الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

وزارة التعليم العالي و البحث العلمي
جامعة محمد بوضياف بالمسيلة
كلية التكنولوجيا
قسم الهندسة الكهربائية

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
MOHAMED BOUDIAF UNIVERSITY OF M'SILA
FACULTY OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING

جامعة محمد بوضياف - المسيلة

رقم:....68../ ق.ه.ك /2025 المسيلة في : 13 أفريل 2025

شهادة إدارية

بخصوص مطبوعة الدروس الخاصة بالأستاذ

حريزي عبد الغفور

بناءً على محضر اللجنة العلمية لقسم الهندسة الكهربائية تحت رقم: 41/ق.ه.ك/2025 المنعقد بتاريخ 17 فيفري 2025 والمتضمن تعين الخبراء: الأستاذ عابد احسن أستاذ محاضر-أ- بجامعة المسيلة الأستاذ واقني فيصل أستاذ محاضر –أ- بجامعة المسيلة، والأستاذ بورقيق قادة أستاذ محاضر –أ بجامعة تيارت وذلك لتقييم مطبوعة الأعمال التطبيقية الخاصة بالأستاذ حريزي عبد الغفور أستاذ محاضر "أ" بقسم الهندسة الكهربائية لجامعة المسيلة تحت عنوان :

" TP Traitement du signal "

مطبوعة أعمال تطبيقية مكتوبة باللغة الإنجليزية تحت عنوان:

" Practical work_Signal processing "

وبعد إطلاع رئيس اللجنة العلمية ورئيس القسم على التقارير الواردة و التي كانت كلها ايجابية، وعليه فإن اللجنة لا ترى مانعا أن تتخذه سندا في تدريس طلبة السنة الأولى ماستر في الألية ، شعبة الألية ، ميدان علوم و تكنولوجيا و أن تعتمد في أي تقييم للمسار العلمي للأستاذ المعني.

رئيس اللجنة العلمية
أ.د بوقرة عبد الرحمان

رئيس القسم
دفداف المبروك

ملاحظة: سلمت هذه الشهادة للمعني(ة) لاستعمالها في حدود ما يسمح به القانون.

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Mohamed BOUDIAF University - M'sila

1985
جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

Faculty of Technology
Department of Electrical Engineering

Practical Work : Signal Processing

Dr. HERIZI Abdelghafour

This course allows students to consolidate the knowledge acquired during the "Signal Processing" lectures through practical work to better understand and assimilate the content of this subject.

Material Polycopy - Specialty: 1st year Master's in Automation and Systems 2025.

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Mohamed BOUDIAF University - M'sila

Faculty of Technology
Department of Electrical Engineering

Practical Work : Signal Processing

Dr. HERIZI Abdelghafour

This course allows students to consolidate the knowledge acquired during the "Signal Processing" lectures through practical work to better understand and assimilate the content of this subject.

Material Polycopy - Specialty: 1st year Master's in Automation and Systems 2025.

Author

Dr. HERIZI Abdelghafour

abdelghafour.herizi@univ-msila.dz
abdelghafour_herizi @yahoo.fr

Faculty: Technology
Department: Electrical Engineering
Institution: Mohamed BOUDIAF University of M'sila – Algeria
Course Title: Signal Processing Practical Work
Semester: 1
Teaching Unit: Methodological Unit
Code: UEM 1.1, **Credits:** 2, **Coefficient:** 1
Semester Hour Volume: 22h30 (15 weeks)
Weekly Hour Volume: Practical Work: 1h30
Evaluation Method: Continuous Assessment: 100%

Course Description

The signal processing practical work (TP) aims to experiment with and analyze the different processing methods applied to analog and digital signals. The objectives of these practical sessions are: to understand the fundamental concepts of signal processing, to analyze a signal in the time and frequency domains, to apply filtering and transformation techniques, to use tools like MATLAB, Python or a digital oscilloscope, etc.

This document is a pedagogical support for the *Signal Processing Practical Work* intended for first-year Master's students in Automation and Systems in the Department of Electrical Engineering (Faculty of Technology). This practical work compendium is organized into five practical sessions:

The first practical session presents how to generate and visualize some signals, perform convolution between two analog signals, and how to determine the correlation between two continuous signals using MATLAB.

The objective of the second practical session is to apply the acquired knowledge on filtering. We will see, using Matlab, the different ideal filters applied in various fields such as electronics, automation, etc.

The third practical session is to address the basic concepts of the discrete Fourier transform. Then, perform the spectral analysis of non-periodic signals.

In the fourth practical session, we will present the concept of an infinite impulse response (IIR) linear process, alternating simulation programs of the impulse response by two methods: using the "**filter**" and "**impz**" commands, as well as the system response to a noisy input and the representation of poles and zeros.

We will complete this document with a practical session that focuses on the concept of a finite impulse response (FIR) linear process.

Evaluation Method

The student's evaluation is done by:
Continuous assessment: 100%

COURSE CONTENT

People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mohamed BOUDIAF University - M'sila

Faculty of Technology

Department of Electrical Engineering

***MODULE**: Signal Processing*

***ACADEMIC YEAR:** 1st Year Master Automation and Systems*

***TEACHER:** Dr. Abdelghafour HERIZI*

Practical Work 1: Signal Representation and Applications of the Fourier Transform using Matlab

I. Objective of the Practical Work:

- Generate and visualize some signals.
- Perform the convolution between two analog signals.
- Determine the correlation between two continuous signals using MATLAB.

II. Reminder:

A signal is the physical representation of information. The mathematical description of signals is the objective of signal theory. It offers the means to analyze, design, and characterize information processing systems.

II.1 Time-domain representation of signals:

This representation is based on the evolution of the signal as a function of time. There are two fundamental types of signals:

II.1.1 Certain or deterministic signals:

Their evolution as a function of time can be perfectly described by a mathematical model. Among deterministic signals, we distinguish:

1. Periodic signals: these are signals whose evolution in time is predictable and which obey a law of regular cyclical repetition, with a period T.
2. Random signals: These are signals whose temporal behavior is unpredictable, governed by the laws of chance.

II.2 Convolution Product:

In signal processing, convolution is the tool that allows the calculation of the output of a system. Indeed, for an input signal $e(t)$ subjected to a system with transfer function $h(t)$, the output will be the convolution of the two functions $h(t) * e(t)$.

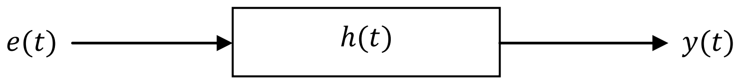


Figure 1: Convolution product.

$$y(t) = \int_{-\infty}^{+\infty} e(\tau)h(t-\tau)d\tau = \int_{-\infty}^{+\infty} e(t-\tau)h(\tau)d\tau$$

II.3 Correlation Functions:

Correlation functions characterize the degree of dependence between random variables.

II.3.1 Autocorrelation function:

We define the autocorrelation function by:

$$R_{xx}(T) = \int x(t)x(t+T)dt$$

If $T = 0$:$R_{xx}(0) = \int |x(t)|^2 dt$

If $x(t)$ is real: R is also real and even. We can show that the function R is maximum at the origin.

II.3.2 Cross-correlation function:

For two continuous signals $x(t)$ and $y(t)$, the correlation product is described by:

$$R_{xy}(\tau) = \int_{-\infty}^{+\infty} x(t)y(t+\tau)dt$$

II.4 Fourier Transform:

The Fourier transform, generalized by the use of distributions, allows us to obtain a spectral representation of deterministic signals. This expresses the frequency distribution of the amplitude, phase, energy or power of the signals considered.

Let $x(t)$ be a deterministic signal, its Fourier Transform is a generally complex function, of the real variable f defined by:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt$$

III. Manipulations :

III.1 Manipulation 1 :

A- Plotting Dirac:

```
clear all; close all; clc;

fs=250;
pas = 1/fs;
t=-5:pas:5;
A=not(t);
plot(t,A,'linewidth',3);
gridon;
title('La fonction A(t)=Dirac');
xlabel('temps(ms)');
ylabel('amplitude');
```

B- Plotting the step function:

```
clear all; close all; clc;

fs=250;
pas = 1/fs;
t=-5:pas:5;
B=1/2*(sign(t)+1);
plot(t,B,'linewidth',3);
gridon;
title('La fonction B(t)=echelon');
xlabel('temps(ms)');
ylabel('amplitude');
```

C- Type the following Matlab code:

```
clear all; close all; clc;
pas = pi/10;
t=0:pas:2*pi;
C=sin(t);
plot(t,C,'linewidth',2); grid on;
title('La fonction c(t)');
xlabel('temps');
ylabel('amplitude');
```

C-1: What does this program do?

C-2: Extend the time interval 4 times, what do you notice?

III.2 Manipulation 2 :

Let the function $y(t)$ be defined as follows:

$$y(t) = \begin{cases} sin(t) & si\ t \leq 0 \\ sin(4 * t) & si\ t > 0 \end{cases}$$

1. Write a program (in a file named Manip02.m) that plots the signal $y(t)$ in the interval [-4*pi , +4*pi] with a step equal to (pi/100).
2. Write on the plot of y(t) the labels "x" and "y" and the title as follows:

" $y = sin(t)$ if $x <= 0$ and $sin(4t)$ if $x > 0$ "

III.3 Manipulation 3 :

Provide the Matlab program that calculates the convolution product (Command: **conv**) of two rectangular signals, one with a duration of 20s and an amplitude of 2v and the other with a duration of 40s and an amplitude of 3v.

III.4 Manipulation 4 :

1. Provide the Matlab program that calculates the Fourier transform of a centered rectangular signal with amplitude A=1v and width T=20s.
2. Fourier Transform of a cosine signal: Provide the program that allows you to construct and display a cosine signal x(t) with amplitude 1v and frequency f0. This vector x is composed of N points and represents r periods of the cosine. The time variable of the signal is between 0 and Tmax = r/f0. The program also allows you to calculate and display the Fourier transform of this signal using the definition of the Fourier transform and also using the fft (Fast Fourier Transform) command in Matlab (See the help for fft, abs, and fftshift).

III.5 Manipulation 5 :

Here is a damped sinusoidal signal described as follows:

$$\begin{cases} x(t) = exp(-0.2 * t) .* s(t) & pour \;\; t > 0 \\ s(t) = sin(0.35 * t) \end{cases}$$

if we take the plotting interval t=-10:1000

1- plot the signal $x(t)$ using the Matlab command stem, we limit the edges of the signal with this command after the stem function: axis([-10 30 -0.6 0.6]);

2- calculate the Fourier transform of this signal using the fft command, then plot its spectrum using the following commands: real, imag.
Note: to create the frequency range, we take
f=(0:N-1)*1000/N;
such that N=length(t);

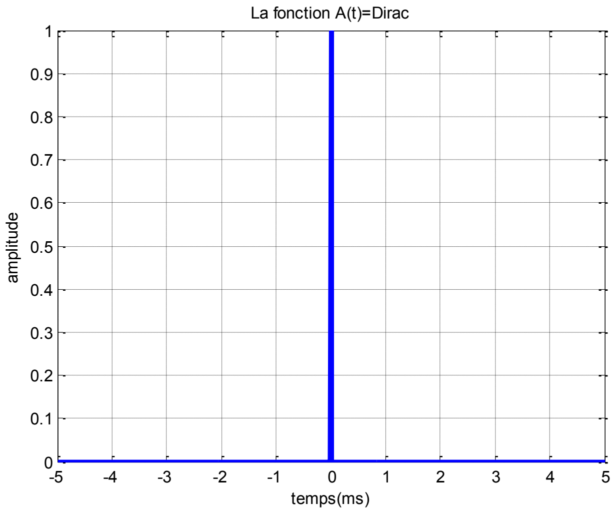
3- plot the modulus and phase of the $x(t)$ transform

4- Display all the figures in the same image using the subplot command. The subplot command allows you to arrange the figures.

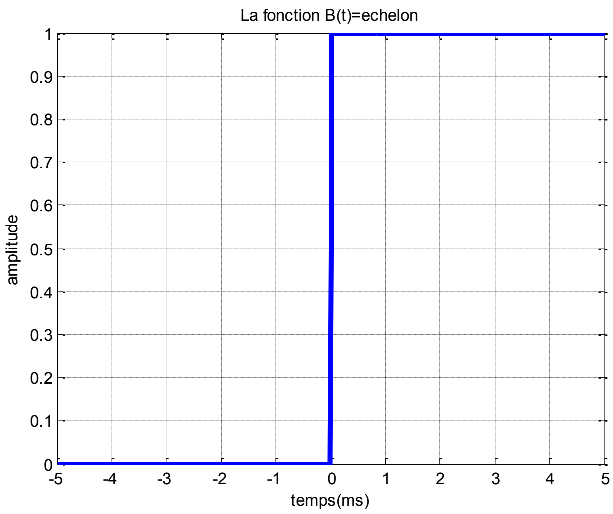
IV. Solution to the manipulations :

Manipulation 1 :

```
clear all; close all; clc;
fs=250;
pas = 1/fs;
t=-5:pas:5;
A=not(t);
plot(t,A,'linewidth',3);
% stem(t,A,'linewidth',3);
grid on;
title('La fonction A(t)=Dirac');
xlabel('temps(ms)');
ylabel('amplitude');
```

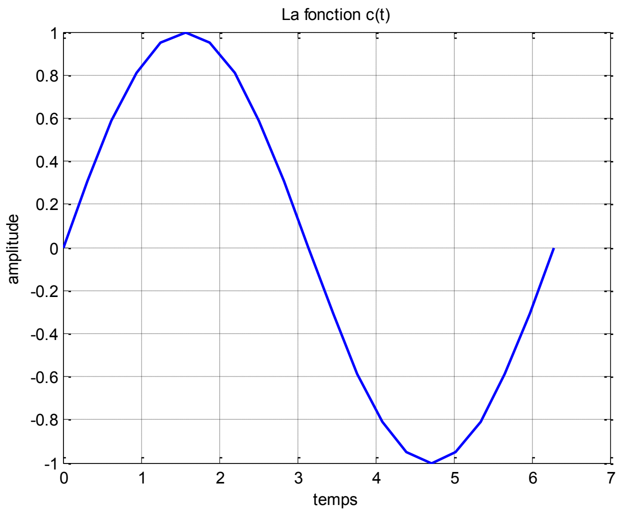



```
clear all; close all;
clc;
fs=250;
pas = 1/fs;
t=-5:pas:5;
B=0.5*(sign(t)+1);
plot(t,B,'linewidth',3);
% stem(t,B,'linewidth',3);
grid on;
title('La fonction B(t)=echelon');
xlabel('temps(ms)');
ylabel('amplitude');
```



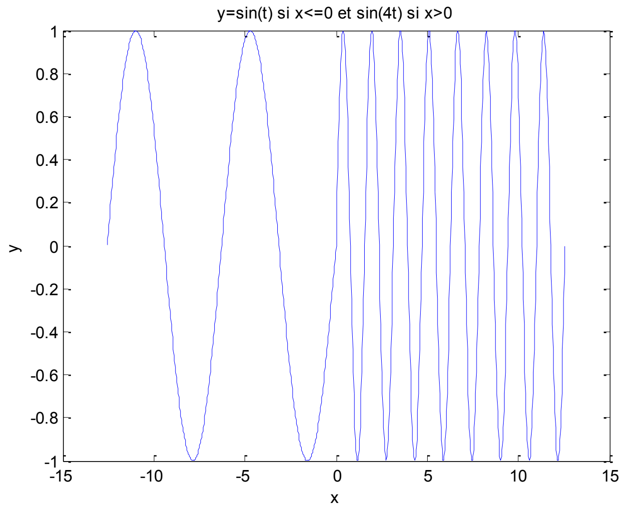

```
clear all; close all; clc;
pas = pi/10;
t=0:pas:2*pi;
```

```
C=sin(t);
plot(t,C,'linewidth',2);
% stem(t,C,'linewidth',2);
grid on;
title('La fonction c(t)');
xlabel('temps');
ylabel('amplitude');
```



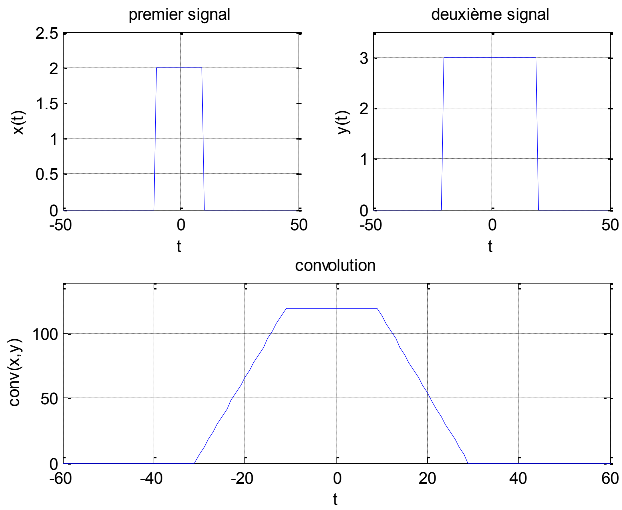
Manipulation 2 :

```
clear all; close all; clc;

% echo on pour afficher le texte suivant en commentaire
% dans la fenetre de commande lors de l'execution du programme
echo on;
%--------------------------------
%génération de sin( kx )
%y=sin x si x <= 0 et sin 4x autre
%-----------------------------------
pas=pi/100;
% déclarer y dans l'intervalle [-4*pi  0]
x1=-4*pi:pas:0;
y1 = sin(x1);
% déclarer y dans l'intervalle ]0  4*pi]
x2=pas:pas:4*pi;
y2 = sin(4*x2);
% Contenation
x=[x1 x2];
y=[y1 y2];
%
plot(x,y);
xlabel('x'); ylabel('y'); title('y=sin(t) si x<=0 et sin(4t) si x>0')

% yy = sin(x).*(x<=0) + sin(4*x).*(x>0);      %test logique
% clf;
```



