

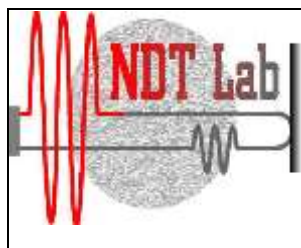


UNIVERSITE MED SEDDIK BEN YAHIA– JIJEL

Faculté des Sciences et de la Technologie

Département D'Automatique

## MASTER I (AUTOMATIQUE ET SYSTEME : AS/S1) TRAITEMENT DU SIGNAL (ANALOGIQUE ET NUMERIQUE)



### TP N°1/2 REPRESENTATION DES SIGNAUX (TEMPORELLE/FREQUENTIELLE)

.... /.... / 2021  
Rapport réalisé par : .....

#### I) PRESENTATION

Chaque Tp de la matière traitement du signal (analogique et numérique) se compose de trois parties distinctes:

- 1) Une partie théorique qui rend le TP indépendant de l'avancement du cours.
- 2) Une partie de préparation qui permet à l'étudiant de se pencher sur tous les points de son TP et de résoudre les exercices proposés.
- 3) Une partie dite "Manipulation" qui permet à l'étudiant de vérifier pratiquement sur machine les différentes notions étudiées en ce TP.

#### II) DEROULEMENT DU TP ET RECOMMANDATIONS

##### 1) PREPARATION :

Le temps imparti pour la manipulation étant limité, il est important que les étudiants s'astreignent à résoudre les problèmes posés dans les temps en se présentant avec une préparation (Individuelle), contenant entre autres la solution des exercices proposés, et les programmes MATLAB demandés. L'étudiant, pourra procéder à des tests et vérifications; pour s'assurer que les solutions et les programmes de la préparation sont correctes.

##### 2) MODALITES DE DEROULEMENT

Les TP nécessitant un logiciel mathématique de calcul numérique, et peuvent être réalisés sous Matlab, Octave ou Scilab (Dans notre cas c'est le **Matlab qui sera utilisé**).

**Le Matlab** est un logiciel commercial, **Octave et Scilab** sont des logiciels librement utilisables (open source). **Octave est un clone de Matlab** avec une syntaxe très proche et des programmes fortement compatibles. Scilab a diverses particularités et une syntaxe quelque peu différente. Les aspects textuels (commentaires, chaînes de caractères, ... etc.) et graphiques de Matlab et Scilab sont pratiquement incompatibles.

##### 3) SEANCE DE TP :

A chaque séance, il vous sera demandé :

1. D'écrire un ou plusieurs **script(s)** mettant en œuvre les appels aux fonctions (Matlab) pour résoudre certains problèmes du traitement du signal.
2. De décrire les problèmes examinés, les opérations effectuées par le logiciel de calcul numérique, et les résultats obtenus.

Les étudiants doivent remettre le compte rendu du TP (Rapport écrit+programmes) **à la fin de la séance**, ou dans le cas échéant lors de **la prochaine séance de TP**.

##### 4) EVALUATION

- \* Préparation et compréhension des exercices et Travail effectué lors des séances de TP (40%).
- \* Rapports de synthèse et Test de TP final (60%).

## Fontcions Matlab :

Fonction Echelon	Fonction Rectangle
<pre>clear all % définition du vecteur temps % Définition de la fonction echelon echelon = @(u) (u&gt;0);</pre>	<pre>% Définition de la fonction Rectangle Rectangle = @(u,tau) (abs(u)&lt;= tau/2) ; % Equivalent a : % fonction [signal] = Rectangle(u,tau) % dans un fichier séparé</pre>

## Fenêtres rectangulaire et triangulaire

<pre>function [W,NN]=tp12Rect(A,k0,T); % Génération d'une impulsion rectangulaire Wr(k) % centrée sur k0 et de largeur 2*T+1 N0=T ;N1=2*T+1; NN=(k0-T-N0:1:k0+T+N0)'; L=length(NN); % L=N1+2*N0; % 1° méthode W(:,1)=A*[zeros(N0,1);ones(N1,1);zeros(N0,1)]; % 2° méthode center=find(NN==k0); W(:,2)=zeros(L,1); W((center-T):(center+T),2)=A;</pre>	<pre>function [Wt,NN]=tp12Rect(A,k0,T); % Génération d'une impulsion triangulaire Wt(k) % centrée sur k0 et de largeur 2*T+1 NN=(k0-T:1:k0+T)'; % 1° méthode W=A*(0:T)/T; W=[W W(T:-1:1)']; Wt(:,1)=W % 2° méthode center=find(NN==k0); K=k0-T:k0+T W(center-T:center+T)=A*(1-abs(K-k0)/T) Wt(:,2)=W</pre>
---	--

