



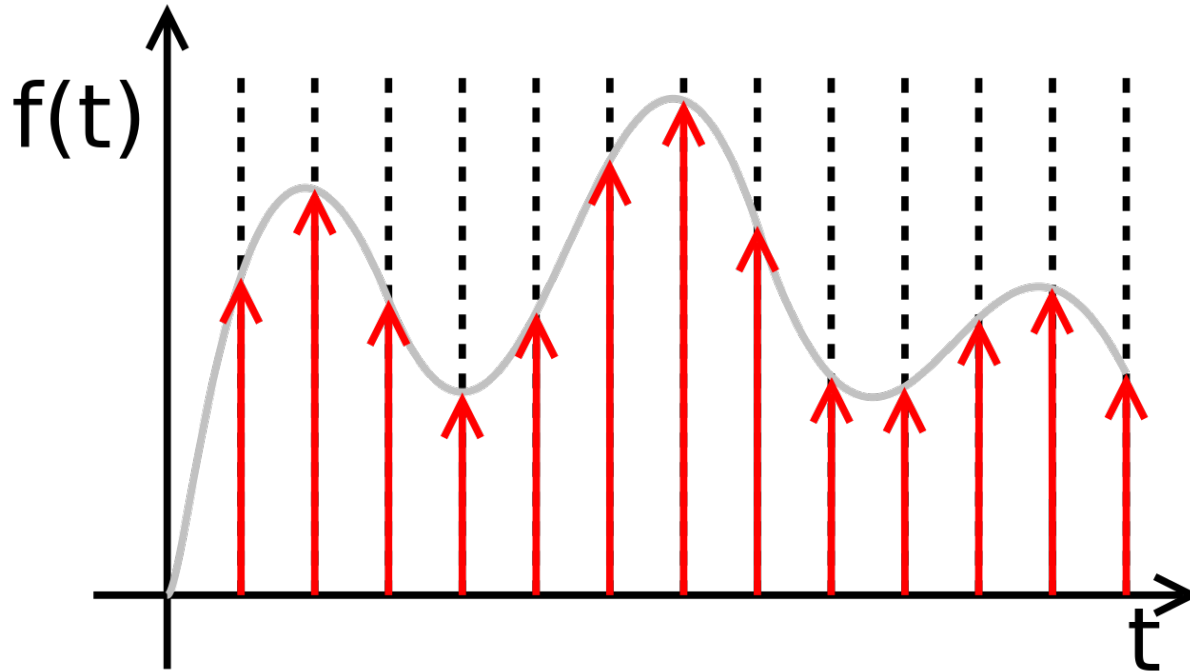
POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

1D discrete signals analysis

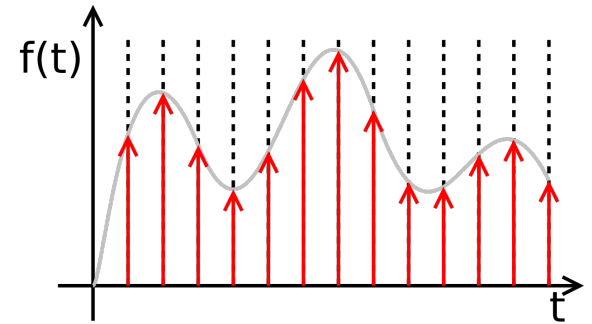
1D signals(t)

- x-axis represents time $\rightarrow y = f(t)$ is a time-variant signal

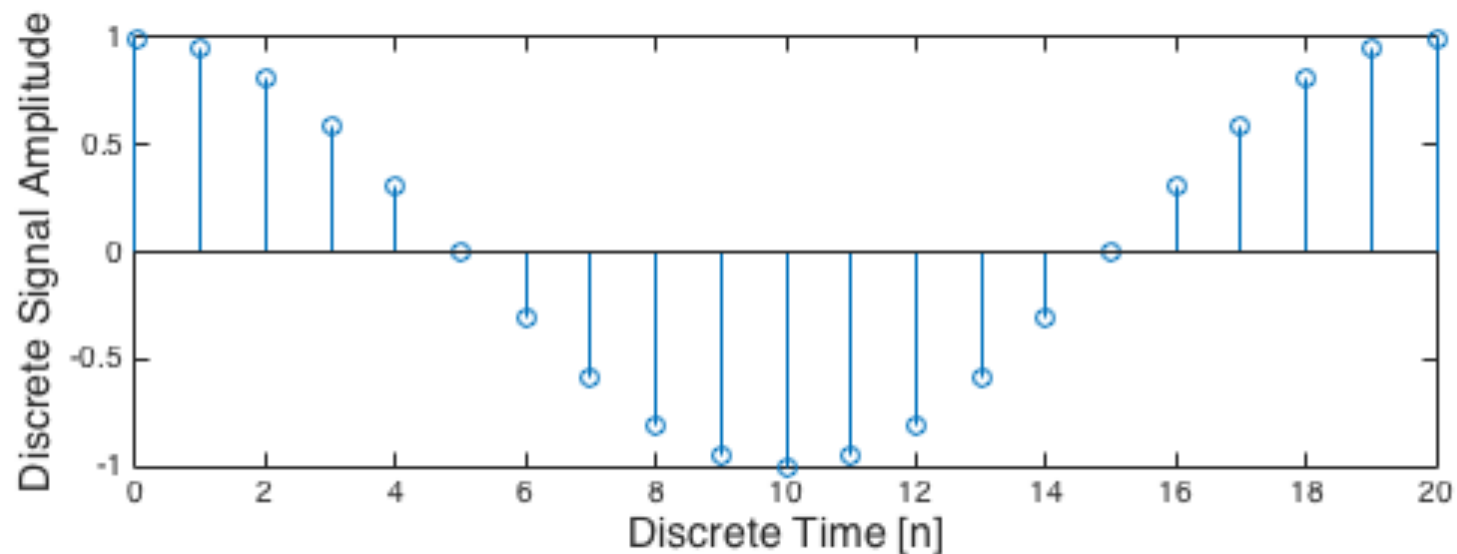
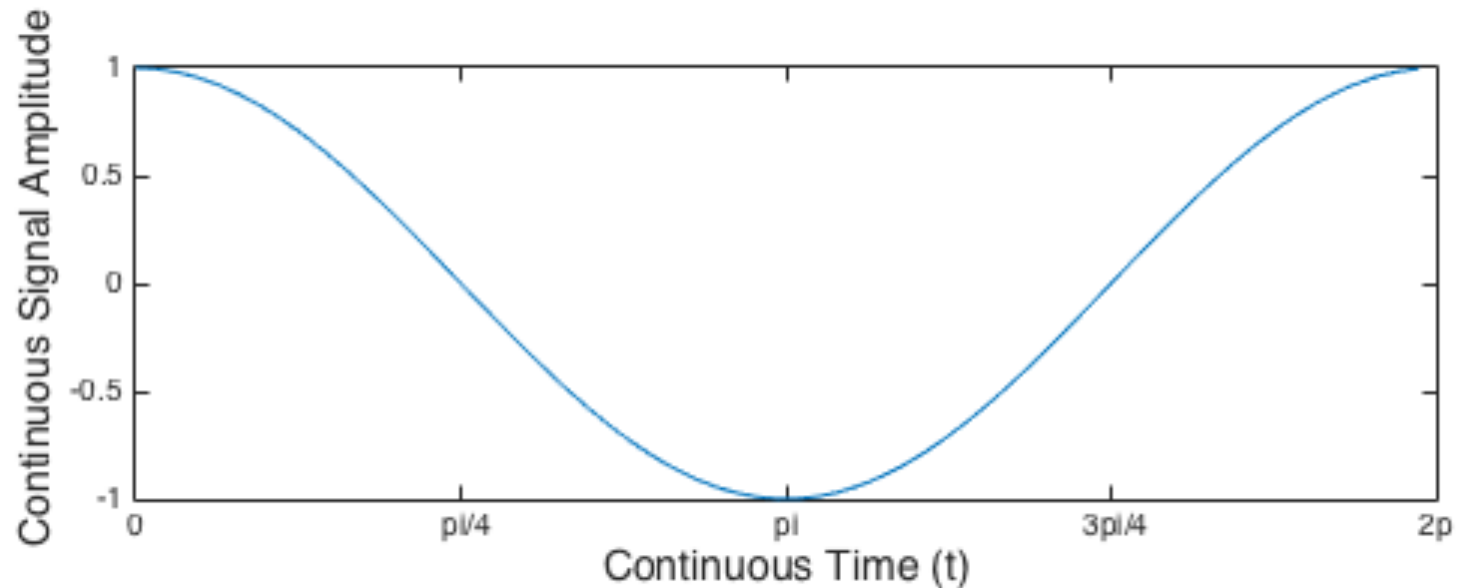


