



Systèmes et convolution en discret

Durée: 4 périodes (1 séance de laboratoire)
Travail **individuel**

1. Objectifs

L'exercice a pour objectifs la compréhension et la mise en œuvre des concepts suivants :

- Utilisation de Matlab en général et de fonctions spécifiques en particulier.
- Réponse à l'impulsion unité, le saut unité et leurs relations.
- Convolution discrète.

2. Environnement

La donnée et les fichiers additionnels éventuels sont disponibles sous Moodle, cours *Signaux & Systèmes*, section *Exercices dirigés*, dossier *ExMatlab2*.

Le travail à réaliser sera fait entièrement dans l'environnement de Matlab à l'aide d'un ou de plusieurs script(s).

3. Travail à réaliser

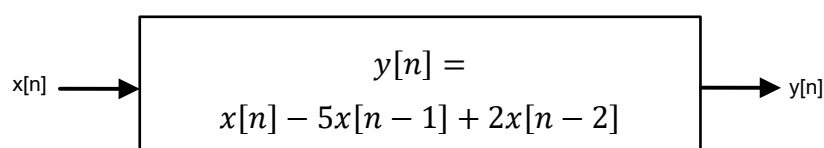
Créez un nouveau script Matlab et sauvez-le en le nommant de la manière suivante:

`ex2_<votre nom>.m`

Copiez la fonction `delta()` que vous avez développée durant le labo1 dans votre répertoire de travail.

3.1 Implémentation de système discret (FIR/IIR)

1) Soit le système discret du 2^{ème} ordre suivant:



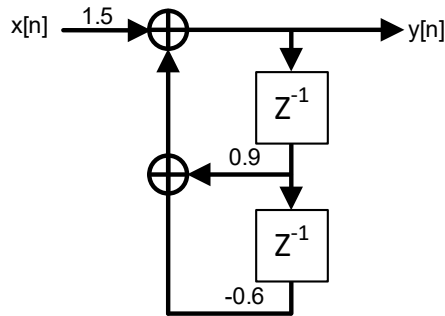
Implémentez-le avec une fonction Matlab en créant une fonction que vous nommez `sys1()`.

Déterminez théoriquement la réponse impulsionnelle $h_1[n]$ du système. Puis, afin de tester votre fonction, mettez-y en entrée une impulsion unité $\delta[n]$ en utilisant votre fonction

développée durant le laboratoire 1 et générez la réponse impulsionnelle de $-10 < n < 10$. Comparez votre réponse à celle donnée par simulation.

Observez et analysez la réponse impulsionnelle. Le système est-il stable? avec ou sans mémoire? qu'en est-il de sa causalité? Justifiez.

2) La fonction `sys2()` a déjà été développée pour simuler le système suivant:



Déterminez théoriquement la réponse impulsionnelle $h_2[n]$ de ce système pour quelques points. Puis, contrôlez votre résultat en générant la réponse impulsionnelle par simulation pour $-10 < n < 10$.

Analysez la réponse impulsionnelle. Le système est-il stable? avec ou sans mémoire? peut-on déterminer son ordre? qu'en est-il de sa causalité? Justifiez.

3.2 Réponse de systèmes discrets et convolution

Créez le signal $x[n]$ suivant pour $-100 < n < 100$:

$$x[n] = \text{square}\left[\frac{n}{2\pi}\right] \cdot u[n]$$

La fonction `square()` de Matlab permet de générer la partie rectangulaire du signal. Pour créer le signal $u[n]$, partez de votre fonction `delta()` développée durant le 1^{er} laboratoire et $u[n] = \sum_{-\infty}^{\infty} \delta[n]$.

Utilisez votre fonction `sys1()` et la fonction `sys2()` pour déterminer les réponses des deux systèmes au signal d'entrée $x[n]$ que vous venez de construire.

Puis, **en réutilisant les réponses impulsionnelles $h_1[n]$ et $h_2[n]$ que vous avez déterminées précédemment en point 3.1** ainsi que la fonction `conv()` de Matlab, calculez $y_1[n] = x[n] * h_1[n]$ et $y_2[n] = x[n] * h_2[n]$

Affichez tous les résultats sur une même figure de $-5 < n < 100$. Comparez les sorties générées par les fonctions et par vos convolutions pour les deux systèmes, qu'en déduisez-vous?

3.3 Convolution sonore

En vous basant sur l'exemple de commandes Matlab ci-contre, chargez et écoutez les sons des fichiers flute.wav et ballonEglise.wav disponibles sur Moodle.

Puis convolvez les deux signaux sonores et normez le signal résultant par sa valeur maximum. Enfin, écoutez le résultat obtenu. Qu'en concluez-vous ?

```
[flute,Fs] = audioread('flute.wav');  
sound(flute,Fs);
```

4. Optionnel

- 1) Créez une fonction sys3() de moyenne glissante paramétrable comme ci-contre. Puis chargez le fichier music.mat à l'aide de la fonction load. Utilisez alors la commande Matlab sound() pour écouter le son enregistré, puis appliquez votre moyenne glissante au signal et ré-écoutez-le. Quel est l'effet de votre système sur la musique?

```
function [ out ] = sys3( x, N)  
% system : y[n]=1/N*sum(x[n-N+1:n])  
    ...  
end
```

5. Références

Notes du cours Signaux & systèmes 1, chapitres "Signaux" et "Systèmes".

Aide en ligne de Matlab : <https://ch.mathworks.com/matlabcentral/>