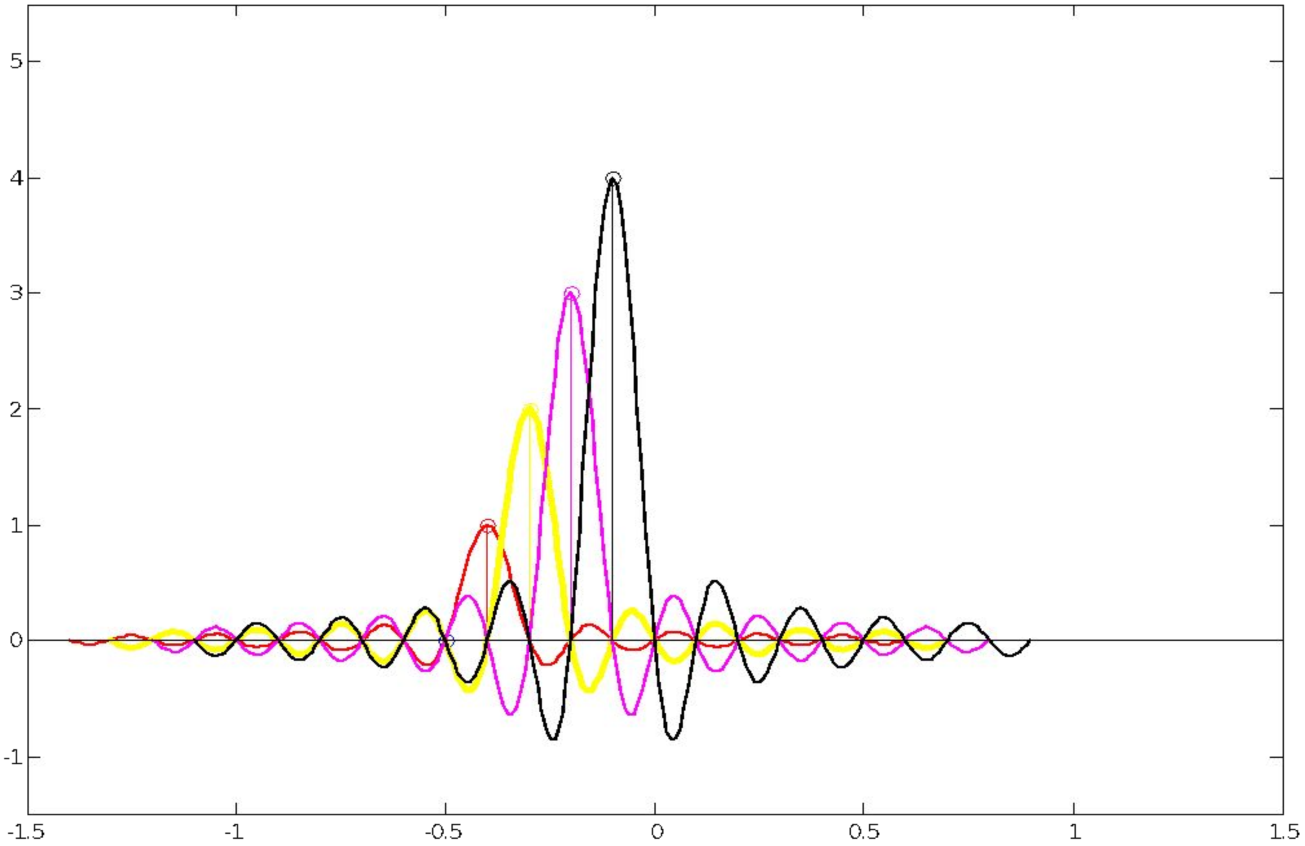
Example



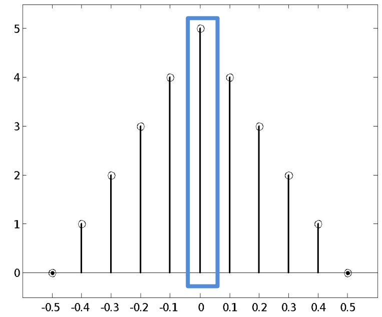


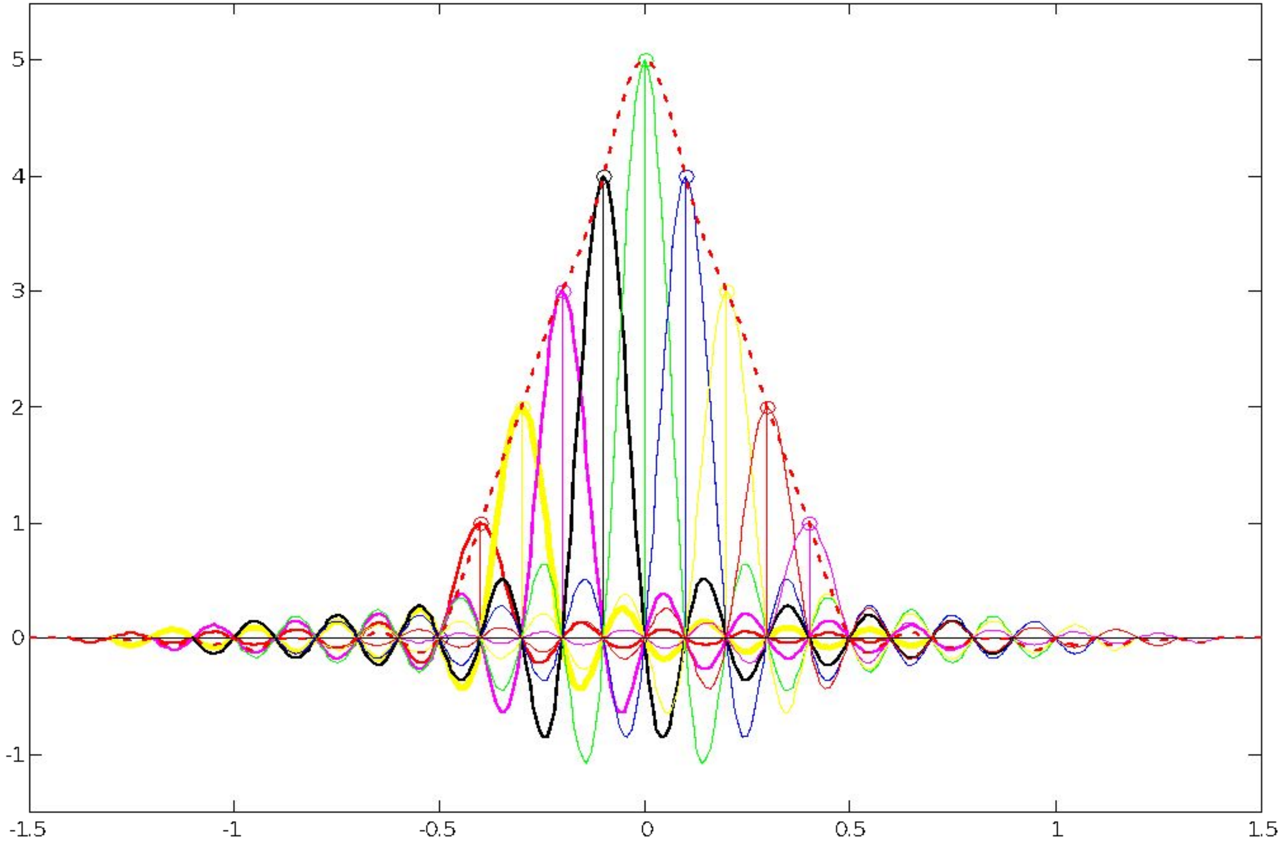
Example



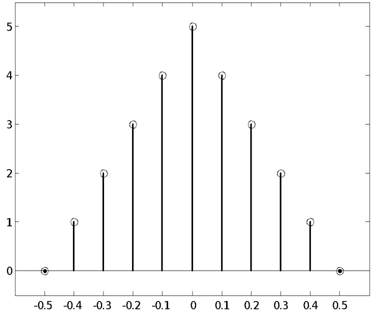


Example

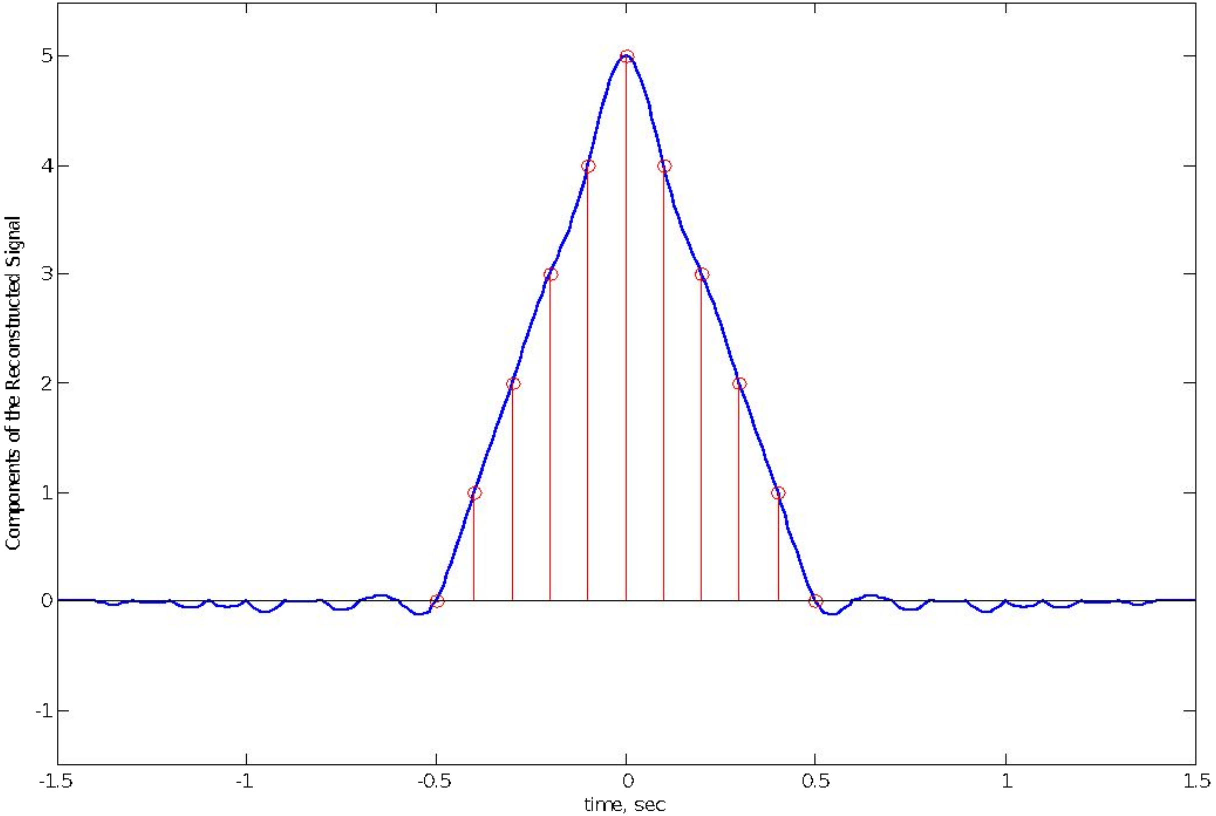




Example



5
4
3
2
1
0
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5



Components of the Reconstructed Signal
5
4
3
2
1
0
-1
-1.5 -1 -0.5 0 0.5 1 1.5
time, sec

Summary

- Discrete signals can be manipulated in several ways such as amplitude scaling, time shifting, and time scaling
- Mathematical operations such as addition and multiplication still applies to discrete time signals
- Convolution operation is applicable to discrete time signals
- In order for a DTS to be reconstructed back into CTS, it must be convolved with an interpolation function

For further reading...

- ➢ Chapters 1.4.6, 2.1.3, 2.3.3

  “Digital Signal Processing, 3rd ed. By Proakis, J. & Manolakis, D.”

- ➢ Chapter 1.2

  “Digital Signal Processing: A Computer-Based Approach by Mitra, S.”

Discrete Time Signal Operations

EE 274/COE 197E

Rhandley Cajote, Ph. D.
Crisron Lucas, MSc.