From continuous to discrete-time signals

- If we sample $y = f(t)$ every T_s time instants
- $t \rightarrow n \cdot T_s$, with $n = 0, 1, 2, \dots$
- $y_n = f(t_n)$
- T_s = sampling time or sampling period
- $F_s = \frac{1}{T_s}$ = sampling frequency or sampling rate
- We can represent y_n and t_n as two arrays with same number of elements and use *MATLAB* to process these signals

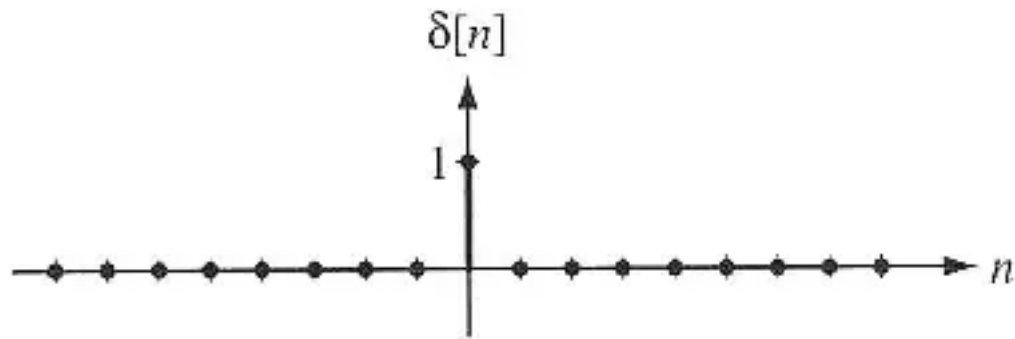


From continuous to discrete-time signals



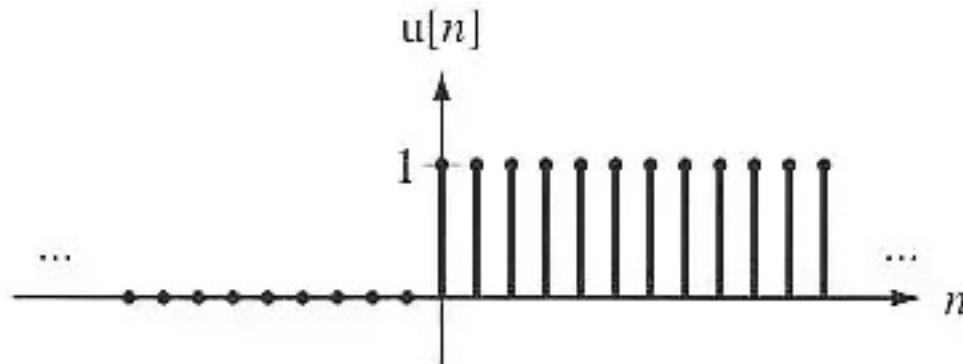
1D discrete impulse

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$



1D discrete unit step

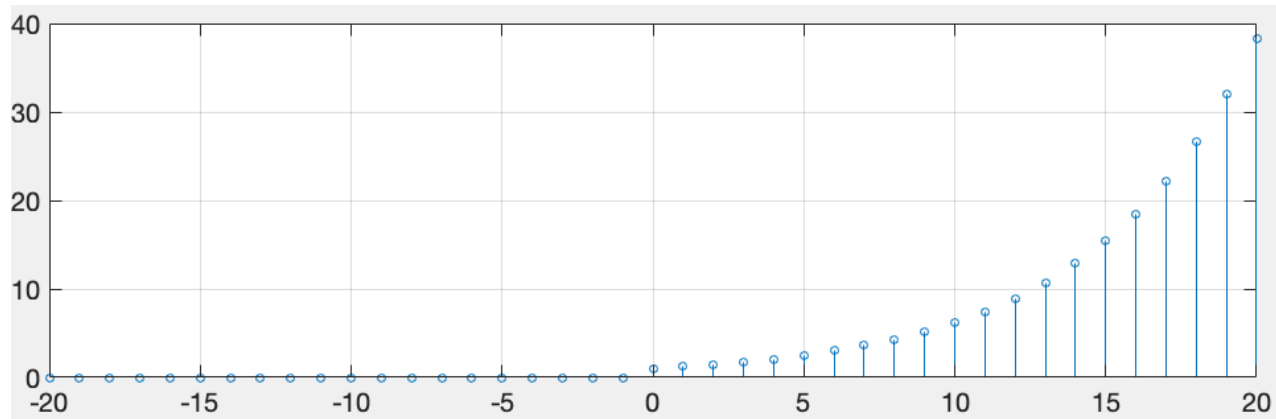
$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$



