CONVCOS

```
close all
clear
F=1/10;
N=20;
x=cos(2*pi*F*(0:N-1));
M=2;
h=ones(1,M)*1/M;
y=conv(x,h);
figure(1)
stem((0:N-1),x)
hold on
if floor(M/2)*2==M
    L=M/2+1
else
    L=(M+1)/2
end
stem((0:N-1),y(L:N-1+L),'r')
% same as y=conv(x,h,'same')
hold off
% help floor
% CONVOLUTION EXAMPLE WITH MATLAB
% Author C. Prati
% February 17, 2020
clear
close all

%---------------------------------------------------
% 1
% Let us generate a discrete cosinusoidal signal x with
% frequency of fo=40Hz, N=200 samples using a sampling interval
% T=1/1000 second

N=200;
fo=40;
T=1/1000;
t=(0:N-1)*T;
x=cos(2*pi*fo*t);

%-----------------------------------------------------
% 2
% Let us plot the discrete signal x in figure 1
% as a sequence of samples ... from 0 to N-1
figure(1)
stem((0:N-1),x)
title('sequence of samples')
pause
% as a sequence on samples along a time axis t
figure(2)
stem(t,x)
xlabel('time (seconds)')
pause
% as a continuous line connecting the samples
figure(3)
plot(t,x)
pause
%-----------------------------------------------------
% 3
% Let us generate a very simple High Pass Impulse response
% h formed by M=3 samples -1/4 1/2 -1/4
M=3;
h=[-1/4 1/2 -1/4];

%------------------------------------------------------
% 4
% Let us convolve the signal x with the impulse response h
% forming the output signal y
y=conv(x,h);
pause

%-----------------------------------------------------
% 5
% Let us plot the discrete signal y of length N+M-1 in
% figure 4 as a sequence of samples ... from 0 to N+M-2
% then as a sequence on samples along a time axis t and
% finally as a continuous line connecting the samples
figure(4)
stem((0:N+M-2),y)
pause
stem((0:N+M-2)*T,y)
pause
plot((0:N+M-2)*T,y)
pause

%-----------------------------------------------------
% 6
% Let us repeat the previous convolution example using as
% imput signal x the sum of 2 cosinusoids with frequencies
% f1=20Hz and f2=400Hz. The impulse response h is now Low-pass
% h=[1/4 1/2 1/4] or High-passh=[-1/4 1/2 -1/4]
f1=20;
f2=400;
x=cos(2*pi*f1*t)+cos(2*pi*f2*t);
figure(5)
stem(t,x)
pause
h=[1/4 1/2 1/4];
y=conv(x,h);
figure(6)
stem((0:N+M-2)*T,y)
```

MediaMobile

```
% Esegue una media mobile di 1000 campioni
% su una sequenza di N campioni costituita
% da una cosinusoide a bassissima frequenza
%
% La media mobile e' eseguita tramite convoluzione con una finestra
% rettangolare di 1000 campioni di valore 1/1000.

close all
clear all
N=11000;
f0=1/5000;
n=1:N;
x=0.25*cos(2*pi*f0*n)+randn(1,N);
plot((1:N-1000),x(501:N-500))
%h=2*ones(1,1000)/1000.*cos(2*pi*f0*(0:999));
h=ones(1,1000)/1000;
y=conv(x,h);
pause
hold on
plot((1:N-1000),y(1001:N),'r')
pause
axis([1 N-1000 -1 1])
```

```
% Filtering Basics
% Author C. Prati
% February 17, 2020
clear
close all
%---------------------------------------------------
% 1
% Let us generate a discrete impulse response hLP of a Low-Pass
Filter
% The length of the filter in  N=64 samples
% The cut-off frequency is fc=200Hz and the sampling frequency is
% fs=2KHz
% The matlab function used to generate h is fir1:
% The general format is hLP = fir1(N,W) where W is the normalized
cut-off
% frequency: W must be between 0 < W < 1.0, with 1.0 corresponding to
half
% the sampling  frequency.