Manipulation 3 :

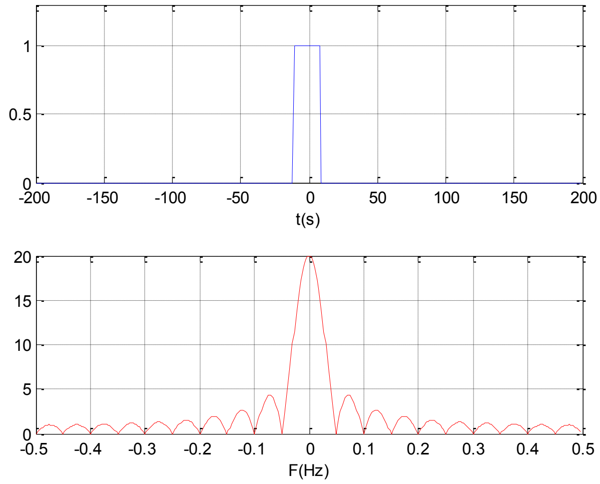
```
clear all; close all; clc;
x=zeros(100,1);
for i=41:60
    x(i)=2;
end
y=zeros(100,1);
for i=31:70
    y(i)=3; end
N=100;
a=-N/2:N/2-1;
b=-N:N-2;
subplot(2,2,1);plot(a,x);axis([-50,50,0,2.5]);grid;
title('premier signal');xlabel('t');ylabel('x(t)');
subplot(2,2,2);plot(a,y);axis([-50,50,0,3.5]);grid;
title('deuxième signal');xlabel('t');ylabel('y(t)');
subplot(2,1,2);plot(b,conv(x,y));grid;
axis([-60,60,0,140]);title('convolution');xlabel('t');ylabel('conv(x,y)');
```



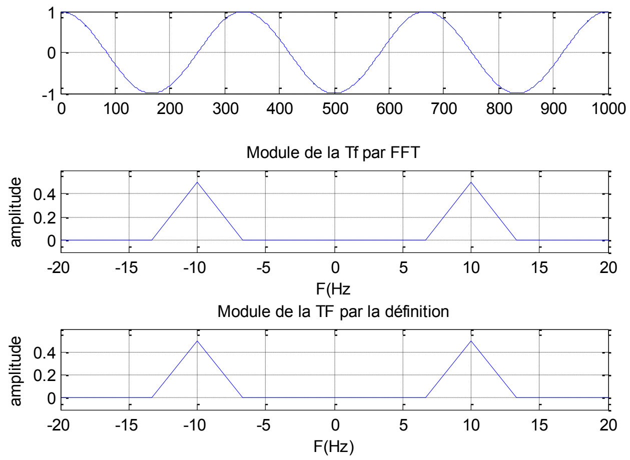
Manipulation 4 :

```
clear all; close all; clc;
N=400;               %nombre de points
x=zeros(N,1);
T=20;                %largeur du signal rect
for i=N/2-T/2:N/2+T/2-1
    x(i)=1;
end
t=-N/2:N/2-1
subplot(2,1,1);plot(t,x);axis([-N/2,N/2,0,1.3]);grid;
```

```
xlabel('t(s)');
f=-0.5:1/N:0.5-1/N;
g=fft(x,N);
subplot(2,1,2);plot(f,fftshift(abs(g(1:N))),'r');xlabel('F(Hz)');grid;
```

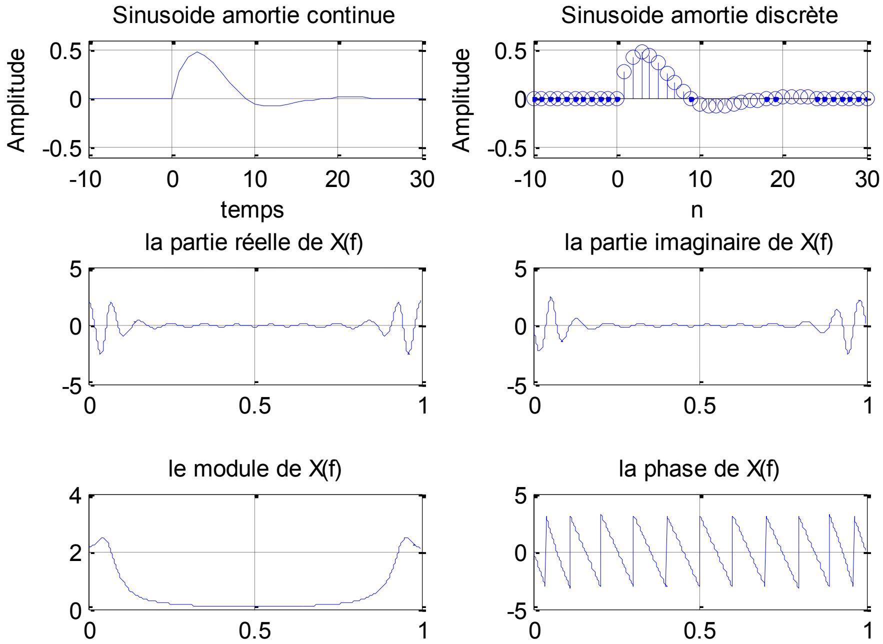



```
clear all; close all; clc;
f0=10;
N=1000;                 %nombre de points
r=3;
Tmax=r/f0;
t=0:Tmax/N:Tmax-Tmax/N;
x=cos(2*pi*f0*t);
subplot(3,1,1);plot(x);grid;
g=fft(x,N)/N;
subplot(3,1,2);
f=(-N/2:N/2-1)/r*f0;
plot(f,fftshift(abs(g)));
axis([-2*f0,2*f0,-0.1,0.6]);grid;
title('Module de la Tf par FFT');
xlabel('F(Hz');ylabel('amplitude');
i=1:N;
d=[];
for u=-0.5:1/N:0.5-1/N
    r=sum(x.*cos(2*pi*u*i))/N;
    im=sum(x.*sin(2*pi*u*i))/N;
    d=[d norm([r im])];
end
subplot(3,1,3);plot(f,d);axis([-2*f0,2*f0,-0.1,0.6]);grid;
title('Module de la TF par la définition');
xlabel('F(Hz)');ylabel('amplitude');
```



Manipulation 5 :

```
clear all; close all; clc;

t=-10:1000;
s=sin(0.35*t);
u=[t>0];
x=exp(-0.2*t).*s.*u;
% traçage continu
subplot(321);
plot(t,x); axis([-10 30 -0.6 0.6]); title('Sinusoide amortie continue');
xlabel('temps'); ylabel('Amplitude'); grid;
% traçage discrèt
subplot(322);
stem(t,x); axis([-10 30 -0.6 0.6]); title('Sinusoide amortie discrète');
xlabel('n'); ylabel('Amplitude'); grid;
%----------- T F
N=length(t);
f=(0:N-1)/N;
X=fft(x);

subplot(323); plot(f,real(X));   title('la partie réelle de X(f)'); grid
on;
subplot(324); plot(f,imag(X));   title('la partie imaginaire de X(f)');
grid on;
subplot(325); plot(f,abs(X));    title('le module de X(f)'); grid on;
subplot(326); plot(f,angle(X));  title('la phase de X(f)'); grid on;
```



People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mohamed BOUDIAF University - M'sila

Faculty of Technology
Department of Electrical Engineering

MODULE: *Signal Processing*

ACADEMIC YEAR: *1st Year Master Automation and Systems*

TEACHER: *Dr. Abdelghafour HERIZI*

Practical Work 2 : Analog Filtering

I. Objective of the Practical Work:

The objective of this practical work (TP) is to apply the acquired knowledge on filtering. Thanks to Matlab, we will see the different ideal filters applied in various fields such as electronics, automation, etc.

II. Reminder:

II.1 Low-pass Filter:

A low-pass filter attenuates frequencies above the chosen cutoff frequency f_c and lets low frequencies pass. Diagram of a passive low-pass filter:

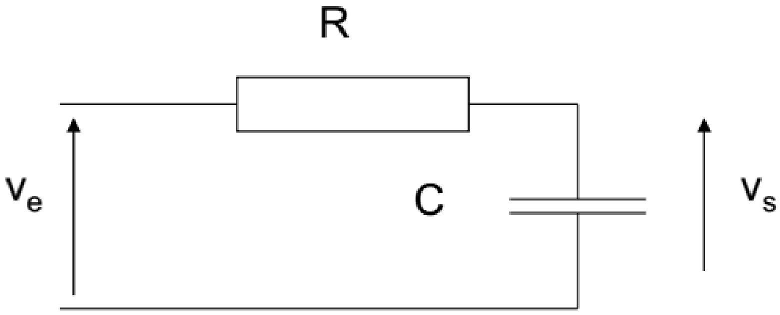


Figure 1: Low-pass filter.

We will determine its transfer function $H(j\omega)$

$$RC\frac{dV_s}{dt} + V_s = V_e \Rightarrow H(p) = \frac{1}{1 + RCp}$$

Therefore:

$$H(j\omega) = \frac{1}{1 + j\omega\tau} = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

With:$\tau = \frac{1}{\omega_c} = RC$

By setting $x = \frac{\omega}{\omega_c}$, therefore:$H(j\omega) = \frac{1}{1+jx}$

II.2 High-pass Filter:

A high-pass filter attenuates frequencies below the chosen cutoff frequency and preserves high frequencies. Diagram of a passive high-pass filter:

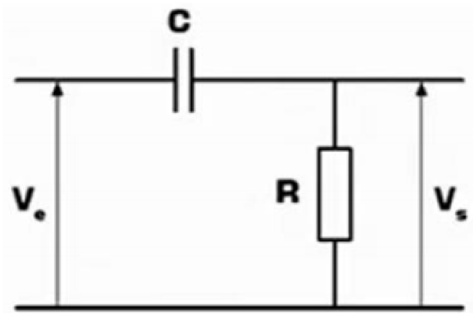


Figure 2: High-pass filter.

We will determine its transfer function$H(j\omega)$

$$H(j\omega) = \frac{R}{R + \frac{1}{jC\omega}} = \frac{jRC\omega}{1 + jRC\omega}$$

With: $\tau = RC$ and $\omega_c = \frac{1}{\tau}$

$$H(j\omega) = \frac{j\frac{\omega}{\omega_c}}{1 + j\frac{\omega}{\omega_c}}$$

By setting $x = \frac{\omega}{\omega_c}$

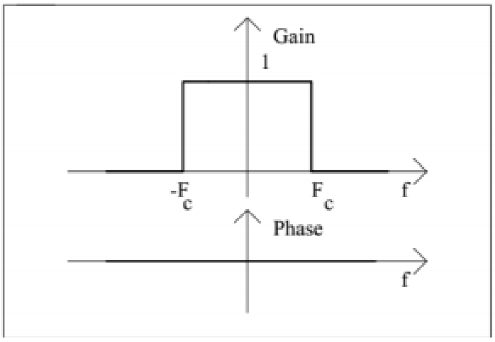
Therefore:

$$H(j\omega) = \frac{jx}{1 + jx}$$

II.3 Band-pass Filter:

In this part, we study the ideal filtering of a rectangular pulse signal.

$$\Pi_T(t) = \begin{cases} 1 & pour - T/2 < t < T/2 \\ 0 & ailleurs \end{cases}$$



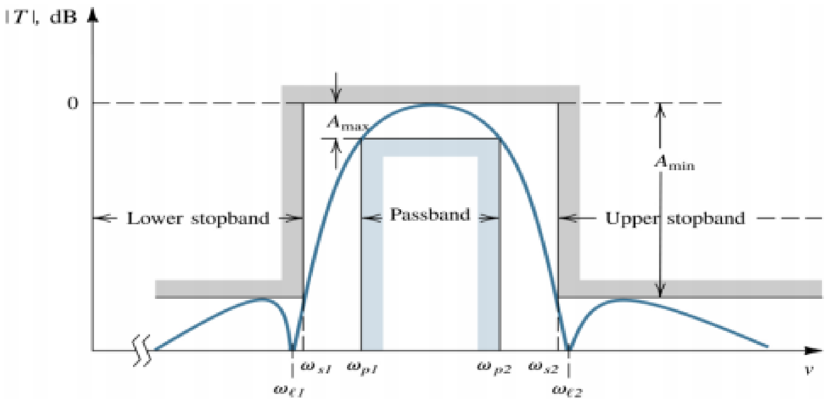
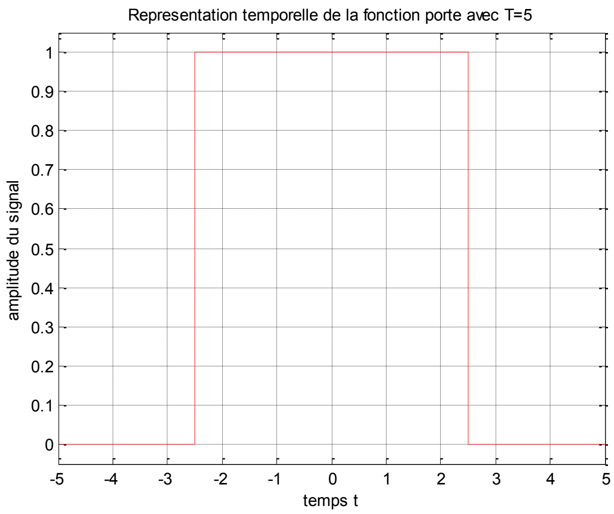


Figure 3: Ideal band-pass filter.

We consider an ideal band-pass filter with a cutoff frequency $H(f)$, $f_c = 2/T$

```
%Définitions préalables
T=5;
Fe=100;
Te=1/Fe;
fc=2/T;
%Definition du domaine temporel
t=-20:Te:20;
%Creation de la fonction porte
x= (((-T/2) <=t)&(t<=(T/2))) ;
%Representation temporelle de la fonction porte avec T=5
figure(1)
plot(t,x,'r');
grid;
axis([-5 5 -0.05 1.05]);
title('Representation temporelle de la fonction porte avec T=5')
xlabel('temps t')
ylabel('amplitude du signal')
```



In the frequency domain, it is also necessary to represent f as a vector. To be able to move from the time domain to the frequency domain without any problems, the dimension of f must be equal to the dimension of t.

We therefore have: $f = (F_e/2) + k * \Delta(f)$

With:

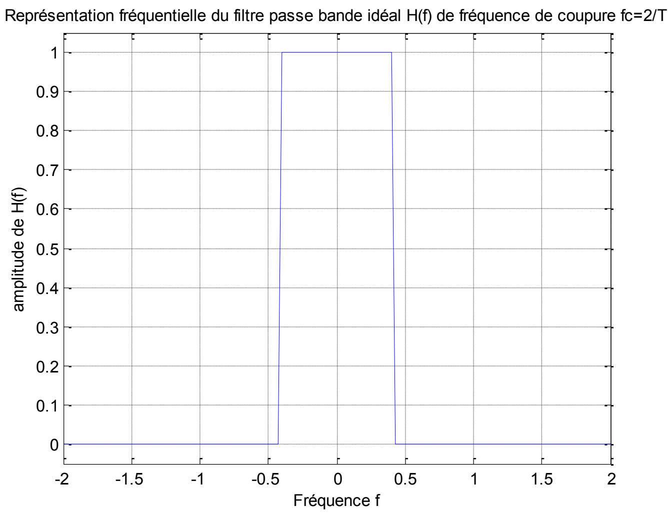
$\Delta(f) = ((F_e/2 - (-F_e/2))/4000 = F_e/4000 = 1/40HZ$

With the frequency domain defined, we can now plot $H(f)$. $H(f)$ is an ideal band-pass filter with a cutoff frequency$f_c = 2/T = 2/5 = 0.4HZ$

We therefore have:

$$H(f) = \begin{cases} 1 & pour - T/2 < f < T/2 \\ 0 & ailleurs \end{cases}$$

```
%Définition du domaine fréquentiel
f=linspace(-Fe/2,Fe/2,length(t));
%Création du filtre passe bande de fréquence de coupure fc=2/T
Hf=(((-2/T)<=f)&(f<=(2/T)));
%Représentation fréquentielle du filtre passe bande idéal H(f)
figure(2)
plot(f,Hf)
grid
axis([-2 2 -0.05 1.05])
title('Représentation fréquentielle du filtre passe bande idéal H(f) de fréquence de coupure fc=2/T')
xlabel('Fréquence f')
ylabel('amplitude de H(f)')
```



III. Manipulations :

III.1 Manipulation 1 :

We consider $X(f)$ the Fourier transform of the rectangular pulse function $\Pi_5(t)$. For this, we use the fft command (fast Fourier transform).

1. Calculation and plot of $X(f)$ between -2 and 2 Hz
2. What is the Fourier transform equal to?
3. What do you conclude? .

