

But :

Ce TP a pour but de familiariser l'étudiant avec l'utilisation du logiciel Matlab pour la génération des différents types de formes d'ondes.

The *Signal Processing Toolbox* de MATLAB contient des fonctions pour la génération des formes d'ondes couramment utilisées périodiques et non périodiques, séquences (impulsion, échelon, rampe), train d'impulsions...

Exercice :

1. Génération des signaux : Impulsion Unité et Echelon Unité

Deux signaux élémentaires de base sont : l'impulsion Unité et l'échelon Unité.

L'impulsion unité $u[n]$ de longueur N peut être générée en utilisant la commande Matlab suivante :

$u = [1 \text{ zeros}(1, N-1)]$;

De même, l'impulsion unité $u_d[n]$ de longueur N et décalées de M échantillons, tel que $M < N$, peut être générée par la commande MATLAB suivante :

$u_d = [\text{zeros}(1, M) \ 1 \ \text{zeros}(1, N-M-1)]$;

L'échelon unité $s[n]$ de longueur N peut être généré en utilisant la commande Matlab suivante :

$s = [\text{ones}(1, N)]$

La version décalée de l'échelon unité s'obtient de la même manière que dans le cas de l'impulsion unité.

Le programme P1 peut être utilisé pour générer et tracer un signal impulsion unité :

%Programme P1

% Génération d'une impulsion unité

% Génération d'un vecteur de -10 à 20

$n = -10:20$;

% Génération de l'impulsion unité

$u = [\text{zeros}(1, 10) \ 1 \ \text{zeros}(1, 20)]$;

% Tracer le signal généré

$\text{stem}(n, u)$;

$\text{xlabel}(\text{'Temps indexe en n'})$;

$\text{ylabel}(\text{'Amplitude'})$;

$\text{title}(\text{'Impulsion Unite'})$;

$\text{axis}([-10 \ 20 \ 0 \ 1.2])$;

Travail demandé 1 :

1- Modifier le programme précédent pour générer une impulsion unité décalée de 11 échantillons. Exécuter le programme et tracer le signal généré.

2- Modifier le programme précédent pour générer un échelon unité décalé de 20 échantillons.

2. Génération des signaux périodiques

Une autre classe de signaux très utile en traitement du signal et en automatique sont les signaux sinusoïdaux de la forme : $x[n] = A \cdot \cos(\omega n + \varphi)$

Avec Matlab, ce type de signaux peut être généré en utilisant les opérateurs trigonométriques cos et sin.

Le programme P2 est un exemple qui génère un signal sinusoïdal.

```
*****
% Programme P2
% Generation d'un signal sinusoidal
n = 0:100;
f = 0.1;
phase = 0;
A = 1.5;
arg = 2*pi*f*n - phase;
x = A*cos(arg);
clf; %efface l'ancienne figure
stem(n,x);
axis([0 100 -2 2]);
grid;
title('Signal Sinusoidal');
xlabel('Temps indexe en n');
ylabel('Amplitude') ;
axis ;
*****
```

Travail demandé 2 :

1- Modifier le programme précédent pour générer et tracer un signal sinusoïdal de longueur 50, fréquence 0.08, amplitude 2.5 et phase décalée de 90 degrés.

2- Générer et tracer les séquences définies ci-dessous :

$$x_1(n) = \sin\left(\frac{\pi}{17}n\right) \quad 0 \leq n \leq 35$$

$$x_2(n) = \sin\left(\frac{\pi}{17}n\right) \quad -15 \leq n \leq 25$$

$$x_3(n) = \sin\left(3\pi n + \frac{\pi}{2}\right) \quad -15 \leq n \leq 15$$

$$x_4(n) = \sin\left(\frac{\pi}{\sqrt{23}}n\right) \quad 0 \leq n \leq 50$$

3. Comparaison entre un signal continu et un signal discret :

Exécuter sur Matlab le programme P3 suivant :

```
*****
% Programme P3
% Génération d'un signal continu et un signal discret
x = 1:0.1:2;
y=sin(x);
clf; % efface l'ancienne figure
stem(x,y);
grid;
xlabel('Temps');
ylabel('Amplitude') ;
hold on
plot(x,y, 'r') % 'r' : couleur rouge
*****
```

Travail demandé 3 :

1- Ecrire un programme pour générer et tracer un signal sinusoïdal de longueur 10, fréquence 0.6, amplitude 5 et phase décalée de -90 degrés.

1. Soit le système continu suivant :

$$G(p) = \frac{1}{p^2 + 0.15p + 0.01}$$

- Définir la fonction de transfert $G(p)$ du système sous Matlab (utiliser les commandes *num*, *den* et *tf*).
- Calculer les pôles du système (racines du dénominateur), (utiliser la commande *roots* ou *pole*).
- Tracer avec Matlab le lieu des racines de $G(p)$ (utiliser la commande *rlocus*). Vérifier la stabilité de ce système.
- Ce système est-il stable ? Pourquoi.
- Transformée en z avec BOZ (Bloqueur d'Ordre Zéro), prenez le temps d'échantillonnage $T_e = 1$, utiliser la commande *c2d*.
- Calculer l'amplitude de chaque pôle.
- Ce système est-il stable ? Pourquoi.
- Vérifier la stabilité de système discret par le lieu des racines de $G(z)$ utiliser la commande *rlocus*.
- Tracer dans une même figure la réponse de système continu et discret, pour une entrée en échelon unitaire.
- Ce système est-il stable ? Pourquoi.

2. Représenter le système $G(p)$ sous la forme de la représentation d'état, utiliser la commande *tf2ss*.

- Représenter $G(p)$ comme un système continu sous Simulink, utiliser ses matrices A, B, C et D comme des blocs. (Voir figure 1)

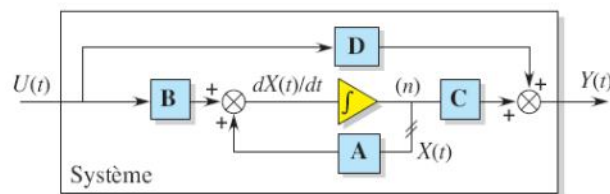


Figure 1

- Représenter $G(p)$ comme un système discret sous Simulink, utiliser ses matrices A, B, C et D comme des blocs. (Voir figure 2)

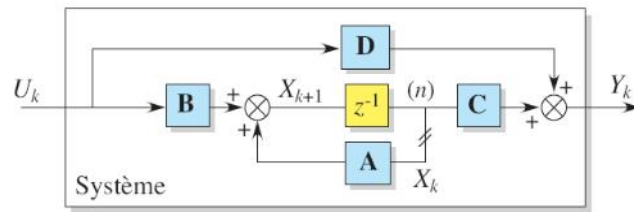


Figure 2