N=64;
fc=200;
fs=2000;
W=fc/fs*2;

hLP=fir1(N,W);

figure(1)
stem((0:N),hLP,'filled')
title('Low Pass impulse response')

pause

%---------------------------------------------------
% 2
% Let us generate a discrete impulse response hHP of a High-Pass
Filter
% The length of the filter in N=64 samples
% The cut-off frequency is fc=200Hz and the sampling frequency is
% fs=2KzH.
% The matlab function used to generate h is again fir1:
% The general format is hHP = fir1(N,W,'high') where W is the
normalized cut-off
% W must be between 0 < W < 1.0, with 1.0 corresponding to half the
sampling  frequency.

hHP=fir1(N,W,'high');

figure(1)
```

```
clf
stem((0:N),hHP,'filled')
title('high Pass impulse response')

pause




%---------------------------------------------------
% 3
% Let us generate a discrete impulse response hBP of a Band-Pass
Filter
% The length of the filter in N=64 samples
% The lower and higher cut-off frequencies are f1=100Hz and f2=200Hz
% the sampling frequency is fs=2KzH.
% The matlab function used to generate h is again fir1:
% The general format is hBP = fir1(N,[W1 W2],'bandpass') where W1 ans
W2
% are the normalized cut-off frequencies. W1 and W2 must be between 0
< W < 1.0,
% with 1.0 corresponding to half the sampling  frequency.
%---------------------------------------------------

N=64;
f1=100;
f2=200;
fs=2000;
W1=f1/fs*2;
W2=f2/fs*2;

hBP=fir1(N,[W1 W2],'bandpass');

figure(1)
clf
stem((0:N),hBP,'filled')
title('Band Pass impulse response')

pause


%---------------------------------------------------
% 4
% Let us generate a chirp signal x using a sampling frequency
fs=2kHz:
% a chirp is a cosinusoid with frequency that linearly varies with
time.
% x = chirp(t,f0,t1,f1) generates samples of a linear swept-frequency
% signal at the time instances defined in array t.  The instantaneous
% frequency at time 0 is f0 Hertz.  The instantaneous frequency f1
% is achieved at time t1.  By default, f0=0, t1=1, and f1=100.
%---------------------------------------------------

T=1/fs;
t=(0:T:2);
f0=10;
f1=300;
t1=2;
x=chirp(t,f0,t1,f1);
figure(1)
clf
plot(t,x)
pause

%---------------------------------------------------
% 5
% Let us plot the signal y obtained by filtering the chirp x with
% a) the LP filter
% b) the HP filter
% c) the BP filter
%-----------------------------------------------------

yLP=conv(x,hLP,'same');
yHP=conv(x,hHP,'same');
yBP=conv(x,hBP,'same');
figure(2)
clf
plot(t,yLP)
pause
plot(t,yHP)
pause
plot(t,yBP)

pause

%---------------------------------------------------
% 6
% Let us generate boxcar signal x and let's filter it with
% the low-pass, high-pass and bandpass filters

x=zeros(1,500);
x(200:300)=1;
figure(1)
clf
plot(x)
axis([1 500 -.3 1.2])
```

```
pause
yLP=conv(x,hLP,'same');
yHP=conv(x,hHP,'same');
yBP=conv(x,hBP,'same');
figure(2)
clf
plot(x,'r')
hold on
plot(yLP,'b')
axis([1 500 -.3 1.2])
pause
hold off
plot(x,'r')
hold on
plot(yHP,'b')
axis([1 500 -.3 1.2])
pause
hold off
plot(x,'r')
hold on
plot(yBP,'b')
axis([1 500 -.3 1.2])
hold off
```

Filtering 2D

```
%%%%%%%%%%%%%%%
%--------------------------------------------------
% 1
% Let us load a 2D image (e.g.in jpeg format)
%
%--------------------------------------------------

IMAGE=imread('gatto.jpg');
%IMAGE=imread('GEMSclass.jpg');
X=(IMAGE(:,:,1));
%X=(IMAGE(:,:,1));
figure(1)
imagesc(X)
colormap gray
axis('image')
title('original')
pause



%--------------------------------------------------
% 2
% Let us generate a 2D low-pass filter by using the
% fir1 Matlab function in 1D and then the function ftrans2