III.2 Manipulation 2 :

y is the filter's output in response to the gate input $x(t)$. Let. According to Plancherel's theorem, we obtain: $y(t) = h(t) * x(t) \Rightarrow Y(f) = H(f).X(f)$

$$Y(f) = \begin{cases} X(f) & pour -T/2 < f < T/2 \\ 0 & ailleurs \end{cases}$$

1. Calculation and plot of $Y(f)$ between -2 and 2 Hz.
2. What do you conclude? .
3. Calculation and plot of $y(t)$.
4. What do you conclude? .
5. Plot of y(t) for increasing values of the cutoff frequency $f_c = [2/T, 1,5,15,50]$.
6. What do you conclude? .

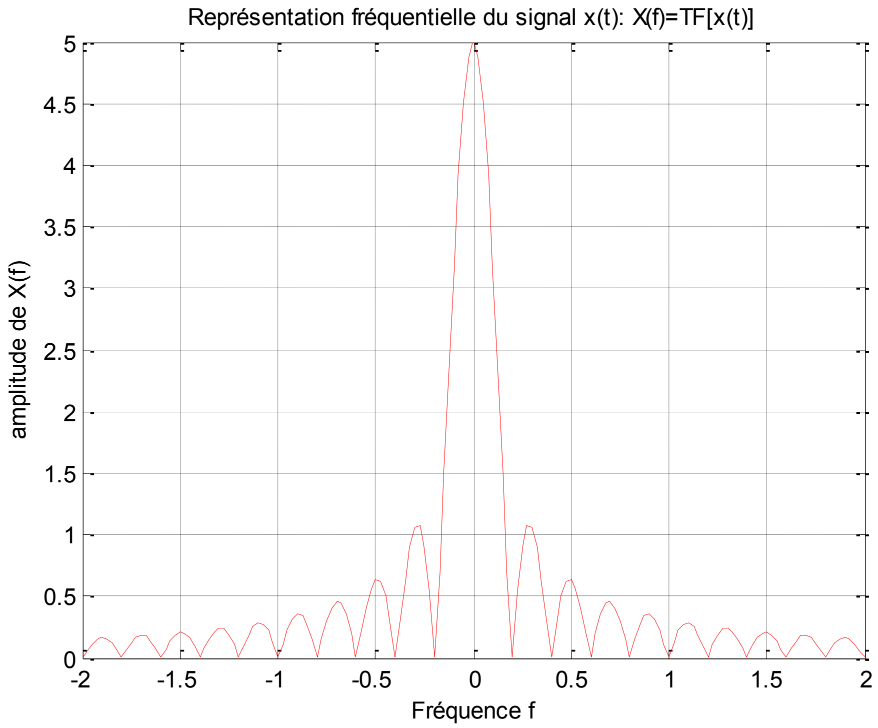
III.3 Manipulation 3 :

Provide the Matlab program that gives the function of the low-pass filter and high-pass filter.

IV. Solution to the manipulations :

Manipulation 1 :

```
%Définitions préalables
T=5;
Fe=100;
Te=1/Fe;
fc=2/T;
%Definition du domaine temporel
t=-20:Te:20;
%Creation de la fonction porte
x= (((-T/2) <=t)&(t<=(T/2))) ;
%Définition du domaine fréquentiel
f=linspace(-Fe/2,Fe/2,length(t));
%Transformée de Fourier de la fonction porte (appelée x):X(f)=TF[x(t)]
Xf=fftshift(fft(x)*Te);
%Représentation graphique de X(f)
figure(3)
plot(f,abs(Xf),'r')
grid
axis([-2 2 0 5])
title('Représentation fréquentielle du signal x(t): X(f)=TF[x(t)]')
xlabel('Fréquence f')
ylabel('amplitude de X(f)')
```



Note:

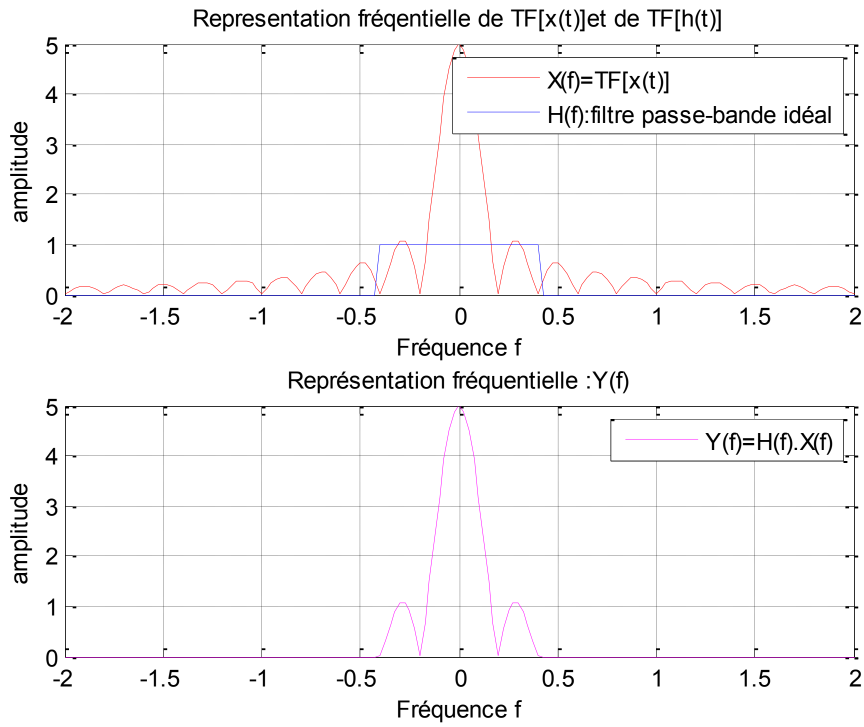
The Fourier transform of the rectangular pulse function is a sinc function, which is characterized by a main lobe and secondary lobes that gradually decrease in amplitude.

$TF[\Pi 5(t)]=5[\sin(5\pi f)]/5\pi f=5\text{sinc}(5f)$

Manipulation 2 :

```
%Définitions préalables
T=5;
Fe=100;
Te=1/Fe;
fc=2/T;
%Definition du domaine temporel
t=-20:Te:20;
```

```
%Creation de la fonction porte
x= (((-T/2) <=t)&(t<=(T/2))) ;
%Définition du domaine fréquentiel
f=linspace(-Fe/2,Fe/2,length(t));
%Transformée de Fourier de la fonction porte (appelée x):X(f)=TF[x(t)]
Xf=fftshift(fft(x)*Te);

%Création du filtre passe bande de fréquence de coupure fc=2/T
Hf=(((-2/T)<=f)&(f<=(2/T)));

%Calcul de Y(f)=H(f)X(f)
Yf=Hf.*Xf;
%Représentation fréquentielle
figure(4)
%A) Représentation graphique de H(f) et X(f)
subplot(2,1,1);plot(f,abs(Xf),'r',f,Hf)
grid
axis([-2 2 0 5])%LIMITE LA REPRESENTATION DE X(f)ET H(f)
title('Representation fréqentielle de TF[x(t)]et de TF[h(t)]')
xlabel('Fréquence f')
ylabel('amplitude')
legend('X(f)=TF[x(t)]','H(f):filtre passe-bande idéal','de fréquence de
coupure Fc=2/T')
%B) Représentation graphique de Y(f)=H(f)X(f)
subplot(2,1,2);plot(f,abs(Yf),'m')
grid
axis([-2 2 0 5]) % limite la représentation de Y(f) à l'intervalle de
fréquences [-2 2HZ]
title('Représentation fréquentielle :Y(f)')
xlabel('Fréquence f')
ylabel('amplitude')
legend ('Y(f)=H(f).X(f)')
```

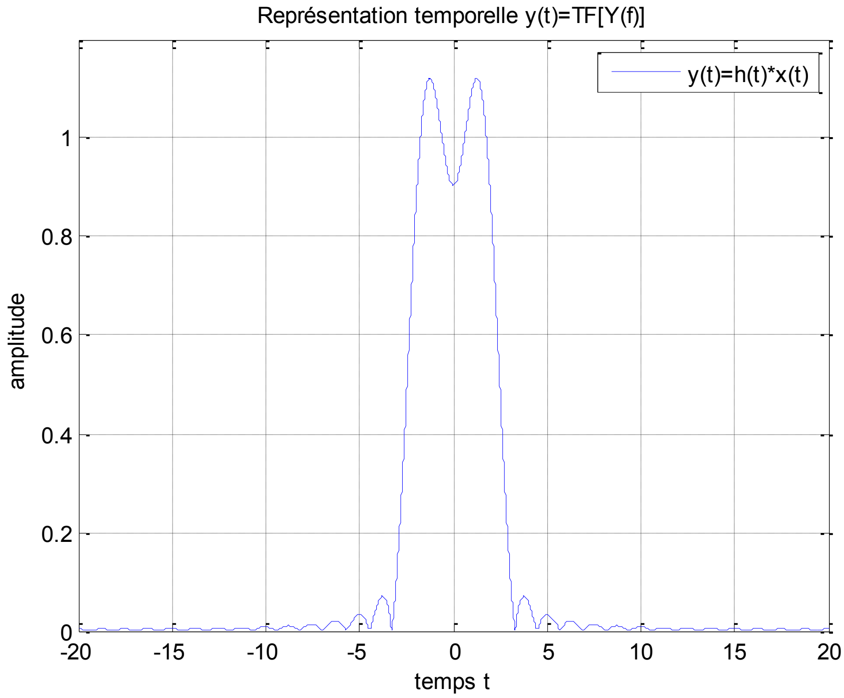


Note:

It is evident that the input signal X(f) is filtered by the ideal filter H(f) with a passband of [-0.5, 0.5], resulting in the output Y(f) containing only a main lobe and two side lobes, with frequencies ranging between -0.5 and 0.5.

The following formula was utilised in order to calculate and plot y(t):
In order to obtain y(t), it is necessary to return to the time domain. The Inverse Fourier Transform (IFT) is utilised to achieve this objective, whereby y(t) is expressed as $TF^{-1}[Y(f)]$. Within the MATLAB environment, the ifft command is utilised.

```
%Calcul de y(t)=TF-1[Y(f)]
yt=abs(ifft(fftshift(Yf)/Te));
%Représentation graphique y(t)=h(t)*x(t)
figure(5)
plot(t,yt)
grid
axis([-20 20 0 1.2])
title('Représentation temporelle y(t)=TF[Y(f)]')
xlabel('temps t')
ylabel('amplitude ')
legend('y(t)=h(t)*x(t)')
```

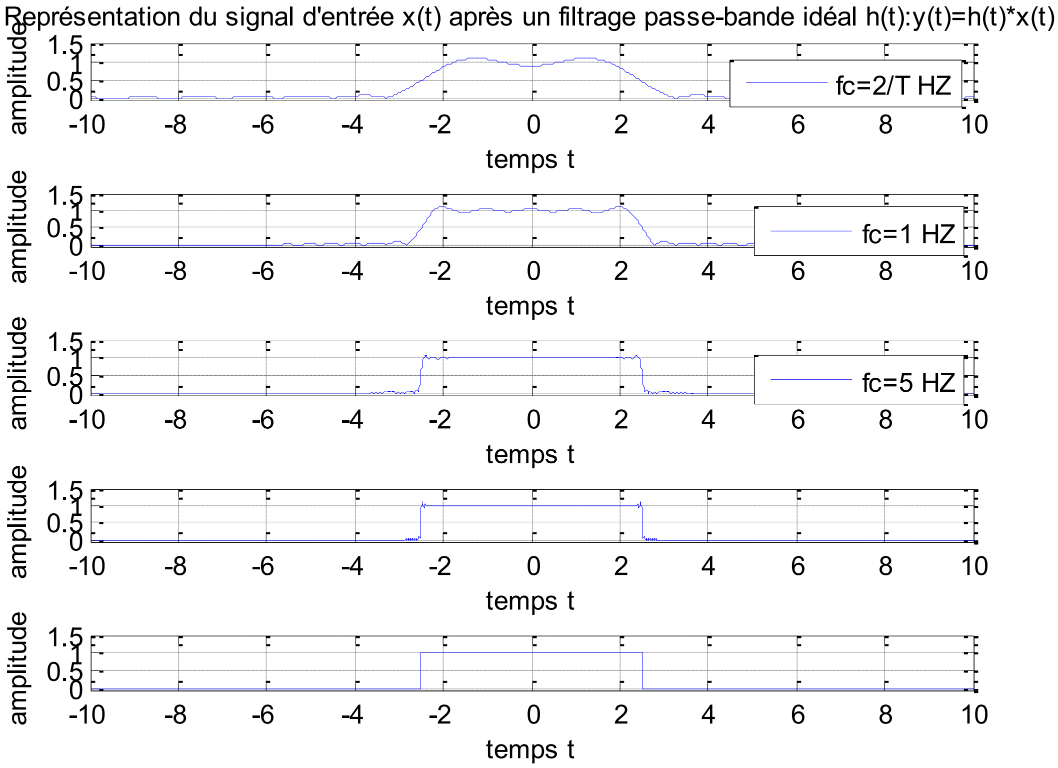


Note:
It is evident that the signal at the filter output is not congruent with the input signal; the waveform is distorted due to the filtration of frequencies that fall outside the [-0.5, 0.5] band.

Plot of y(t) for increasing values of the cutoff frequency

```
%Definition des différentes valeurs de fc pour lesquelles on étudiera y(t)
fc=[2/T,1,5,15,50];
figure(6)
for k=1:5
Hf=((-fc(k)<=f)&(f<=fc(k)));
Yf=Hf.*Xf;
yt=abs(ifft(fftshift(Yf)/Te));
subplot(5,1,k);plot(t,yt);
grid
axis([-10 10 -0.05 1.5])
if (k==1)
title('Représentation du signal d''entrée x(t) après un filtrage passe-
bande idéal h(t):y(t)=h(t)*x(t)')
end
xlabel('temps t')
ylabel('amplitude ')
switch k
```

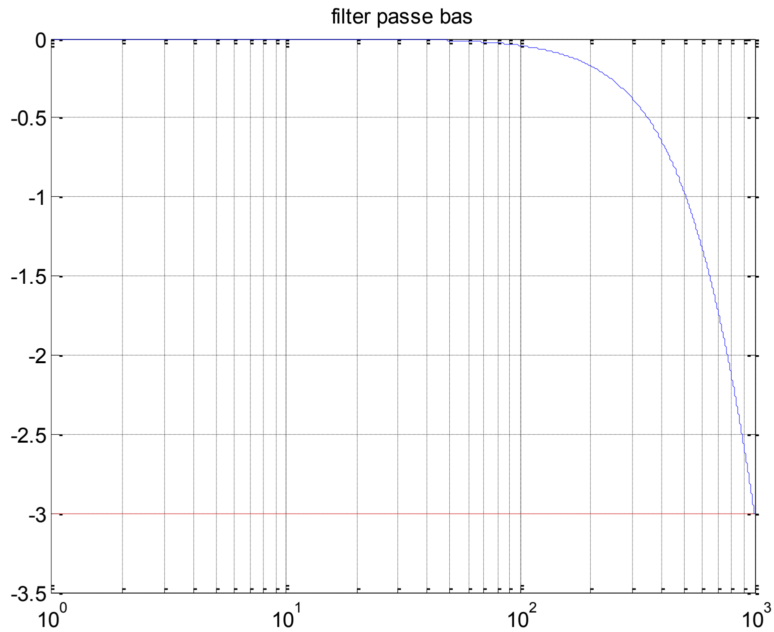
```
case 1,legend('fc=2/T HZ')
case 2,legend('fc=1 HZ')
case 3,legend('fc=5 HZ')
case 3,legend('fc=15 HZ')
case 3,legend('fc=50 HZ')
end
end
```



Note:
It is evident that as the cutoff frequency of an ideal filter is increased, the reconstructed signal at the filter's output assumes a configuration that is analogous to the signal prior to the application of the filter. It is imperative to emphasise the significance of selecting an appropriate cutoff frequency for a filter, given its pivotal role in attaining optimal filtering outcomes.