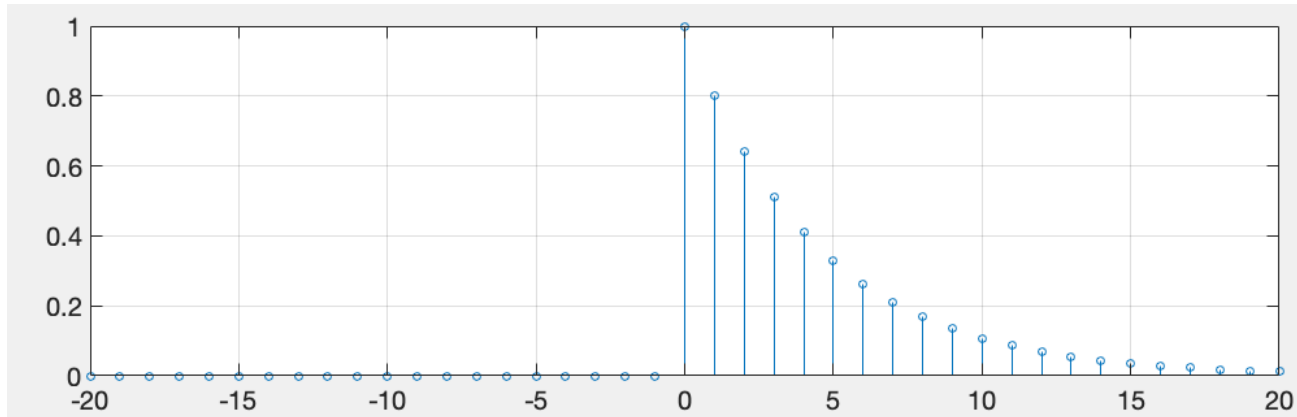
1D discrete exponential step

- $x(n) = a^n u(n)$

- $a = 1.2$

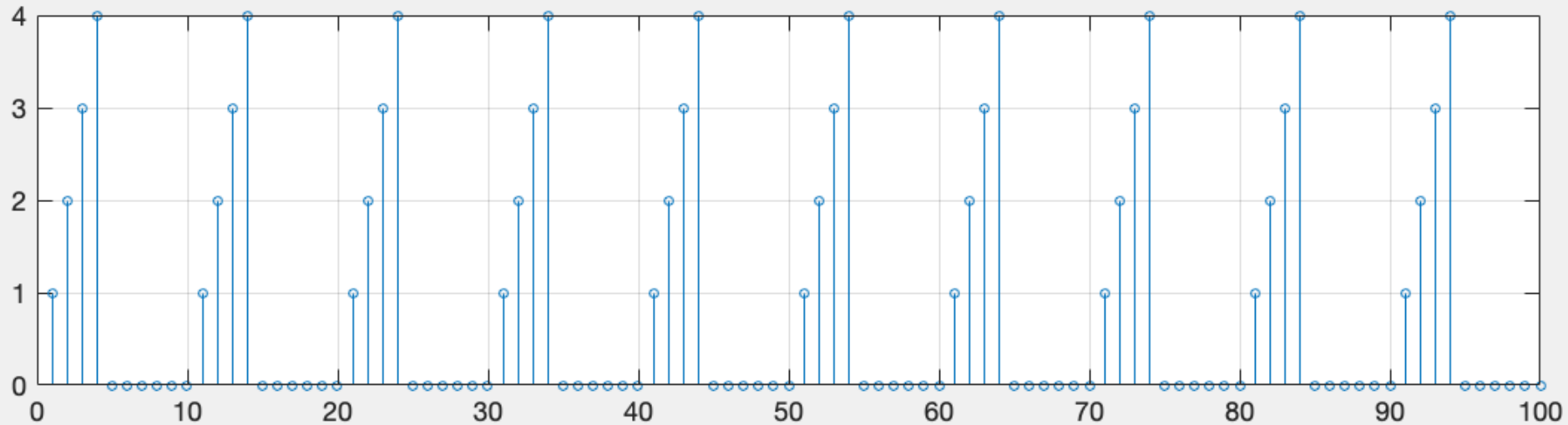


- $a = 0.8$



1D discrete periodic signals

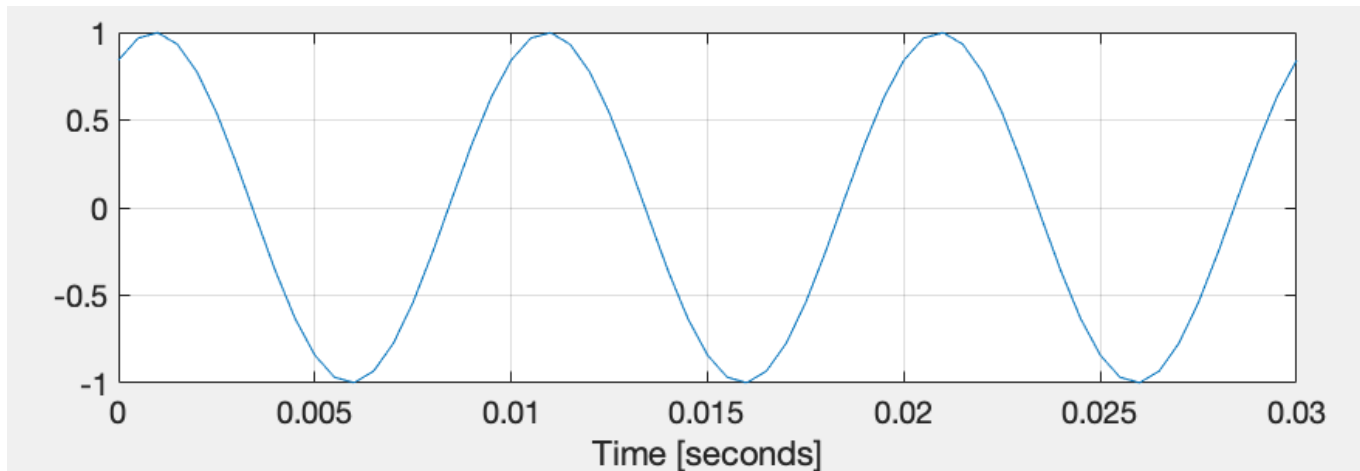
$$x(n) = x(n + kT) \quad \forall k \in \mathbb{Z}$$



$$T = 10$$

1D continuous-time sinusoids

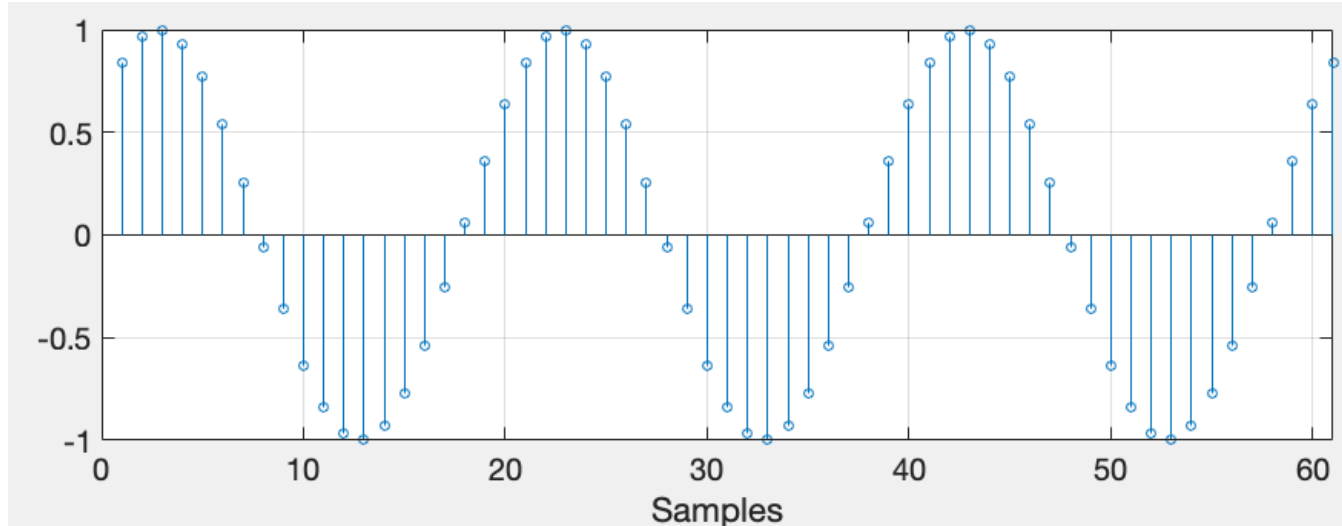
- $y(t) = A \cdot \cos(2\pi f_0 t + \phi)$



- A = amplitude
- f_0 = frequency [Hz]
- $\omega_0 = 2\pi f_0$ = angular frequency [rad/s]
- ϕ = phase [rad]
- $\frac{1}{f_0}$ = period of the sinusoid [s]

1D discrete-time sinusoids

- $y(t) = A \cdot \cos(2\pi f_o t + \phi) \rightarrow y(n) = A \cdot \cos(2\pi f_o T_s n + \phi)$