% which produces filters that are approximately circularly
% The normalized "radial" cut-off frequency is fc=0.1
% The filter length is N=35
%--------------------------------------------------
fc=0.1;
Wc=fc*2;
N=34;
hLP=fir1(N,Wc);
hLP_2D=ftrans2(hLP);
figure(2)
imagesc(hLP_2D)
colormap jet
axis('image')
pause

%--------------------------------------------------
% 3
% Let us apply the 2D low-pass filter to the original image
%--------------------------------------------------

XLP=conv2(X,hLP_2D,'same');
figure(2)
```

```
imagesc(XLP)
colormap gray
axis('image')
title('LP filtered')
pause

%---------------------------------------------------
% 4
% Let us generate a 2D high-pass filter by using the
% fir1 Matlab function in 1D and then the function ftrans2
% which produces filters that are approximately circularly
% The normalized "radial" cut-off frequency is fc=0.05
% The filter length is N=35
%---------------------------------------------------
fc=0.05;
Wc=fc*2;
N=34;
hHP=fir1(N,Wc,'high');
hHP_2D=ftrans2(hHP);
figure(2)
imagesc(log(abs(hHP_2D)))
colormap jet
axis('image')
pause

%---------------------------------------------------
% 5
% Let us apply the 2D high-pass filter to the original image
%---------------------------------------------------

XHP=conv2(X,hHP_2D,'same');
figure(2)
imagesc(XHP)
colormap gray
axis('image')
title('HP filtered')
```

```
% Fourier Transform Basics
% Author C. Prati
% February 17, 2020
clear
close all
%------------------------------------------------------
% 1
% Let us generate a discrete cosinusoidal signal x with
% frequency of fo=40Hz, N=50 samples using a sampling interval
% T=1/1000 second

N=50;
fo=40;
T=1/1000;
t=(0:N-1)*T;
x=cos(2*pi*fo*t);

%--------------------------------------------------------
% 2
% Let us plot the discrete signal x in figure 1
% as a sequence of samples ... from 0 to N-1
figure(1)
subplot(211)
stem((0:N-1),x)
xlabel('samples')
pause
% as a sequence of samples along a time axis t
figure(1)
subplot(212)
stem(t,x)
xlabel('time (seconds)')
pause

%--------------------------------------------------------
% 3
% Let us generate the Fourier Transform X of the discrete signal x
X=fft(x);

%--------------------------------------------------------
% 4
% Let us plot the absolute value and the phase of X in figure 2
% as a sequence of samples from 0 to N-1

figure(2)
subplot(211)
stem((0:N-1),abs(X))
xlabel('samples')
```

```
title('absolute value')
pause

figure(2)
subplot(212)
stem((0:N-1),angle(X))
xlabel('samples')
title('phase')
pause

%-----------------------------------------------------
% 5
% Let us plot the absolute value of X in figure 3
% as a sequence of samples and as a sequence along a
% normalized frequency axis and as sequence along a
% frequency axis

figure(3)
subplot(311)
stem((0:N-1),abs(X))
xlabel('samples')
pause

subplot(312)
stem((0:N-1)/N,abs(X))
xlabel('normalized frequency')
pause

subplot(313)
stem((0:N-1)/N/T,abs(X))
xlabel('frequency (Hz)')
pause

%-----------------------------------------------------
% 6
% Let us generate the FFT sequence centered around
% the zero frequency

Xshift=fftshift(X);

%-----------------------------------------------------
% 7
% Let us plot the absolute value of X in figure 4
% as a sequence of samples from 0 to N-1 and as a sequence
% of samples from -N/2 to N/2-1

figure(4)
subplot(211)
```

```
stem((0:N-1),abs(X))
xlabel('samples')
pause

subplot(212)
stem((-N/2:N/2-1),abs(Xshift))
xlabel('samples')
pause

%-----------------------------------------------------