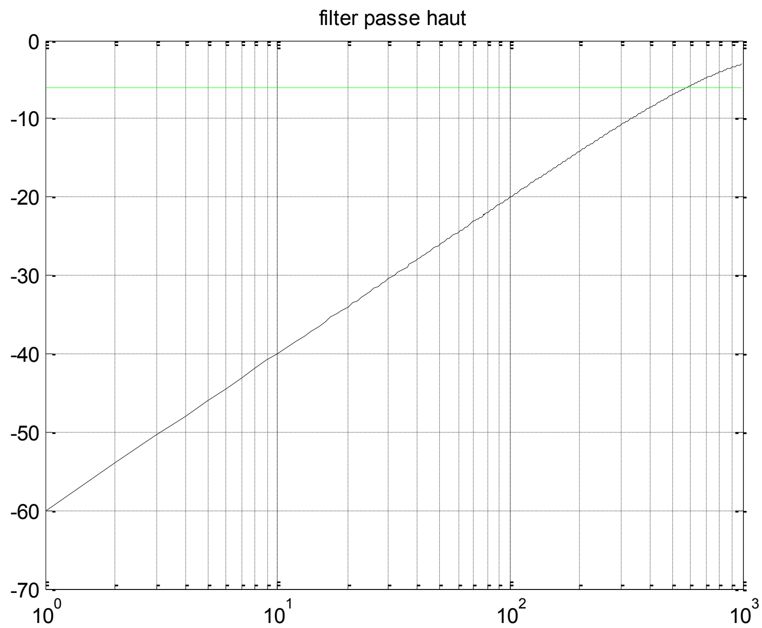
Manipulation 3 :
Our goal is to program the low-pass filter function in MATLAB.

```
clear all; close all; clc;
fe=1000 ;%fréquence d'échantillonnage
fc=62.5;%frquence de coupure
te=1/fe;%période d'échantillonnage
t=[0:te:1];
freq=(0:length(t)-1)*(fe/length(t));
w=2*pi*freq;
wc=2*pi*fe;
x=w/wc;
H=1./(1+i*x);
Module=20*log10(abs(H));
figure
semilogx(freq,Module,'')
hold on
semilogx(freq,max(Module)-3*ones(1,length(H)),'r--')
grid
title ('filter passe bas')
```



The objective of this study is to implement the high-pass filter function in the MATLAB programming language.

```
clear all; close all; clc;
fe=1000 ;%fréquence d'échantillonnage
fc=62.5;%frquence de coupure
te=1/fe;%période d'échantillonnage
t=[0:te:1];
freq=(0:length(t)-1)*(fe/length(t));
w=2*pi*freq;
wc=2*pi*fe;
x=w/wc;
H=(j*x)./(1+j*x);
Module=20*log10(abs(H));
figure
semilogx(freq,Module,'k')
hold on
semilogx(freq,max(Module)-3*ones(1,length(H)),'g')
grid
title ('filter passe haut')
```



People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mohamed BOUDIAF University - M'sila

جامعة محمد بوضياف - المسيلة

Université Mohamed Boudiaf - M'sila

Faculty of Technology

Department of Electrical Engineering

MODULE: *Signal Processing*

ACADEMIC YEAR: *1st Year Master Automation and Systems*

TEACHER: *Dr. Abdelghafour HERIZI*

Practical Work 3 : Discrete Fourier Transform

I. Objective of the Practical Work:

The objective of this practical work is to address the basic concepts of the discrete Fourier transform. Then, perform the spectral analysis of non-periodic signals using the tools available in Matlab.

II. Reminder:

II.1 Sampling and Quantization:

Most signals (at least those of natural origin) are inherently analog. It is therefore logical that the machines used to create or modify these signals have long been exclusively analog themselves. The telephone is a good example. It is these analog signals that interest us in this course. However, since the 1970s, electronic systems have been available (CAN: analog-to-digital converter, or ADC: analog-digital converter) that allow analog signals to be sampled and quantized, thus transforming them into digital signals.

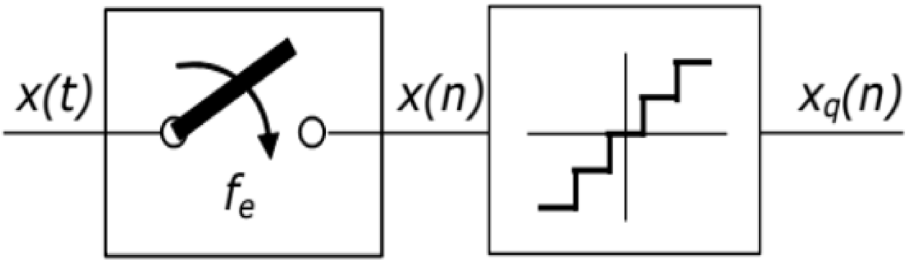
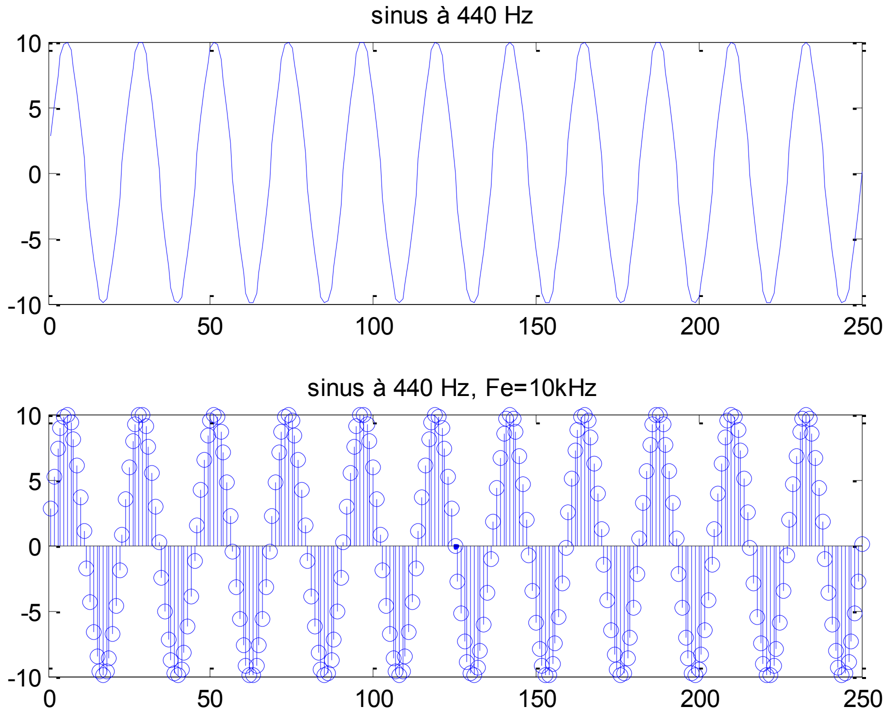


Figure 1: Schematic representation of the sampling and quantization of an analog signal.

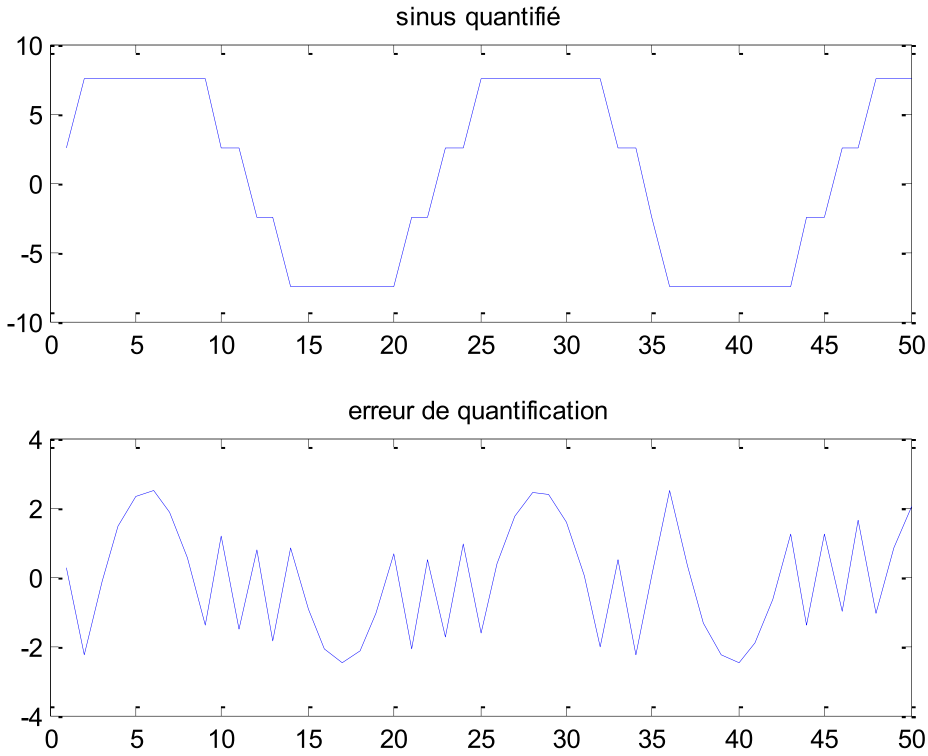
Example of sampling:

```
A=10; f=440; Fe=10000;
t=(1:20000)/Fe;
signal=A*sin(2+pi*f*t);
soundsc(signal,10000);
subplot(2,1,1);plot(signal(1:250));title('sinus à 440 Hz');
subplot(2,1,2);stem(signal(1:250));title('sinus à 440 Hz, Fe=10kHz');
```



Example of quantization:

```
A=10; f=440; Fe=10000;
t=(1:20000)/Fe;
signal=A*sin(2*pi*f*t);

b=2;
quantizer_max=max(signal);
q=(2*quantizer_max)/2^b;
signal_quantized=floor(signal/q)*q+q/2;
error=signal-signal_quantized;

soundsc(signal_quantized,10000);
plot(signal(1:50));
hold on;
subplot(2,1,1);plot(signal_quantized(1:50));title('sinus quantifié');
subplot(2,1,2);plot(error(1:50));title('erreur de quantification');
```



II.2 Discrete Fourier Transform:

The Discrete-Time Fourier Transform is conceptually simpler than the Fourier Transform of an analog signal, nevertheless, calculating the DTFT requires, in principle, an infinite computational load, since the series that defines it has an infinite number of terms, and the estimation must be made for all values of the normalized frequency F between 0 and 1.

This is the reason why the Discrete Fourier Transform or DFT was introduced. Its calculation is indeed limited to a finite number of values of n and for a finite number of values of F.

The discrete Fourier transform of a sequence of N terms $x(0), x(1), \dots, x(N-1)$ is called the sequence of N terms $X(0), X(1), \dots, X(N-1)$, defined by:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi\frac{nk}{N}}$$

In practice, the N terms $x(n)$ can be N samples of a sampled analog signal: $x_n = x(nT_e)$, and the N terms $X(k)$ correspond to an approximation of the Fourier transform of this signal at the N frequency points $f_k = kf_e/N$, with k between 0 and N-1, i.e., f between 0 and f_e.

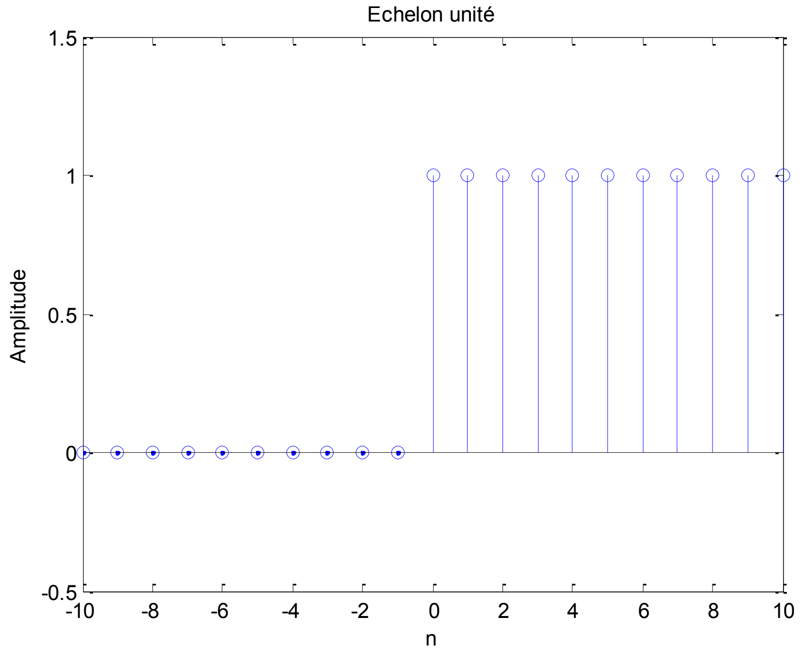
The inverse discrete Fourier transform is defined by:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi\frac{nk}{N}}$$

II.3 Digital Signals:

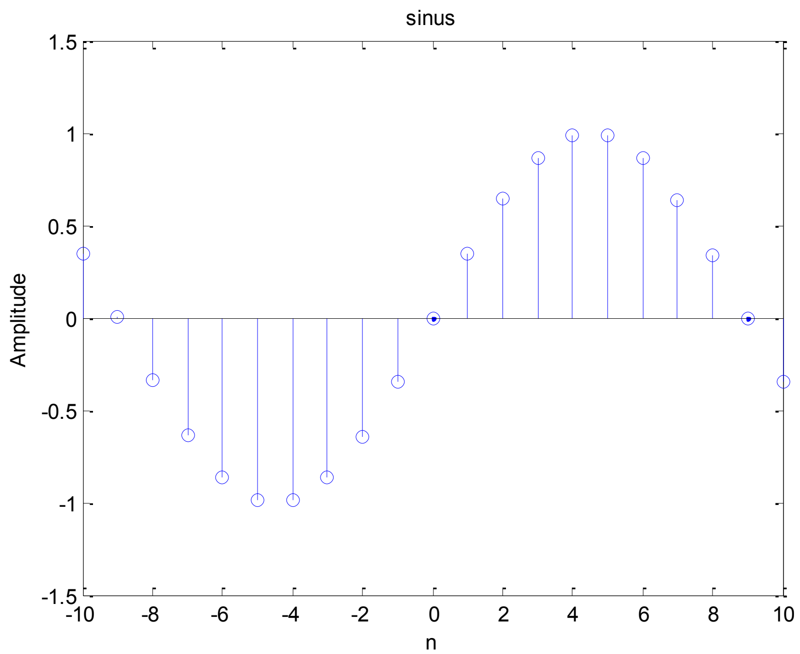
II.3.1 The unit step:

```
t=-10:10;
x=[zeros(1,10),ones(1,11)];
stem(t,x);
axis([-10 10 -0.5 1.5]);
title('Echelon unité');
xlabel('n');
ylabel('Amplitude');
```



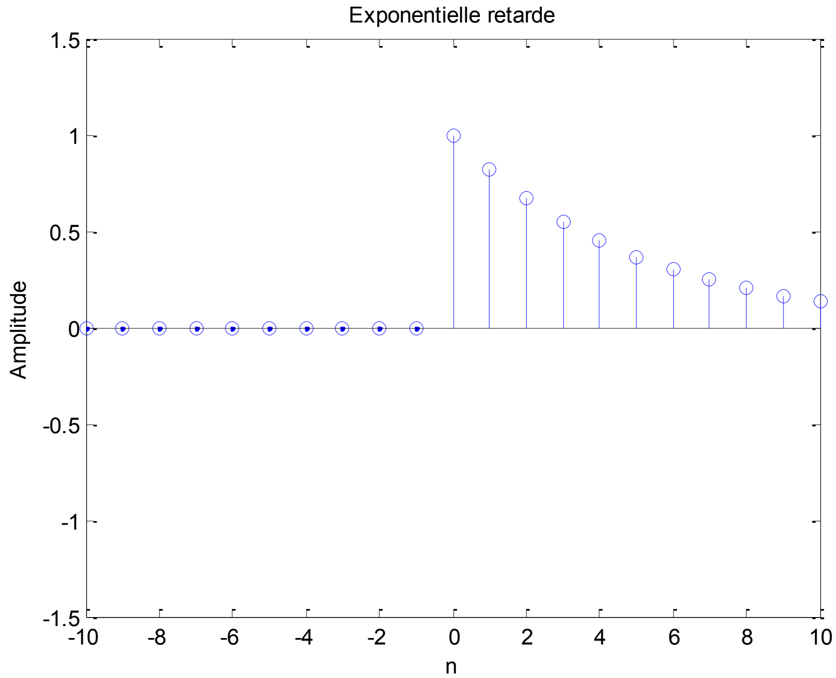
II.3.2 Sine Wave :

```
t=-10:10;
x=sin(0.35*t);
stem(t,x);
axis([-10 10 -1.5 1.5]);
title('sinus'); xlabel('n');
ylabel('Amplitude');
```



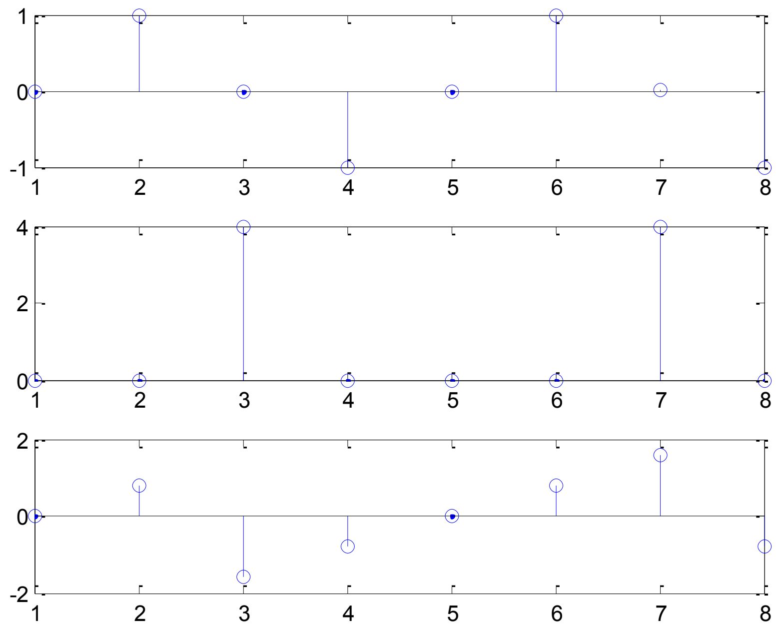
II.3.3 Exponential :

```
t=-10:10;
u=[zeros(1,10),ones(1,11)];
x=exp(-0.2*t).*u;
stem(t,x);
axis([-10 10 -1.5 1.5]);
title('Exponentielle retarde');
xlabel('n');
ylabel('Amplitude');
```



II.3.4 Example of DFT :

```
x=sin(2*pi/8*2*(0:7));
X=fft(x,8);
subplot(3,1,1);stem(x);
subplot(3,1,2);stem(abs(X));
subplot(3,1,3);stem(angle(X));
```



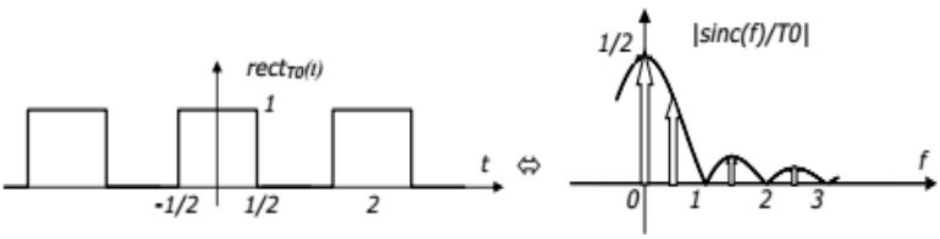
II.4 Fast Fourier Transform :

In 1965, Cooley and Tukey proposed a method that significantly reduces the computation time of the DFT of a sequence whose number of samples N is decomposable into factors (typically, a power of 2). Subsequently, many algorithms have been published; they are known under the general term Fast Fourier Transform (FFT). All these algorithms are based on the same principle, which consists of decomposing the calculation of the DFT into several DFTs of smaller length. The implementation of this principle leads to different methods whose performances are comparable. We will describe here more particularly the so-called radix-2 algorithm with time interleaving.