- A = amplitude
- f_0 = frequency [Hz]
- $\omega_0 = 2\pi f_0$ = angular frequency [rad/s]
- ϕ = phase [rad]
- $\frac{1}{f_0}$ = period of the sinusoid [s]
- $t = n \cdot T_s$
- $n = 0, 1, 2, \dots N$ = samples
- T_s = sampling time or sampling period
- $\frac{1}{T_s} = F_s$ = sampling rate

1D discrete-time sinusoids

- NB:

$$y(t) = A \cdot \cos(2\pi f_o t + \phi) \rightarrow y(n) = A \cdot \cos(2\pi f_o T_s n + \phi)$$

OR

$$y(t) = A \cdot \cos(2\pi f_o t + \phi) \rightarrow y(n) = A \cdot \cos(2\pi \tilde{f}_o n + \phi)$$

- $t = n \cdot T_s$
- $n = 0, 1, 2, \dots N = \text{samples}$
- $\tilde{f}_o = \text{normalized frequency} = f_o / F_s = f_o \cdot T_s$

How to define and plot 1D discrete-time sinusoids

```
close all
clearvars
clc
```

```
%% parameters
```

```
T_s = .001; % sampling time
T_f = .5; % temporal duration [seconds]
f_0 = 20; % sinusoid frequency
phi = .2; % phase
A = 1.3; % amplitude
```

```
%% temporal axis
```

```
t = 0:T_s:T_f;
```

```
%% y-axis
```

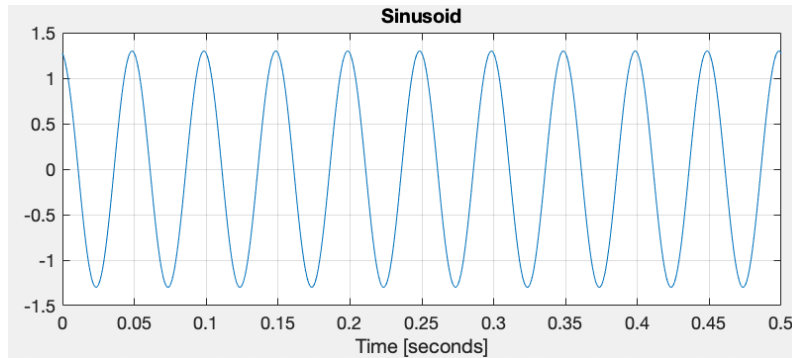
```
y = A*cos(2*pi*f_0*t + phi);
```

```
%% plot
```

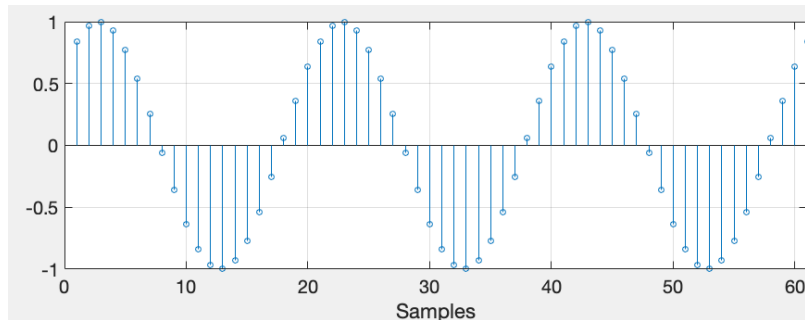
```
figure(1); % open new figure and call it Figure 1
plot(t, y); % --> NB: dimensions must be consistent!
grid; % insert a grid
title('Sinusoid'); % title
xlabel('Time [seconds]'); % label of x-axis
set(gca, 'fontsize', 18) % increase fontsize
```

How to plot 1D discrete-time sinusoids

- Use `plot(x-axis, y-axis)` or `plot(y-axis)` for a continuous line



- Use `stem(x-axis, y-axis)` for highlighting the single samples

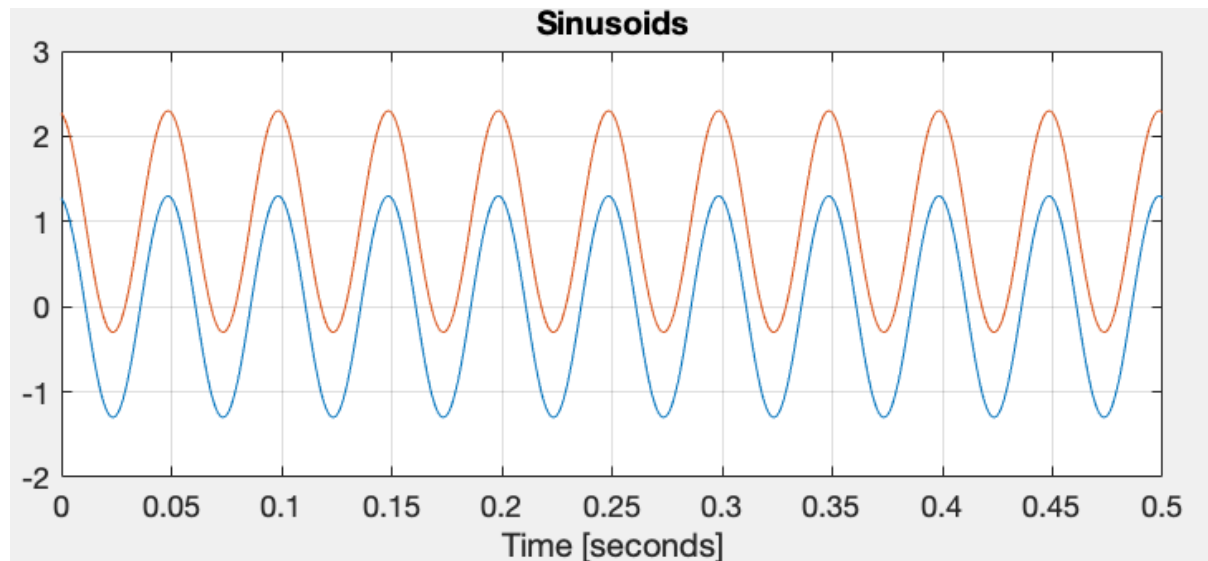


How to plot 1D discrete-time sinusoids

- 'hold on' allows to insert multiple plots into the same figure

```
figure(1); % open new figure and call it Figure 1
plot(t, y); % --> NB: dimensions must be consistent!
grid; % insert a grid
title('Sinusoids'); % title
xlabel('Time [seconds]'); % label of x-axis
set(gca, 'fontsize', 18) % increase fontsize
```

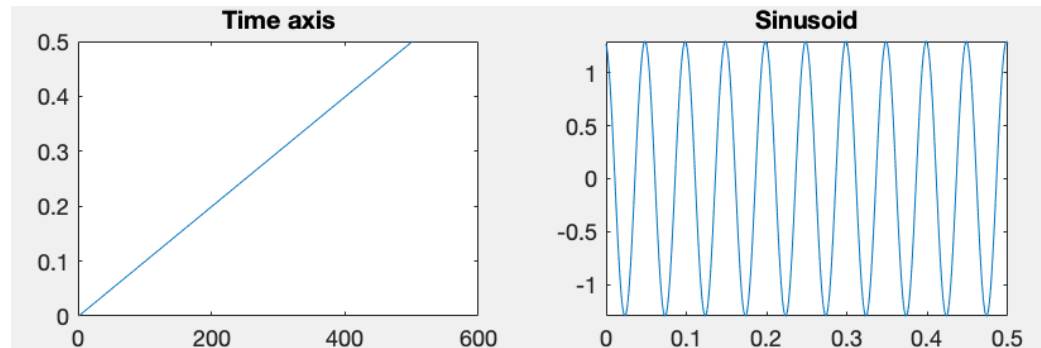
```
hold on,
plot(t, y + 1);
```



How to plot 1D discrete-time sinusoids

- Once a Figure has been opened, you can insert whatever you want:
 - 'xlabel' and 'ylabel'
 - 'title'
 - 'legend'
 - grid
 - markers, colors, linestyle etc...
- You can put multiple non-overlapping plots inside the same figure: 'subplot(#rows, #cols, #plot index)'

```
figure(2)
subplot(1, 2, 1)
plot(t)
title('Time axis');
set(gca, 'fontsize', 18)
subplot(1, 2, 2)
plot(t, y)
title('Sinusoid');
set(gca, 'fontsize', 18)
```





POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

Exercises

Exercise 1: discrete-time sinusoids

- Given the signal $x(t) = A \cos(2\pi f t + \phi)$
 1. Write the script 'ex1.m' to create the signal $x(n)$ as $x(t)$ from 0 to 0.5 seconds, sampled at F_s (sampling rate) = 1000Hz; $A = 0.8$, $f = 50\text{Hz}$, phase 30 deg.
 2. Write the function 'cosinusoid.m' which takes as input the time-axis, the amplitude, the frequency, the phase of a discrete cosinusoid and returns the signal.
 3. Generate the same signal as 1. with 'cosinusoid.m'
 4. In 'ex1.m', plot the signal as a function of samples.
 5. In 'ex1.m', plot the signal as a function of time.

Exercise 2: discrete-time sinusoids

- Build a signal $x(n)$ as the sum of three different sinusoids at the normalized angular frequencies $\tilde{\omega}_1 = \pi/5$, $\tilde{\omega}_2 = \pi/8$, $\tilde{\omega}_3 = \pi/4$. The sampling period is $T = 0.3$ seconds, and the signal is defined for time t in $[0, 100]$ seconds.
- Plot the signal as a function of time.
- Compute the period P (in seconds) for each of the three sinusoids.
- Which is the period of the sinusoids in number of samples?
- Which is the period of $x(n)$?

Exercise 3: previous exams

- 18/02/2021: [2 pt] You are given three cosinusoidal signals: x , y and z . In particular, x has frequency = 1KHz, y has frequency = 1.6KHz and z has frequency = 8KHz. One entire cycle of the signal z is completed every 10 signal samples. The signal w is the summation of the three sinusoids, and the signals' duration is equal to one period of w . Define the signals x , y , z , w .
- 30/08/2021: [2 pt] A signal x with length = 10000 samples is the contribution of three cosinusoidal signals and has a period of 300 samples. The three signals have normalized frequencies f_0 , $f_1=f_0/25$, $f_2=f_0/3$, and have all the same amplitude = 0.25. Define the signal x .

Exercise 4: signal shift

- Generate the signal $x(n) = (0.8)^n u(n)$, $n = 1:20$
- Generate the signal $y1(n) = x(n-5)$, $n = 1:20$
- Generate the signal $y2(n) = x(n+5)$, $n = 1:20$
- Hint: Consider using 'circshift' instead of for loops.
- Plot the signals in the same figure.

Exercise 5: periodic sequences

- Generate the signal $x(n) = u(n-5) - u(n-10)$, considering $n = 1:15$.
- Generate the periodic signal $x_p(n)$ with period $N = 15$, considering $n = 1:200$.
- Hint: Consider using 'repmat' instead of for loops.
- Plot the periodic signal $x_p(n)$ considering only 8 periods.



POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

Linear Time-Invariant Systems (LTI)

Definition of LTI

- The defining properties of any LTI system are *linearity* and *time invariance*.
 - Linearity = input-output relationship is LINEAR
 - Time invariance = the output does not depend on the particular time the input is applied. If the output due to $x(t)$ is $y(t)$, the output due to $x(t-k)$ is $y(t-k)$.
- The system can be completely characterized by its impulse response $h(t)$.

Output of LTI discrete systems

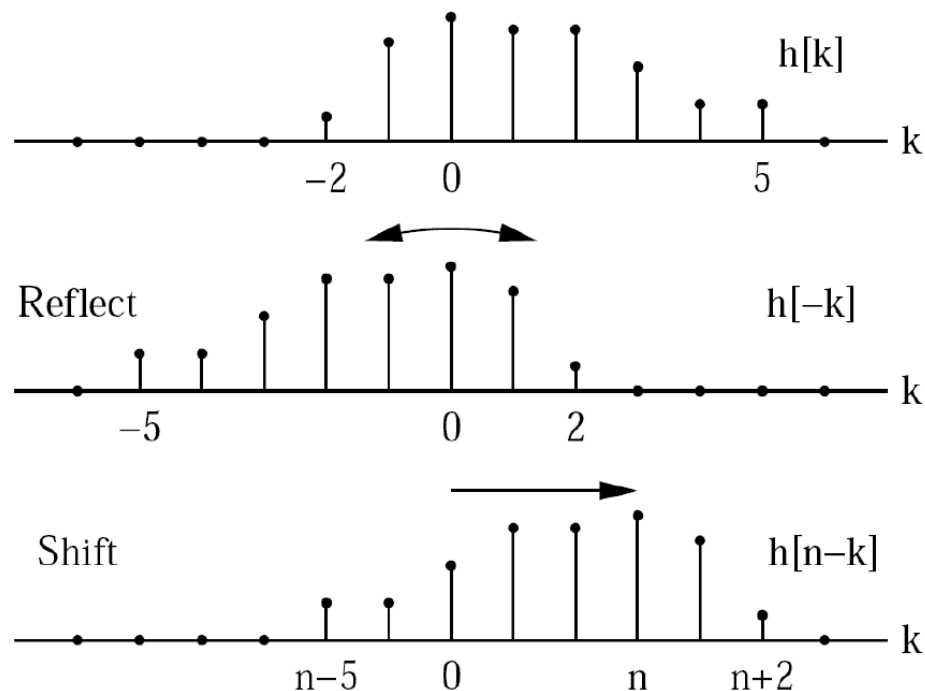
- The output of LTI discrete systems is always the convolution between the input signal and the impulse response $h(n)$.