## Fonction Sinc

<pre>function y = sincxl( x, L ) %ASINC compute sin(L.x/2)/sin(x/2) (for matrix x) % Usage: y = sincxl(x, L) % copyright 1994, by C.S. Burrus, J.H. McClellan, A.V. Oppenheim, y = sin(0.5*x); b01 = abs(y) &lt; 1e-10; y = b01*L + (~b01).*sin((L/2)*x)./(y + b01);</pre>	<pre>function Y=sinc(X) %calcul de sin(x)/x Y=sin(X)./X; pos=find(X==0) ; Y(pos)=1; End disp('Calcul et tracé de la fonction sinc'); clear; clc; % x=input('Intervalle de définition (sous la forme x1:pas:x2) : '); x=linspace(-30,30,200); y=sinc(x); plot(x,y)</pre>
--	---

## Fonction fmagplot : Visualisation de la TF d'un signal analogique

<pre>function fmagplot( xa, dt ) %FMAGPLOT Plot Fourier Transform (Mag) of "ANALOG" signal % copyright 1994, by C.S. Burrus, J.H. McClellan, A.V. Oppenheim, % "Computer-Based Exercises for Signal Processing /MATLAB" % Usage: fmagplot( xa, dt ) % xa : "ANALOG" signal % dt : sampling interval for the simulation of xa(t)</pre>	<pre>L = length(xa); Nfft = round( 2.^nextpow2(5*L) ); Xa = fft(xa, Nfft); range = 0:(Nfft/4); ff = range/Nfft/dt; plot( ff/1000, abs( Xa(1+range) ) ) title('FOURIER TRANSFORM (MAG)') xlabel('FREQUENCY (kHz)'), grid</pre>
---	---

## Exemple 1:

<pre>Génération d'un train d'impulsions x1=[0;1;0;0;0] ; x2=x1 ones(1,7) x3=x2(:) ; size(x3) pause subplot(211);stem(x1); subplot(212);stem(x3);</pre>	<pre>function [Y,TT]=sinus(A,w,phi,t1,t2); % Génération de s(t)=A*Sin(w*(t1:t2)+ phi) % t=n.Te, ==&gt; s(n)=A*Sin(w*(n1:n2)*Te+ phi) TT=linspace(t1,t2, 200); Y=A*sin(w*TT+phi);</pre>
--	--

## Exemple 2:

<pre> T = 5; a = 3 ; n = 100 ; t = linspace(-5,5,n) Signal1 = Rectangle (t-1,T) ; Signal2 = t.*echelon(t) ; Signal3 = (t-2).*echelon(t-3) ; Signal4 = exp(-a*t).*echelon(t-1); figure(1) ; subplot(2,2,1); plot(t,signal1) ; title('Signal 1') ; ylim([0 1.5]) ; xlabel('Temps') ; ylabel('Signal'); subplot(2,2,2) ; plot(t,signal2) ; title('Signal 2') subplot(2,2,3) ; plot(t,signal3) ;title('Signal 3') subplot(2,2,4) ; plot(t,signal4) ;title('Signal 4') </pre>	<pre> L=50; k0=0;a=1; %initialisation nn=0:(L-1); %intervalle imp=zeros(L,1); imp(k0+1)=a; stem(nn,imp) </pre>
--	--

### Exemple 3 :

Compléter le code Matlab suivant pour représenter sur le même graphe (où dans une même fenêtre graphique) les variations de trois signaux de base (sinusoïdaux) ainsi que leur combinaison.

Code Matlab	<pre> % définition des paramètres f0 = 0.51; A0 = 1; f1 = 0.11; A1 = 2; f2 = 0.21; A2 = 2; % déclaration de signaux de base x0 = A0*sin(2*pi*f0*t); x1 = A1*sin(2*pi*f1*t); x2 = A2*sin(2*pi*f2*t); x3=x0+x1+x2 ; % combinaison % affichage des signaux plot(t, x0, 'y'); plot(t, x1, 'g'); plot(t, x2, 'c'); plot(t, x3, 'k.');</pre>
----------------	--

### Exemple 4:

Réécrire le script suivant pour le transformer en une fonction bsinus() avec paramètres afin de générer une sinusoïde de fréquence f0 bruitée avec un bruit blanc gaussien de variance Var.

```

t = (0:0.001:1)'; % Te
y = 3*sin(2*pi*80*t) % F1
yn = y + 0.5*randn(size(t)); % Var
plot(t(1:50),yn(1:50)) % Ne

```

## Exemple 5

```
clear all
% représentation temporelle du signal
t = (-5:0.01:5); a = 1 ;
signal = exp(-a*abs(t));
subplot(3,1,1) ;plot(t,signal)
f = (-5:0.01:5) ;
TFCalculée = 2*a./(a^2+(2*pi*f).^2);
subplot(3,1,2)
plot(f, TFCalculée)

%% Calcul d'une TF par Matlab
Te = 0.01 ; TFx = fftshift(Te*fft(signal));
f2 = (-1/(2*Te):(1/round(length(signal)*Te)):1/(2*Te));
subplot(3,1,3) ; plot(f2,abs(TFx))
xlim([-5 5])
xlabel('Frequence'); ylabel('Amplitude')
```