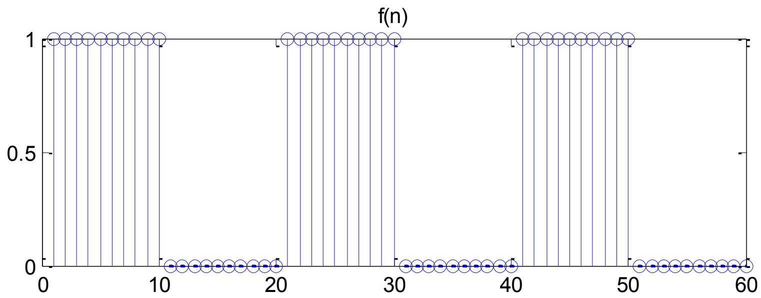
II.4.1 DFT of Periodic Signals:

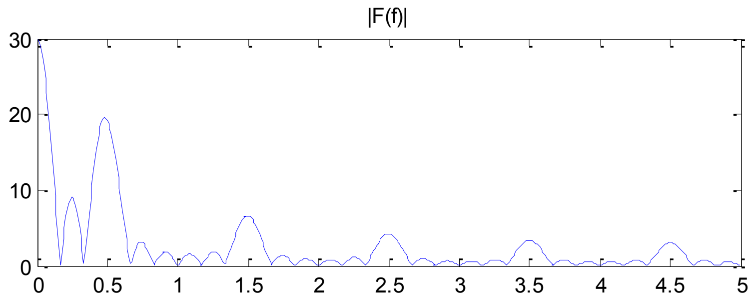
Let's make the rectangular signal rect(t) periodic with a period equal to 2s. Its Fourier transform will show lines at f=0, 1/2, 1, 3/2, etc., with a value of sinc(f)/2.

We observe that the transform shows a line at f=0 (DC component=1/2), then only odd harmonics of F*, with amplitudes decreasing as 1/k.




```
signal=[rectwin(10);zeros(10,1)];     %rectangle width : 1s =10samples
signal=[signal;signal;signal];
[fourier,frequencies]=freqz(signal); % computes the FT in f=[0,Fe/2]

subplot(2,1,1);stem(signal);title('f(n)')
Fe=10;
subplot(2,1,2);plot(frequencies/pi*(Fe/2),abs(fourier));title('|F(f)|');
```



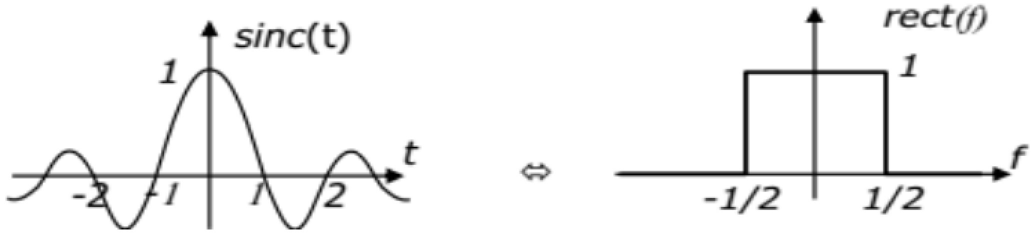


II.4.2 DFT of Non-Periodic Signals:

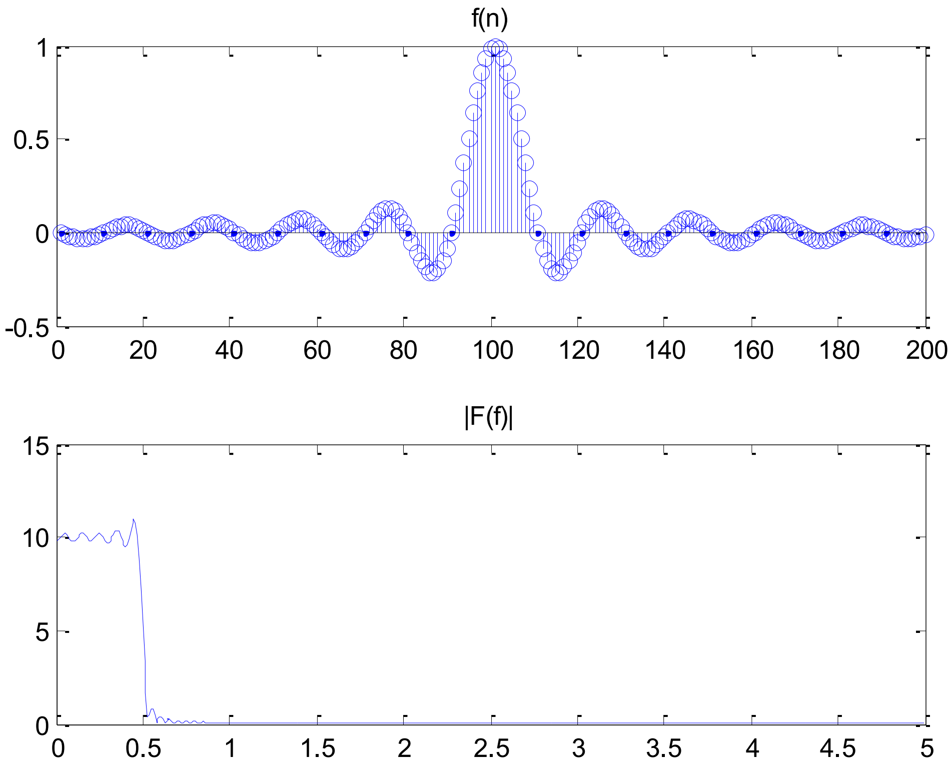
If the signal whose spectrum we are trying to observe is of finite duration, and the number N of samples available covers a time range greater than or equal to this duration, the effect of N is obviously transparent: the DFT then displays N values of the original DFT.

Transform of the cardinal sine: By applying the duality property of the Fourier transform, we get:

$$\text{sinc}(t) \overset{F}{\Leftrightarrow} = \text{rect}(-f) = rect(f)$$




```
Fe=10; t=(-100:99)*1/Fe;              %Fe=10; 200 samples ; 20 oscillations
signal=sinc(t);
[fourier,frequencies]=freqz(signal); % computes the FT in f=[0,Fe/2]

subplot(2,1,1);stem(signal);title('f(n)')
Fe=10;
subplot(2,1,2);plot(frequencies/pi*(Fe/2),abs(fourier));title('|F(f)|');
```



III. Manipulations :

III.1 Manipulation 1 :

We sample a sinusoid $f(t) = sin\ (2\pi f_0 t)$ at the sampling frequency of 10000Hz. Let's draw the shape of the samples (i.e., the shape of the corresponding function $f(t)$) for values of f_0 equal to: 1000Hz, 2500Hz, 5000Hz, 7500Hz, 9000Hz and 31000Hz. To ensure the graphs have identical time axes, let's choose to show the first 10 ms of the signals.

1. Provide the program?
2. What do you conclude? .

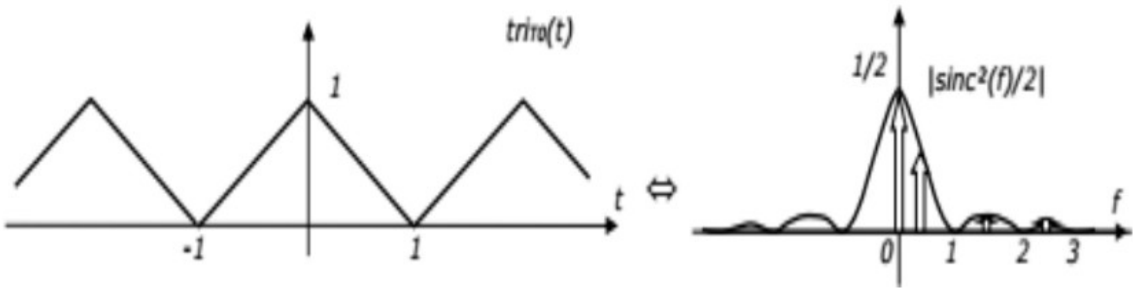
III.2 Manipulation 2 :

Consider the DFT of the imaginary part (the sine) of one of the basis vectors, for example w2 for N=8. Since Im(w2)= -j/2 (w2-w-2)= -j/2 (w2-wN-2), all the values X(k) of the DFT will be zero, except X(2) and X(6) which will be –4j and +4j, which corresponds well to a conjugate sequence modulo N.

Give the DFT of sin in Matlab?

III.3 Manipulation 3 :

Let's make the triangular signal tri(t) periodic with a period equal to 2s. Its Fourier transform will show lines at f=0, 1/2, 1, 3/2, etc., with a value of $sinc^2(f)/2$.



1. Provide the program that represents the Fourier transform of this signal?
2. What do you conclude? .

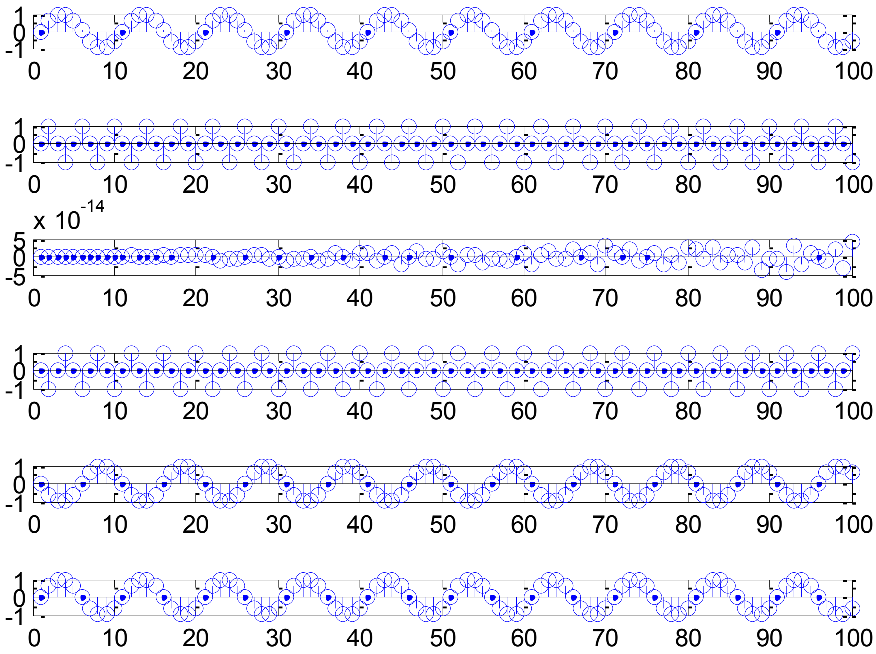
III.4 Manipulation 4 :

Let's calculate the DFT of a 16-point rectangular window, using FFT on 16, 8, and 4 points.

IV. Solution to the manipulations :

Manipulation 1 :

```
clear all; close all; clc;
subplot(6,1,1);stem(sin(2*pi*1000*[0:99]/10000));
subplot(6,1,2);stem(sin(2*pi*2500*[0:99]/10000));
subplot(6,1,3);stem(sin(2*pi*5000*[0:99]/10000));
subplot(6,1,4);stem(sin(2*pi*7500*[0:99]/10000));
subplot(6,1,5);stem(sin(2*pi*9000*[0:99]/10000));
subplot(6,1,6);stem(sin(2*pi*31000*[0:99]/10000));
```

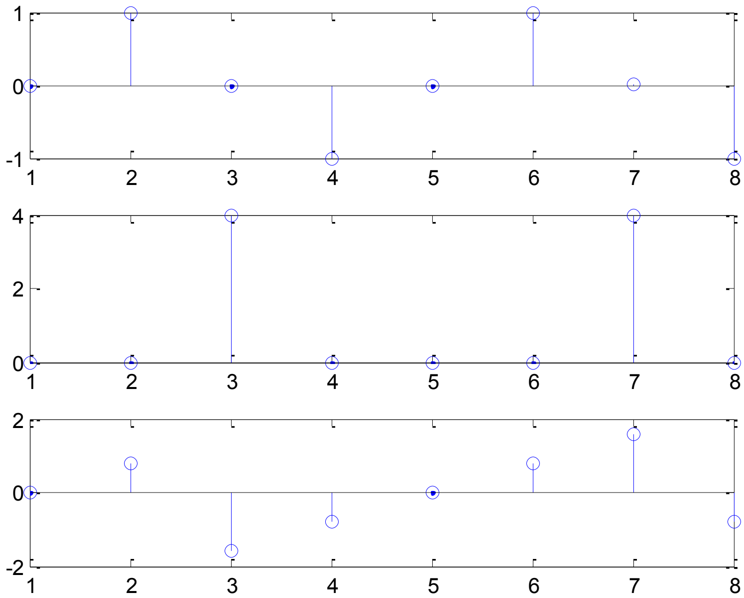


The following observations can be made:

1. It is evident that the sampled signals of the 1000 Hz and 2500 Hz sinusoids provide a realistic representation of the underlying signals. The frequency of the corresponding sinusoids can be determined by measuring their period on the graph.
2. When the 5000-Hz sinusoid is sampled, the resultant signal exhibits a negligible amplitude. It is a fundamental principle that the presence of any sample should be indicative of a zero value. The presence of non-zero values on the graph can be attributed to rounding errors in MATLAB.
3. The sampled signals of the 7500 Hz and 9000 Hz sinusoids are identical (with the exception of the sign) to those of the 2500 Hz and 1000 Hz sinusoids. Consequently, it is impossible to ascertain the precise frequency of the underlying sinusoids post-sampling. This phenomenon can be attributed to the spectral aliasing of the original sinusoids around the Nyquist frequency, which is defined as 5000 Hz.
4. The results obtained from the sampling of the 31,000-Hz sinusoid are consistent with those obtained from the sampling of the 1,000-Hz sinusoid. This phenomenon can be attributed to the superposition of all the shifted versions (by multiples of the sampling frequency) of the Fourier transform of the original signal.