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

Discrete signal convolution

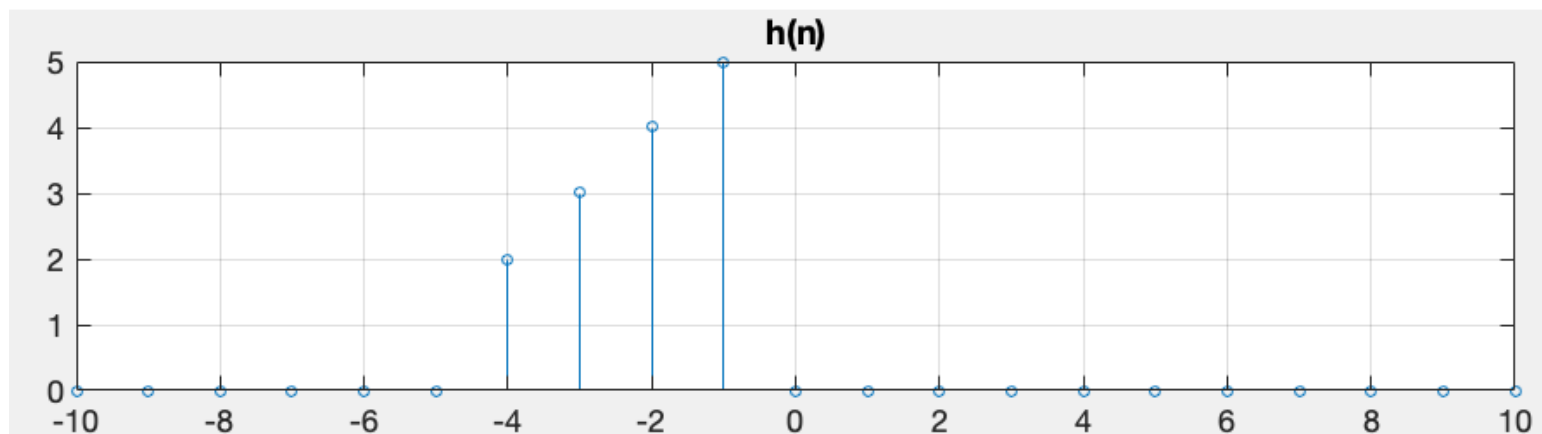
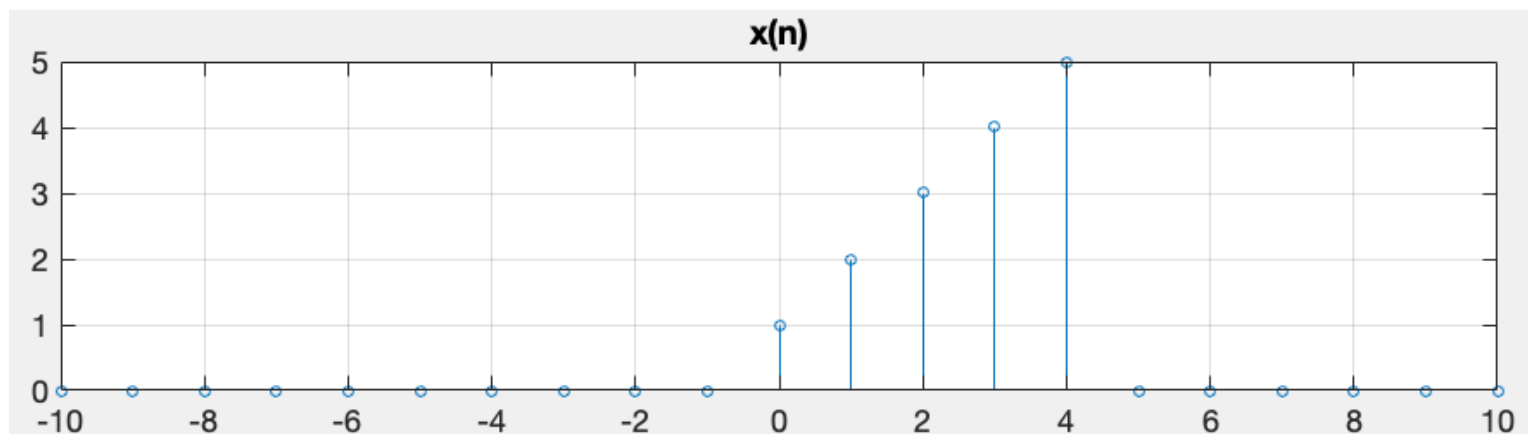
$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k) h(n-k)$$

NB: $h(n-k) = h(-(k-n)) \rightarrow$ Operation order:



Discrete signal convolution: example

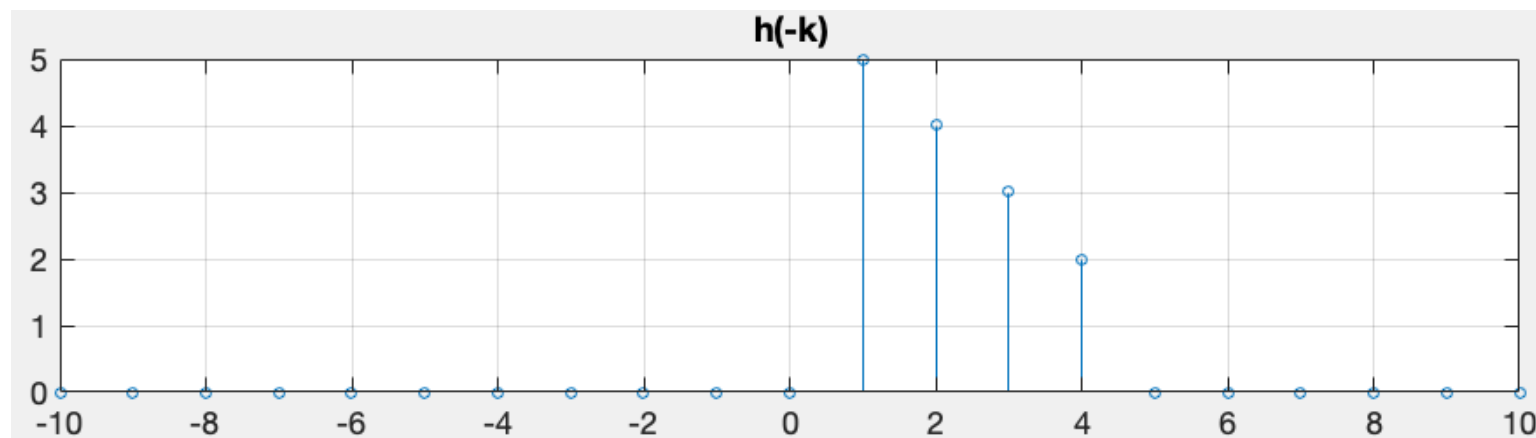
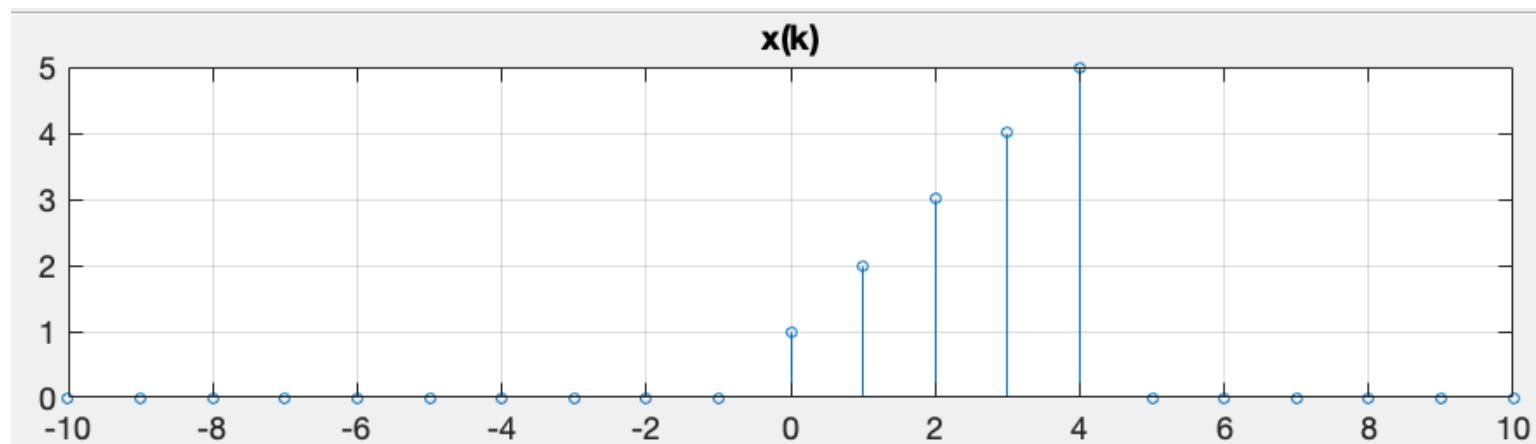
$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k) h(n - k)$$