% 8
% Let us plot the absolute value of X in figure 5
% as a sequence of samples from -N/2 to N/2-1, as a sequence
% along a normalized frequency axis from -1/2 to 1/2 and
% as a sequence of samples along a frequency axis from
% -1/(2T) to 1/(2T)

figure(5)
subplot(311)
stem((-N/2:N/2-1),abs(Xshift))
xlabel('samples')
pause
subplot(312)
stem((-N/2:N/2-1)/N,abs(Xshift))
xlabel('normalized frequency')
pause
subplot(313)
stem((-N/2:N/2-1)/N/T,abs(Xshift))
xlabel('frequency (Hz)')

%-----------------------------------------------------
% 9
% Now repeat the same analysis on a diferent signal x
% e.g. x=sinc((-N/2:N/2-1)/4); or x=ifftshift(gausswin(N,12));
% Fourier Transform Alias
% Author C. Prati
% February 17, 2020
clear
close all

%---------------------------------------------------
% 1
% Let us generate a discrete cosinusoidal signal x with
% frequency sweeping from f=0Hz to f=1000Hz in step of 50Hz
% N=100 samples using a sampling interval T=1/1000 second
% For each frequency, the signal x and its corresponding FFT X
% are plotted

N=100;
T=1/1000;
t=(0:N-1)*T;
tcontinuo=(0:10*N-1)*T/10;

for f=0:50:1000

  x=cos(2*pi*f*t);
  xcontinuo=cos(2*pi*f*tcontinuo);

  figure(1)
  subplot(211)
  stem(t,x)
  xlabel('time (seconds)')
  hold on
  plot(tcontinuo,xcontinuo,'r')
  hold off

  X=fft(x);
  Xshift=fftshift(X);

  subplot(212)
  stem((-N/2:N/2-1)/N/T,abs(Xshift),'filled')
  xlabel('frequency (Hz)')

  pause
end

  pause
%---------------------------------------------------
% 2
% Let us generate a "continuous" cosinusoidal signal x with
% frequency f1=2Hz spanning a time interval of To=0.99 second.
% The simulation of a continuous time signal is obtained by
% setting a very high sampling frequency compared with the
% signal frequency. Let us use fs=1000Hz.
fs=1000;
To=0.99;
t=(0:1/fs:To);
f1=2
x=cos(2*pi*f1*t);

%-----------------------------------------------------
% 3
% Let us plot the "continuous" signal x in figure 2

figure(2)
plot(t,x)
xlabel('time')
pause

%-----------------------------------------------------
% 4
% Let us sample the "continuous" signal x with T=0.05 s.
%(sampling frequency fs=1/T=20Hz)
% Let us plot the obtained samples superimposed to the
% continuous signal x on figure 2.

T=0.05;
nT=(0:T:To);
xn=cos(2*pi*f1*nT);
figure(2)
hold on
stem(nT,xn,'filled','r')
hold off
pause

%-----------------------------------------------------
% 5
% Let us compute the number of sample N of the signal xn
% Let us also check this number N by measuring the length of xn

N=floor(To/T)+1
Nmeasured=length(xn)
pause

%-----------------------------------------------------
% 6
% Let us compute the FFT X of the sampled signal xn and
% let us plot its absolute value in figure 3
```

```
Xn=fft(xn);
Xnshift=fftshift(Xn);

figure(3)
stem((-N/2:N/2-1)/N/T,abs(Xnshift))
xlabel('frequency (Hz)')

%--------------------------------------------------------
% 7
% Let us repeat steps from 2 to 4 by increasing the frequency of the
% "continuous" cosinusoidal signal x from f1=2Hz to f2=(2+1/T) Hz

f2=2+1/T
pause
x2=cos(2*pi*f2*t);
figure(4)
plot(t,x2)
xlabel('time')
pause

xn2=cos(2*pi*f2*nT);
figure(4)
hold on
stem(nT,xn2,'filled','r')
hold off
pause

%--------------------------------------------------------
% 8
% Let us compare the samples obtained by sampling this two signals
figure(4)
hold on
plot(t,x,'m')
```

FASTFT

```
tic
x1=rand(1,500000);
y1=rand(1,500000);
z1=conv(x1,y1);
toc

tic
x=rand(1,500000);
y=rand(1,500000);
X=fft(x);
Y=fft(y);
Z=X.*Y;
z=ifft(Z);
toc
```