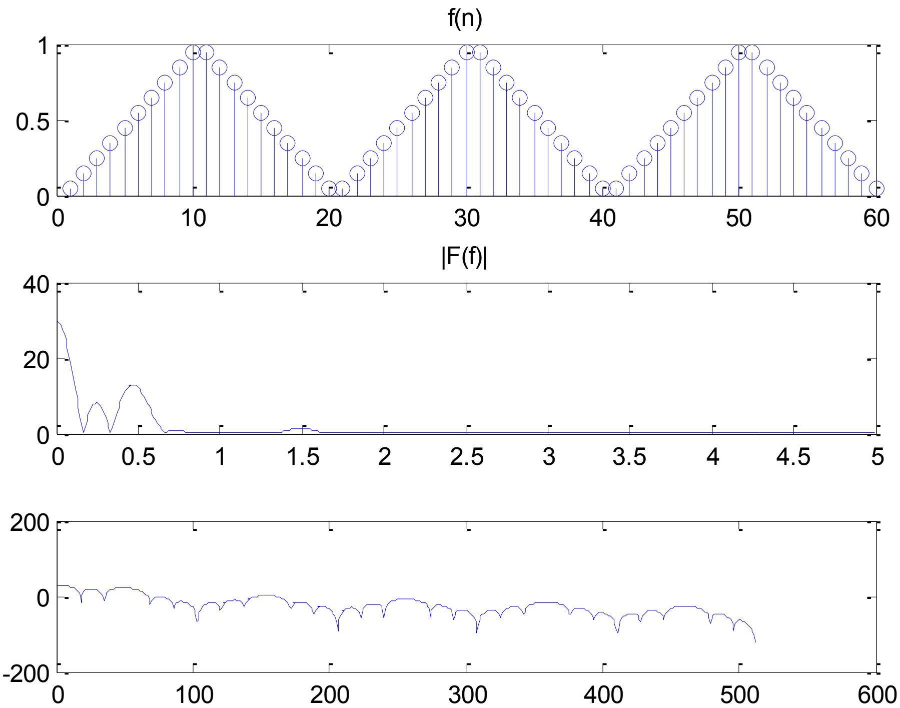
Manipulation 2 :

```
clear all; close all; clc;
x=sin(2*pi/8*2*(0:7));
X=fft(x,8)
subplot(3,1,1);stem(x);
subplot(3,1,2);stem(abs(X));
subplot(3,1,3);stem(angle(X));
```



Manipulation 3 :

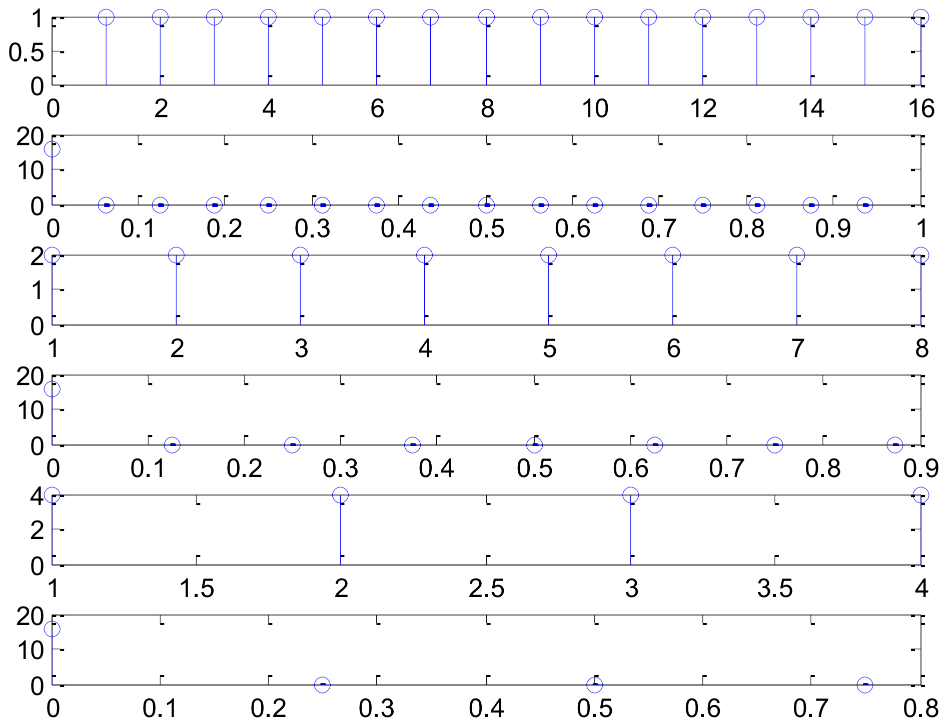
```
clear all; close all; clc;
signal=[triang(20)];    %triangle width:2s=20samples
signal=[signal;signal;signal]
[fourier,frequencies]=freqz(signal) %comptes the FT in f=[0,Fe/2]
subplot(3,1,1);stem(signal);title('f(n)')
Fe=10;
subplot(3,1,2);plot(frequencies/pi*(Fe/2),abs(fourier));title('|F(f)|');
Module=20*log10(abs(freqz(signal)));
subplot(3,1,3);plot(Module);
```



It is evident that the transformed signal exhibits a maximum at f=0 (DC component = 1/2), followed by only odd harmonics (f*) with amplitudes diminishing as 1/k. In consideration of the negligible amplitudes of these peaks, it is advantageous to display the magnitude of the Fourier transform in decibels (dB). This is the function of the freqz function by default.

Manipulation 4 :

```
clear all; close all; clc;
x=ones(1,16);
subplot(6,1,1);stem(x)
subplot(6,1,2);stem((0:15)/16,abs(fft(x,16)));
x=[ones(1,8)+ones(1,8)];
subplot(6,1,3);stem(x)
subplot(6,1,4);stem((0:7)/8,abs(fft(x,8)));
x=[ones(1,4)+ones(1,4)+ones(1,4)+ones(1,4)];
subplot(6,1,5);stem(x)
subplot(6,1,6);stem((0:3)/4,abs(fft(x,4)));
```



People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mohamed BOUDIAF University - M'sila

1985

جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

Faculty of Technology

Department of Electrical Engineering

***MODULE**: Signal Processing*

***ACADEMIC YEAR:** 1st Year Master Automation and Systems*

***TEACHER:** Dr. Abdelghafour HERIZI*

Practical Work 4: IIR Digital Filtering

I. Objective of the Practical Work:

This practical work is based on the concept of a linear process with infinite impulse response (IIR). We will alternate between simulation programs for the impulse response using two methods: the "**filter**" command and the "**impz**" command. , as well as the system's response to a noisy input and the representation of poles and zeros.

II. Reminder:

II.1 Transfer Function:

A transfer function is an input-output description of a linear time-invariant system. In this practical work, we will consider as a linear process: a digital filter whose transfer function is defined in the frequency domain (Z-Transform).

II.2 Definition of a digital filter:

A digital filter can be defined by a difference equation, i.e., the mathematical operation of the filter in the (discrete) time domain. The general form of an N order filter is as follows:

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k)$$

The general transfer function of a digital filter of order N is as follows:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{M} a_k z^{-k}}$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{a_0 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_M z^{-M}}$$

The value of the coefficients a and b will determine the type of filter, low-pass, high-pass, etc.

III. Infinite Impulse Response (IIR) Filter:

IIR, from "Infinite Impulse Response," have an impulse response that never stabilizes, even at infinity. IIR filters can be implemented using the following formula:

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k)$$

This type of filter is recursive, meaning that the filter output depends on both the input signal and the output signal, thus having a feedback loop. IIR filters are mainly the digital version of traditional analog filters: Butterworth, Chebyshev, Bessel, Elliptic.

III.1 Characteristics of Infinite Impulse Response Filters:

- More flexible than FIR filters: The IIR filter with infinite impulse response corresponds to a looped structure where the sample $y(n)$ depends not only on the $x(n)$ but also on the $y(n-p)$: there is therefore a reinjection of the output to the input. Looping implies a "risk of instability" (feedback theory) with a possibly long oscillatory impulse response to stabilize, hence the name IIR.
- Their transfer function is the ratio of two polynomials in z.
- Require less computation than FIR filters.
- Their stability is not guaranteed.
- Pole positions need to be checked.

IV. Manipulations :

IV.1 Manipulation 1 :

Consider a linear process with an infinite impulse response (IIR filter), described by the following digital transfer function:

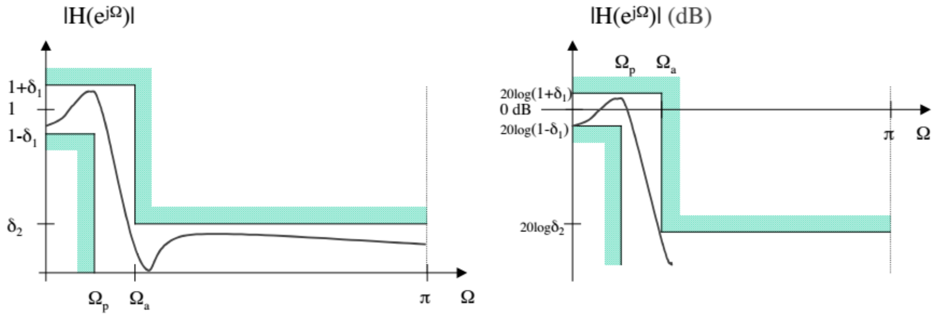
$$H(z) = \frac{0.2 + 0.5z^{-1}}{1 - 0.2z^{-1} + 0.8z^{-2}}$$

1. Represent the poles and zeros in the Z-plane.
2. Plot the impulse response of the IIR filter over the first 20 samples using the "impz" command.

3. Plot the impulse response of the IIR filter over the first 20 samples using the "filter" command.
4. Represent the output of the system with white noise as input.

IV.2 Manipulation 2 :

The digital filter is specified by the template in dB as shown in the figure below:



a) Linear frequency response b) Frequency response in dB

With:

$Fp = 1kHz, Fs = 3kHz, Fe = 10kHz$

$\Delta_1 = 20\log(1 + \delta_1) = 1dB$

$\Delta_2 = 20\log(\delta_2) = -20dB$

The synthesis by the bilinear method of the Chebyshev filter $H(z)$ leads to the function:

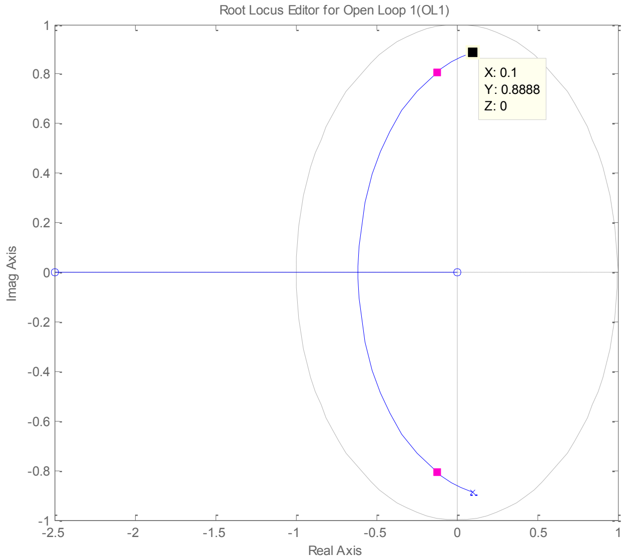
$$H(z) = \frac{0.079(z+1)^2}{z^2 - 1.2z + 0.516}$$

1. Write the Matlab program to visualize the magnitude and phase frequency response of the digital filter.
2. Relate the abscissa scale of the Matlab plot command to the frequencies.
3. What, then, are the relationships between sampling frequency and cutoff frequency? . Does the filter meet the specified template?
4. Visualize the impulse response.

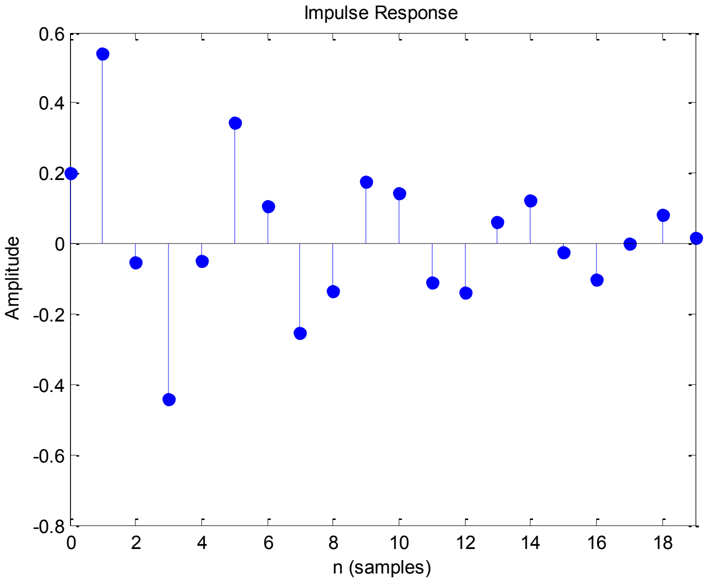
V. Solution to the manipulations:

Manipulation 1 :

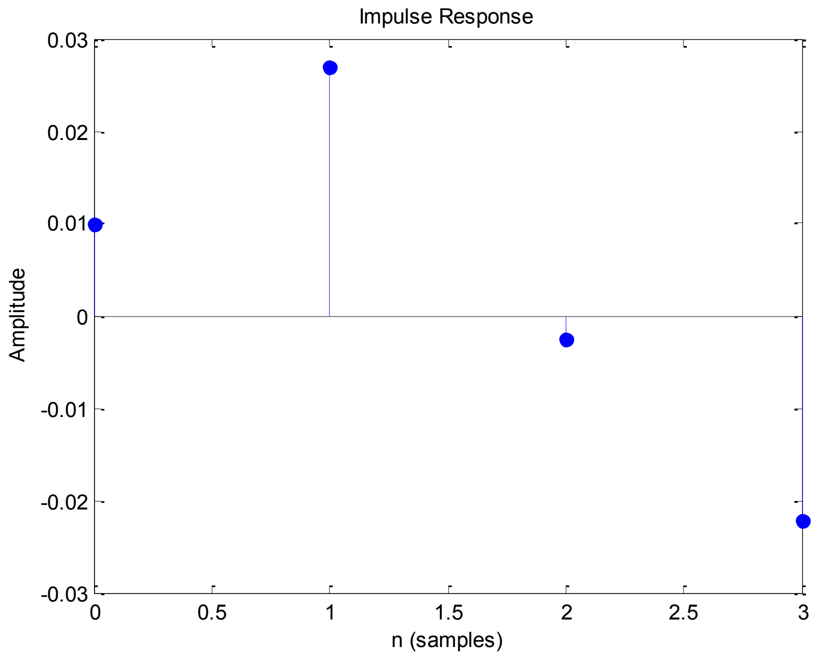
```
close all,clear all;        %Ferme les figures en cours
N=1024;                     %nombes de points
imp=[1 0 0 0];              %l'entrée impulsion de Dirac
a=[0.2 0.5 0];              %initiation de vecteur des coéficients du 
numérateur
b=[1 -0.2 0.8]              %initiation de vecteur des coéficients du 
déumérateur
x=randn(1,N);               %génération du bruit blanc
g=tf(a,b,1)                 %la fonction de transfert de filtre
figure(1); rltool(g);       %les poles et les zeros
figure(2); impz(a,b,20)     %la réponse impulsionnelle du filtre RII sur les 20 
premiers
h = filter(a, b, imp);
figure(3); impz(h, 20);
y = filter(a, b, x);        %générer la sortie du système ayant comme entrée un 
bruit
figure(4); stem(x);
figure(5); stem(y);
```



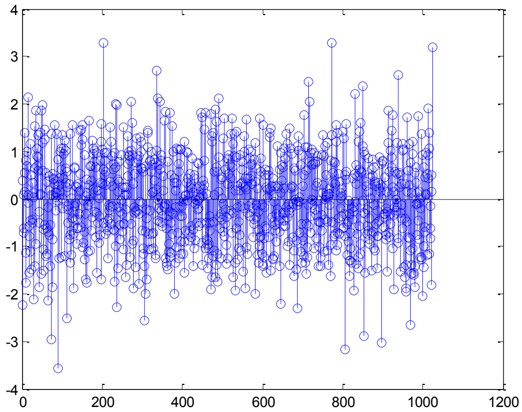
The poles and zeros in the Z-plane

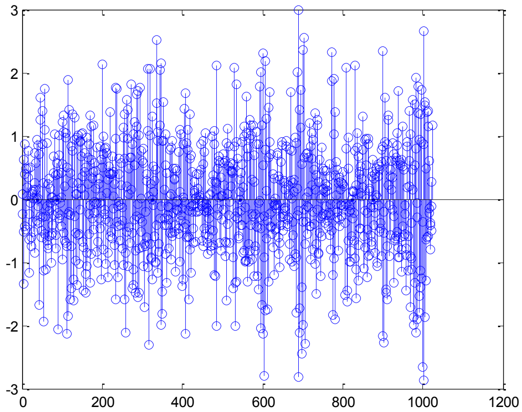


The impulse response of the IIR filter, for the first 20 samples, using the "impz" command.



The impulse response of the IIR filter, for the first 20 samples, using the "filter" command.



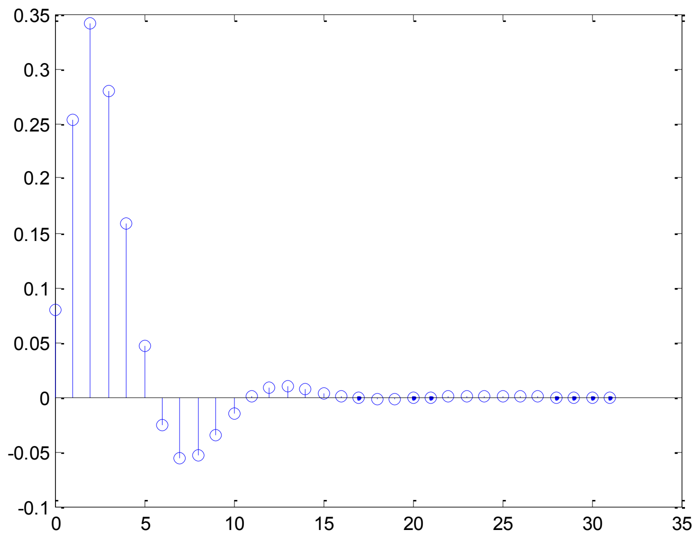


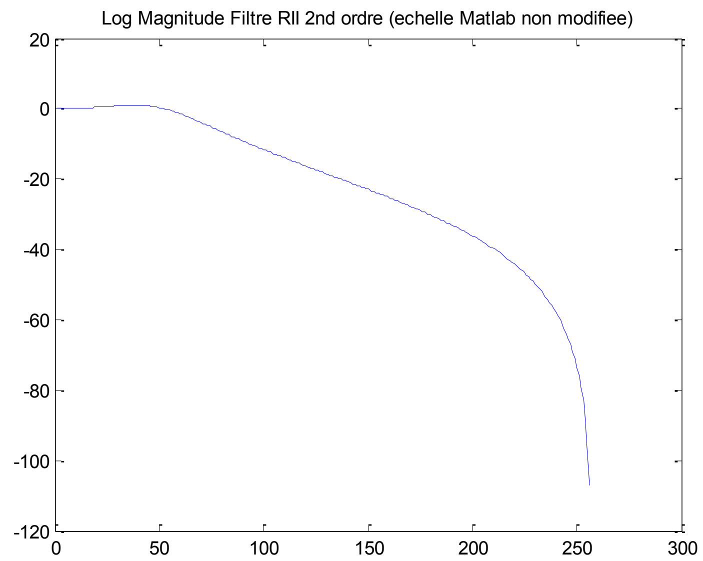
The system's output when fed with white noise as input