Discrete signal convolution: example

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

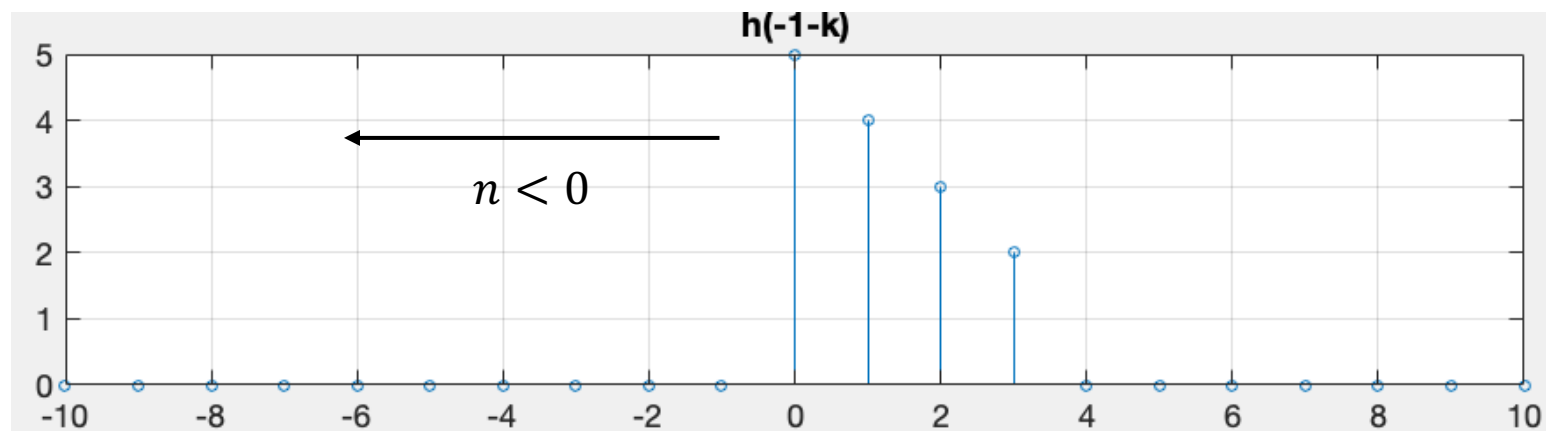
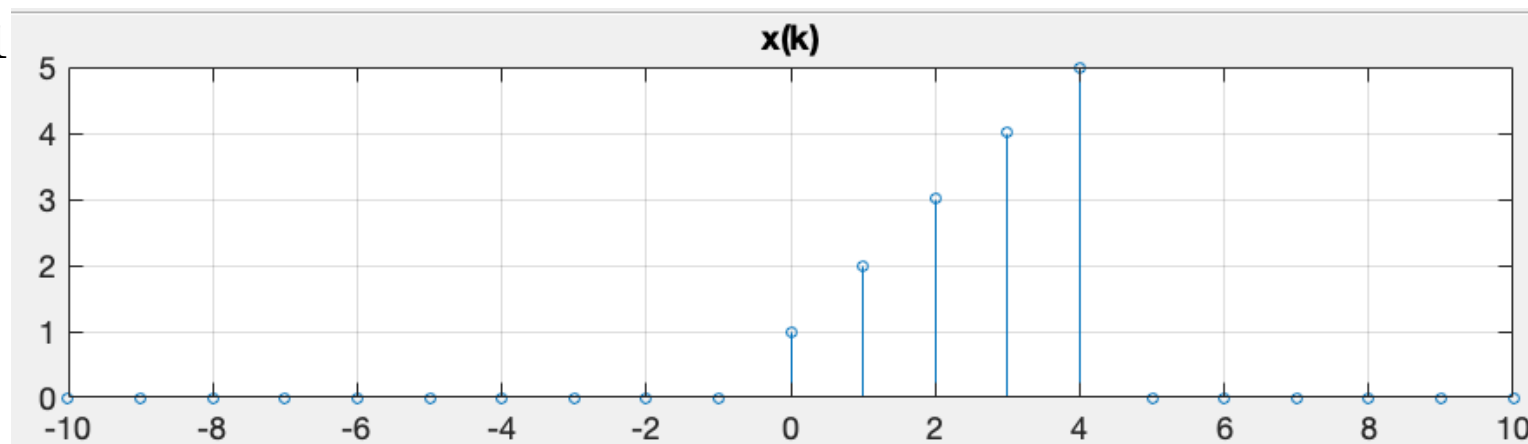
$n = 0$



Discrete signal convolution: example

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

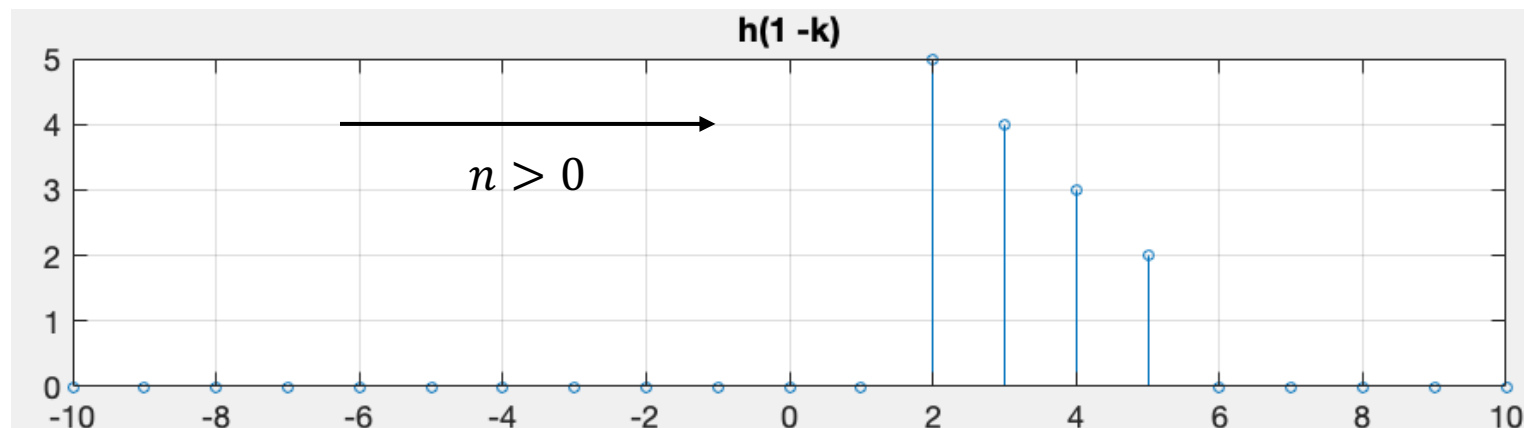
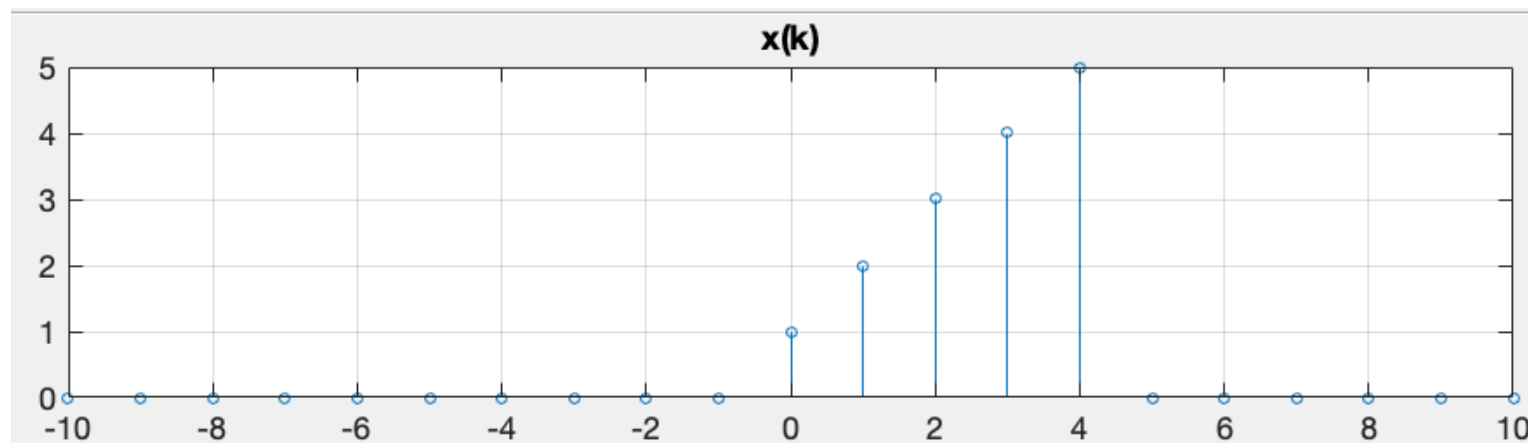
$n = -1$