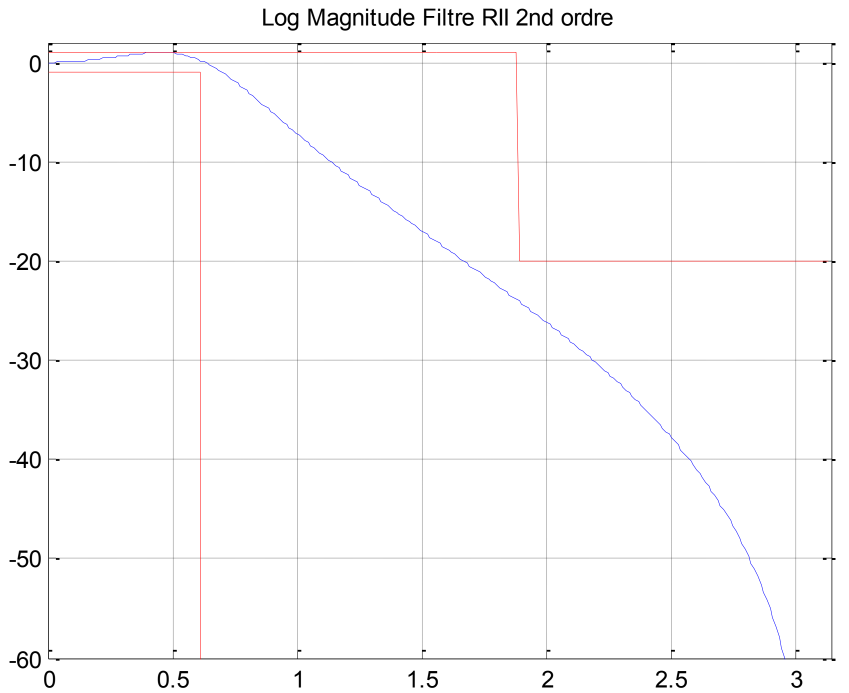
Manipulation 2 :

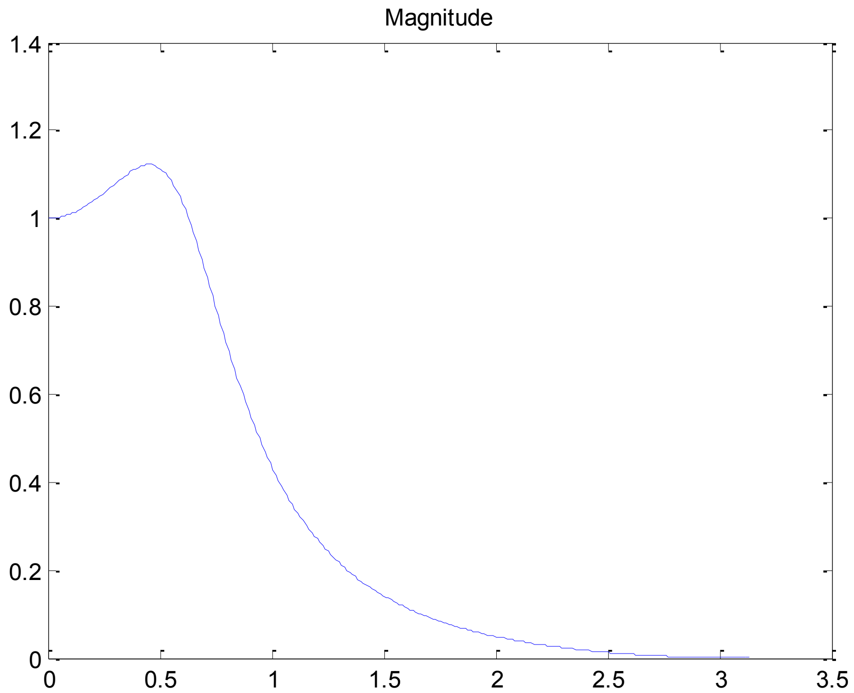
```
close all; %Ferme les figures en cours
% Premiere partie
% 1 Analyse de filtres numériques passe-bas du second ordre
% 1.1 Analyse d'un filtre numérique RII passe-bas du second ordre
b = [0.079 2*0.079 0.079]; %Numérateur
a = [1 -1.2 0.516]; %Dénominateur
N = 32; n=0:N-1;
```

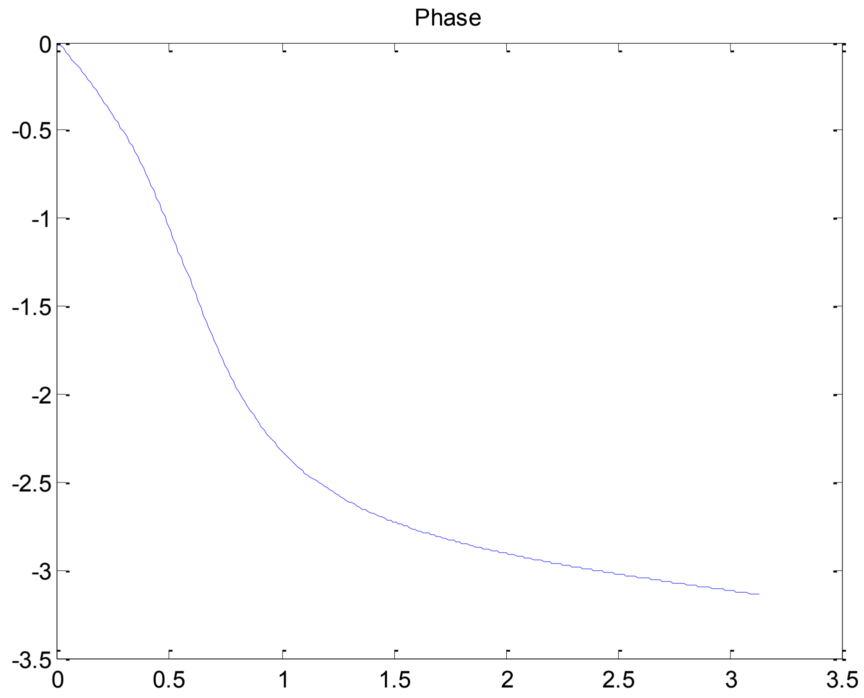
```
delta = [1; zeros(N-1,1)]; %Impulsion
step = ones(N,1); %Echelon unité
h = filter(b, a, delta); %Réponse du filtre (ou impz(b,a);)
figure(1); stem(n,h);
L = 256;
[h,w] = freqz(b,a,L);
m = abs(h); p = angle(h);
figure(2); plot(20*log10(m)); title('Log Magnitude Filtre RII 2nd ordre
(echelle Matlab non modifiee)');
figure(3); plot(w(1:L-1),20*log10(m(1:L-1))); title('Log Magnitude Filtre
RII 2nd ordre');
axis([0 pi -60 2]); grid
figure(4); plot(w,m); title('Magnitude'); figure(5); plot(w,p);
title('Phase');
figure(6); zplane(b,a);
zero = roots(b); pole = roots(a);
Delta1 = 1; Delta2 = -20;
Fe = 10000;
Fp = 1000; NFp = round(L*Fp/(Fe/2));
Fs = 3000; NFs = round(L*Fs/(Fe/2));
gabh = [Delta1*ones(NFs,1); Delta2*ones(L-NFs,1)];
gabl = [-Delta1*ones(NFp,1); -5000*ones(L-NFp,1)];
figure(3); hold on; plot(w,gabh,'r'); plot(w,gabl,'r');
```

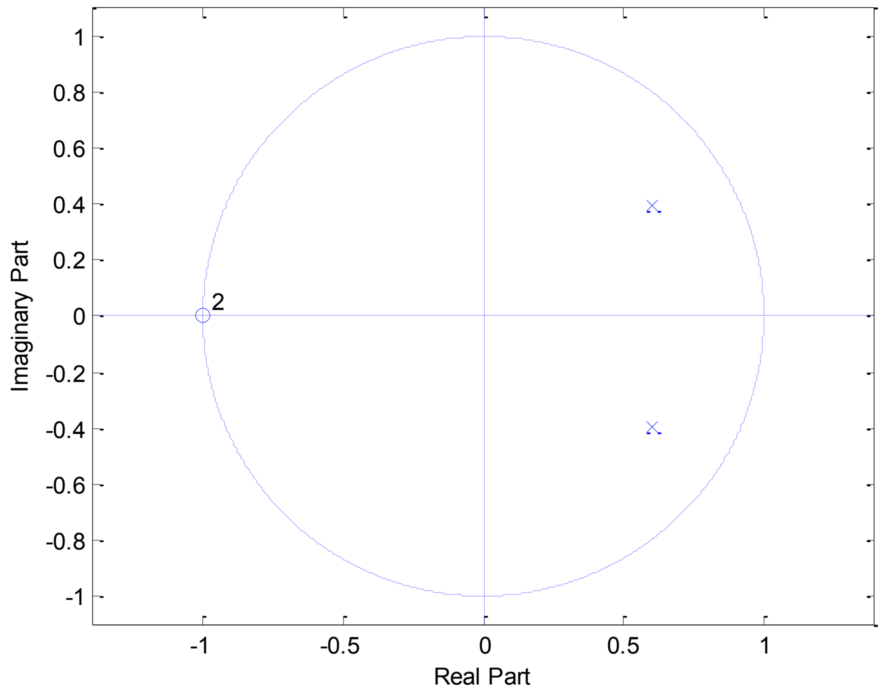





Log Magnitude Filtre RII 2nd ordre
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3


Magnitude
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
3
3.5


Phase
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
0
0.5
1
1.5
2
2.5
3
3.5


2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Imaginary Part
-1
-0.5
0
0.5
1
Real Part

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Mohamed BOUDIAF University - M'sila

جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

Faculty of Technology
Department of Electrical Engineering

MODULE: *Signal Processing*

ACADEMIC YEAR: *1st Year Master Automation and Systems*

TEACHER: *Dr. Abdelghafour HERIZI*

Practical Work 5 : FIR Digital Filtering

I. Objective of the Practical Work:

This practical work is based on the concept of a finite impulse response (FIR) linear process. We will alternate between simulation programs for the impulse response using two methods: the "**filter**" command and the "**impz**" command. , as well as the system's response to a noisy input and the representation of poles and zeros.

II. Reminder:

II.1 Finite Impulse Response (FIR) Filter:

These filters can be implemented using the following formula:

$$y(n) = \sum_{k=0}^{L} b_k x(n-k)$$

This type of filter is called finite because its impulse response will ultimately settle to zero. An FIR filter is non-recursive, meaning that the output depends only on the input signal; there is no feedback. Thus, the coefficients a of the general form of digital filters are all equal to zero. An important property of FIR filters is that the filter coefficients *b* are equal to the impulse response h of the filter. Furthermore, the time-domain form of the filter is simply the convolution of the input signal x with the coefficients (or impulse response) b (or h).

II.2 II.2 Characteristics of Finite Impulse Response Filters:

- The finite impulse response (FIR) filter corresponds to the previous structure: the output sample $y(n)$ depends only on the input samples $x(n), x(n-1), x(n-2), \ldots x(n-p)$.
- This type of filter is, by design, always stable.
- Causal if their impulse response $g(k)$ is zero for $k < 0$. If they are not, it is sufficient to shift this response.
- Can be linear phase.

III. Manipulations :

III.1 Manipulation 1 :

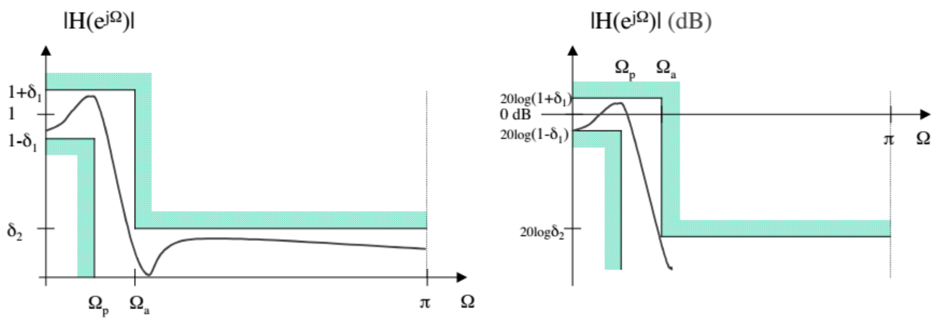
A finite impulse response (FIR) linear process, described by the following transfer function:

$$H(z) = 0.5 + 0.75z^{-1} + 1.2z^{-2}$$

1. Represent the poles and zeros in the Z-plane.
2. Represent the impulse response of the FIR filter over the first 20 samples using the "impz" command.
3. Represent the impulse response of the FIR filter over the first 20 samples using the "filter" command.
4. Represent the output of the system with white noise as input.

III.2 Manipulation 2 :

The template of the low-pass filter is defined as in the figure below:



b) Linear frequency response b) Frequency response in dB

With:

$Fp = 1kHz, Fs = 3kHz, Fe = 10kHz$
$\Delta_1 = 20\log(1+\delta_1) = 1dB$
$\Delta_2 = 20\log(\delta_2) = -20dB$

The filter synthesis gives two transfer functions $H(z)$ depending on the number of points considered ($N = 11$ then $N = 7$):

$$H_{11}(z) = -0{,}0309396.(z^{-1} + z^{-9}) - 0{,}0390182.(z^{-2} + z^{-8}) + 0{,}0766059.(z^{-3} + z^{-7}) + 0{,}288307.(z^{-4} + z^{-6}) + 0{,}4.z^{-5}$$

$$H_7(z) = -0{,}0409365.(1 + z^{-6}) + 0{,}078369.(z^{-1} + z^{-5}) + 0{,}289996.(z^{-2} + z^{-4}) + 0{,}4.z^{-3}$$

1. Write the Matlab program to visualize the magnitude and phase frequency response of the digital filter.
2. Relate the abscissa scale of the Matlab plot command to the frequencies.
3. What, then, are the relationships between sampling frequency and cutoff frequency? . Does the filter meet the specified template?
4. Visualize the impulse response.

IV. Solution to the manipulations:

Manipulation 1 :

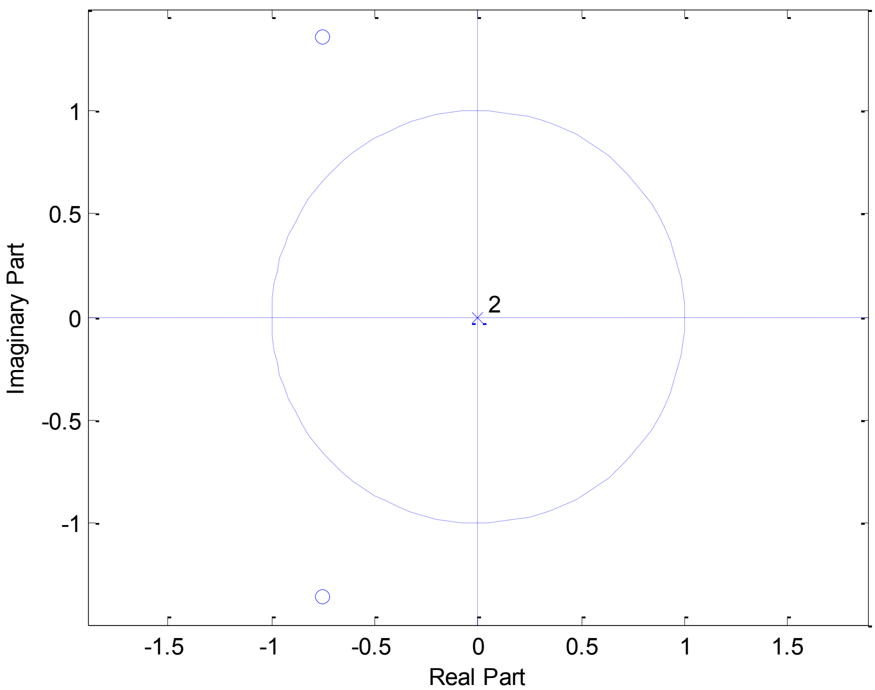
```
close all,clear all;
N=1024;
% nombres de points
imp=[1 0 0 0];
%l'entrée impulsion de Dirac
a=1;
%initiation de vecteur des coéficients du numérateur
b=[0.5 0.75 1.2 0];
%initiation de vecteur des coéficients du déumérateur
x=randn(1,N);
%génération du bruit blanc
y=filter(b,a,x);
%générer la sortie du système ayant comme entrée un bruit blanc
resptf=filter(b,1,imp);
%générer la réponse impulsionnelle
figure(1);zplane(b,a);
%représentation des pôles et des zéros dans le plan Z
figure(2);
impz(b,1,20);
%calcule de la réponse impulsionnelle, avec 20 est la nombre de points
%demandés par la réponse impulsionnelle
figure(3);
plot(resptf,'*');
%représentation de la sortie du système ayant comme entrée une impulsion
figure(4);plot(y);
%représentation de la sortie du système ayant comme entrée un bruit blanc
```