Discrete signal convolution: example

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

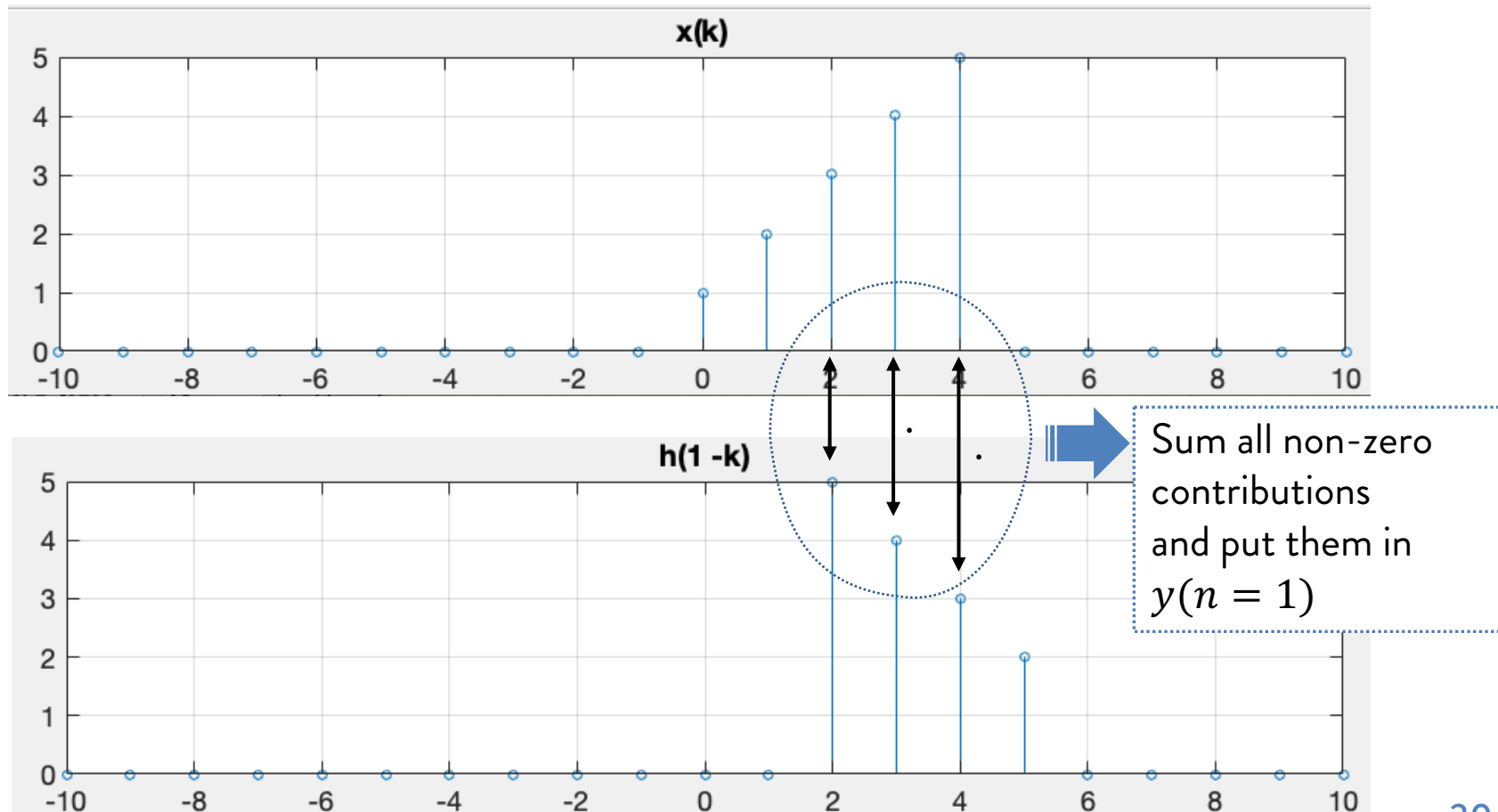
$n = 1$



Discrete signal convolution: example

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

$n = 1$



Properties of convolution

- Commutativity: $x(n) * y(n) = y(n) * x(n)$
- Associativity: $(x(n) * y(n)) * z(n) = x(n) * (y(n) * z(n))$
- Distributivity: $(x(n) + y(n)) * z(n) = x(n) * z(n) + y(n) * z(n)$
- Convolution by pulse: $x(n) * \delta(n) = x(n)$
- Convolution by a shifted pulse : $x(n) * \delta(n - k) = x(n - k)$

Exercise 6: Convolution

- Given $x(n) = [3, 11, 7, 0, -1, 4, 2]$, n in $[-3, 3]$
- Given $h(n) = [2, 3, 0, -5, 2, 1]$, n in $[-1, 4]$
- Define both signals for n in $[-7, 7]$.
- Compute $y(n)$ as $x(n)$ convolved with $h(n)$, n in $[-7, 7]$.
- Use also the MATLAB function 'conv'.
- Which is the support of the convolution?
- Which is the length of the convolution?

Operations on signals

- Discrete delay $\rightarrow y(n) = x(n - k)$

- Moving average $\rightarrow y(n) = \frac{1}{M} \sum_{m=0}^{M-1} x(n - m)$



$y(n)$ can be seen as the output of LTI systems

Exercise 7: LTI systems

- Given $x(n) = [3, 11, 7, 0, -1, 4, 2]$, n in $[-3, 3]$
- Create $y(n) = x(n - 5)$, n in $[0, 10]$, without using 'circshift' or 'for' loops.
- Create $y(n) = \frac{1}{3} \sum_{m=0}^2 x(n - m)$, n in $[0, 10]$, without using 'circshift' or 'for' loops.
- Hint: $y(n)$ has the form of a convolution... (you can use MATLAB function 'conv').