>> zeros

zeros =

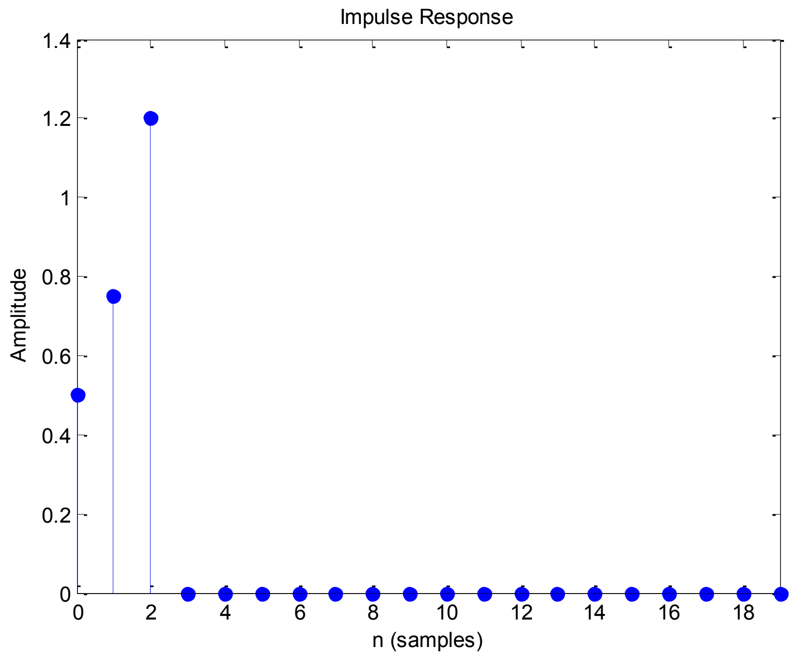
-0.7500 + 1.3555i

-0.7500 - 1.3555i

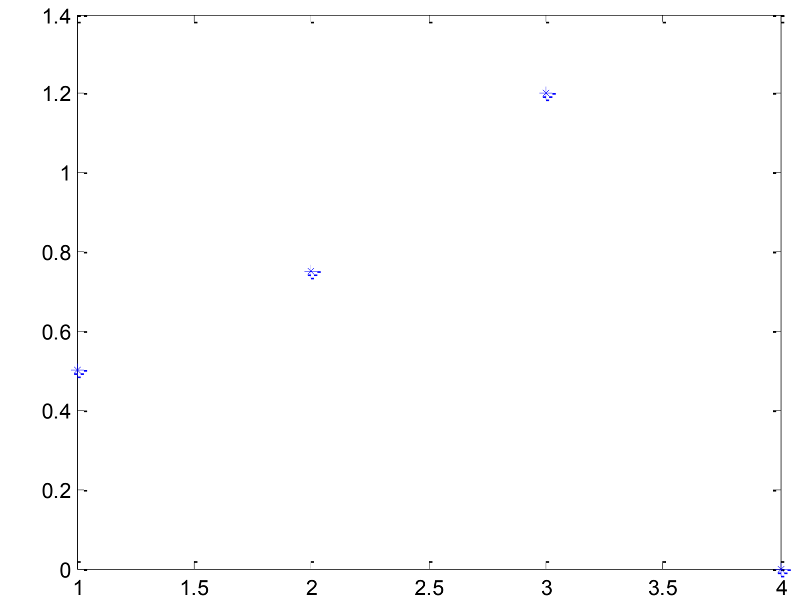


Representation of zeros in the Z-plane

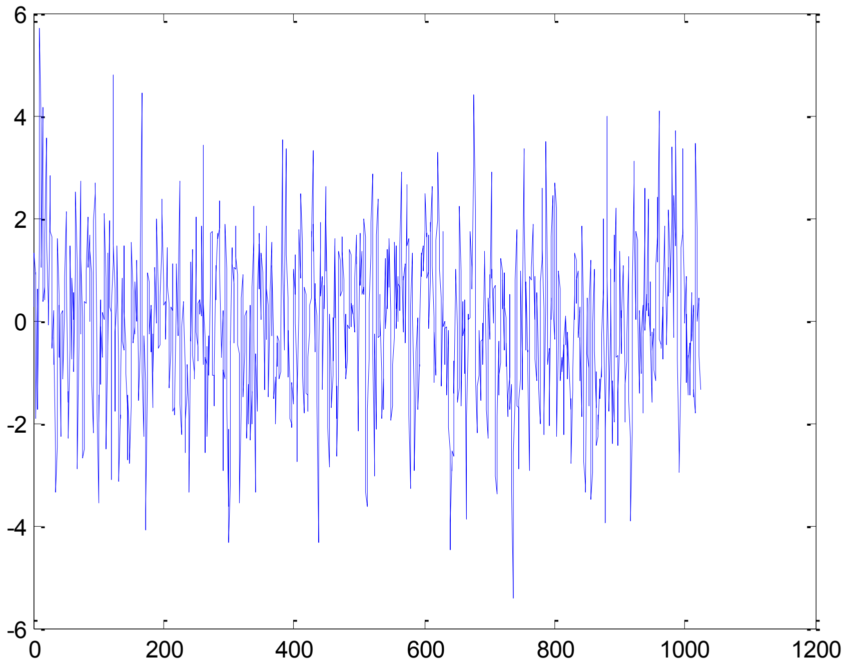
```
>> r=abs(zeros);% module d'un nombre complexe
r =
1.5492
1.5492
>> theta=angle(zeros), % argument d'un nombre complexe
theta =
2.0762
-2.0762
```



Representation of the impulse response of the FIR filter, showing the first 20 values, using the "impz" command.



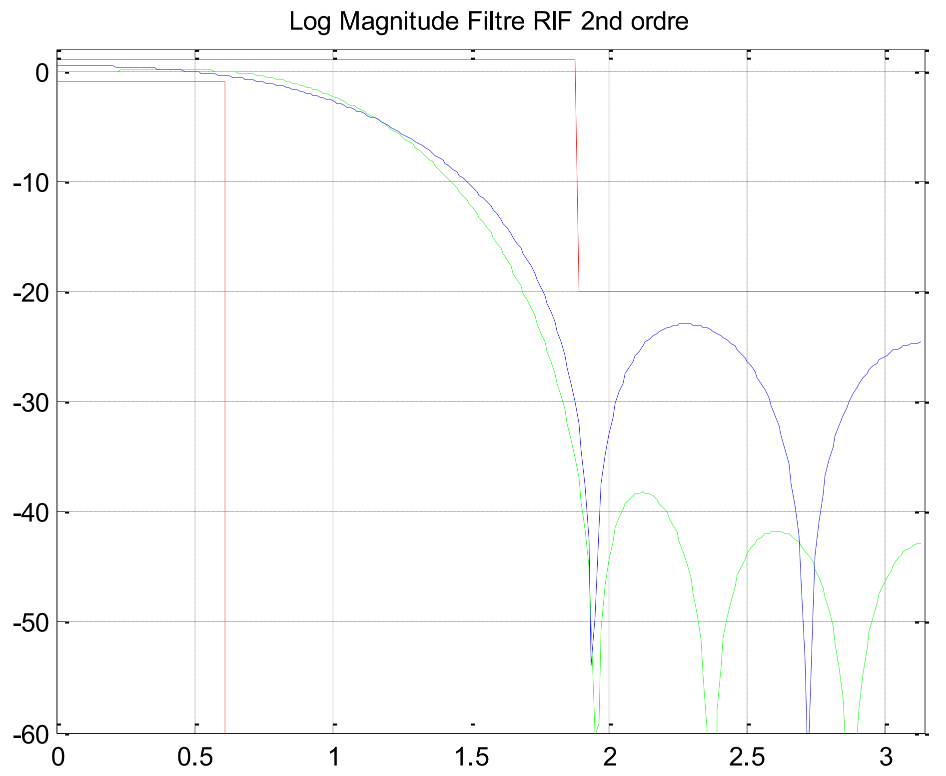
Representation of the impulse response of the FIR filter, showing the first 4 values, using the "filter" command.

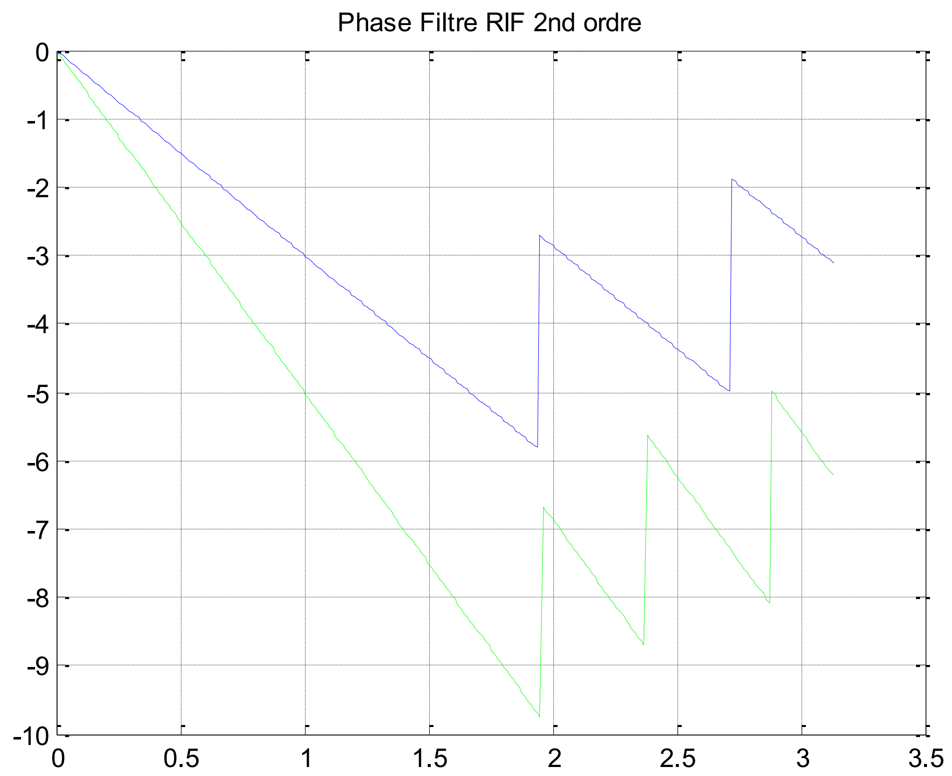


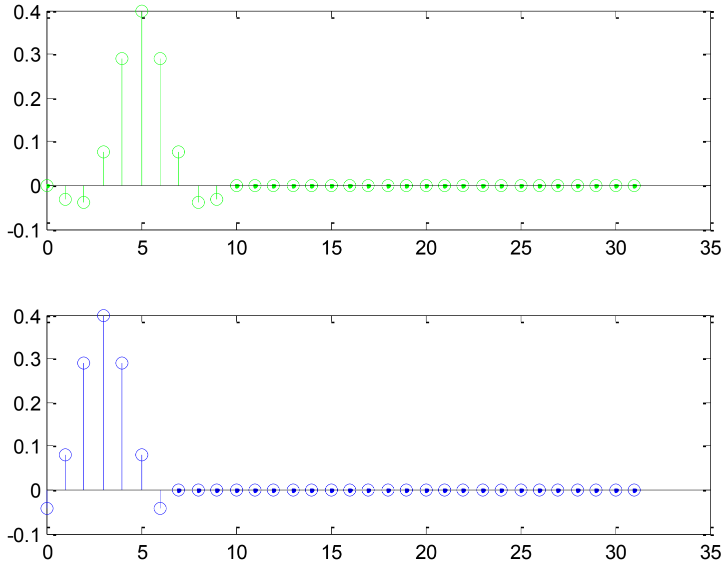
Representation of the system output when the input is white noise.

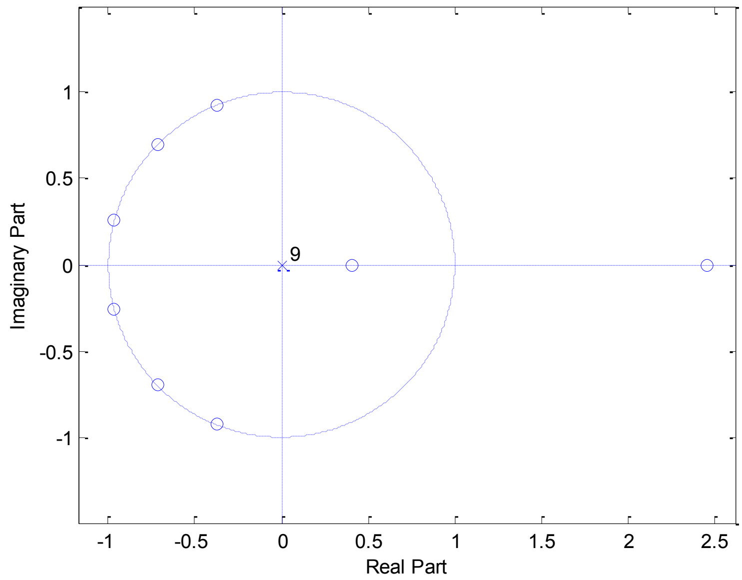
Manipulation 2 :

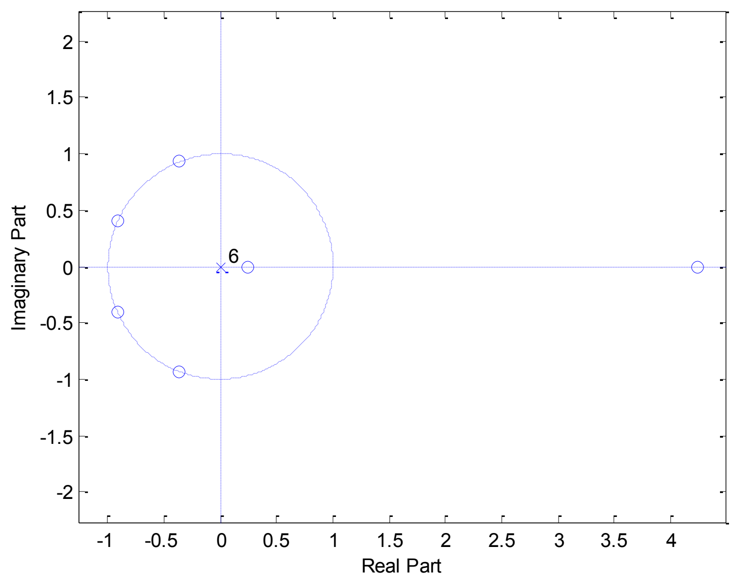
```
close all; %Ferme les figures en cours
% Premiere partie
% 1 Analyse de filtres numériques passe-bas du second ordre
% 1.2 Analyse d'un filtre numérique RIF passe-bas du second ordre
b11 = [0 -0.0309396 -0.0390182 0.0766059 0.288307 0.4 0.288307 0.0766059 -
0.0390182 -0.0309396 0];
b7 = [-0.0409365 0.078369 0.289996 0.4 0.289996 0.078369 -0.0409365];
L=256;
delta = [1; zeros(N-1,1)]; %Impulsion
N = 32; n=0:N-1;
[h11,w] = freqz(b11,1,L);
[h7,w] = freqz(b7,1,L);
figure; plot(w,20*log10(abs(h11)),'g'); title('Log Magnitude Filtre RIF 2nd
ordre');
axis([0 pi -60 2]); hold on;
plot(w,20*log10(abs(h7)),'b'); grid
Delta1 = 1; Delta2 = -20;
Fe = 10000;
Fp = 1000; NFp = round(L*Fp/(Fe/2));
Fs = 3000; NFs = round(L*Fs/(Fe/2));
gabh = [Delta1*ones(NFs,1); Delta2*ones(L-NFs,1)];
gabl = [-Delta1*ones(NFp,1); -5000*ones(L-NFp,1)];
plot(w,gabh,'r');
plot(w,gabl,'r');
figure; plot(w,unwrap(angle(h11)),'g'); title('Phase Filtre RIF 2nd
ordre');
grid; hold on; plot(w,unwrap(angle(h7)),'b');
h11 = filter(b11, 1, delta);
h7 = filter(b7, 1, delta);
figure; subplot(2,1,1); stem(n,h11,'g'); subplot(2,1,2); stem(n,h7,'b')
figure; zplane(b11,1); figure; zplane(b7,1);
```


Log Magnitude Filtre RIF 2nd ordre
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3


Phase Filtre RIF 2nd ordre
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
0
0.5
1
1.5
2
2.5
3
3.5




9
Imaginary Part
Real Part


6
Imaginary Part
Real Part

Bibliography

[1] F. Cottet, "Traitement des signaux et acquisition de données, Cours et exercices corrigés", 4ième édition, Dunod, Paris, 2015.

[2] T. Neffati, "Traitement du signal analogique : Cours", Ellipses Marketing, 1999.

[3] M. Benidir, "Théorie et traitement du signal : Méthodes de base pour l'analyse et le traitement du signal", Dunod, 2004.

[4] M. Bellanger, "Traitement numérique du signal : Théorie et pratique", 9ième édition, Dunod, Paris, 2012.

[5] E. Tisserand, J. Pautex & P. Schweitzer, "Analyse et traitement des signaux méthodes et applications au son et à l'image", 2ième édition, Dunod, Paris, 2008.

[6] P. Duvaut, F. Michaut & M. Chuc, "Introduction au traitement du signal - exercices, corrigés et rappels de cours", Hermes Science Publications, 1996.

[7] M. Bousahla & F. Derraz, "Travaux Pratiques : Taitement numérique du signal", Polycopie, Université Abou Bekr Belkaid–Tlemcen, 2020.

[8] S. Tahraoui, "Travaux Pratiques : Taitement du signal", Polycopie, Université Hassiba Benbouali de Chlef, 2023.

[9] F. Cottet, "Traitement des signaux et acquisition de données, Cours et exercices corrigés", 4ième édition, Dunod, Paris, 2015.

[10] M. Bellanger, "Traitement numérique du signal", Edition DUNOD, 1998.