TD8 ESILV Mai 2020: Analyse des filtres numériques et démodulation d'un signal codé en FSK

Instructions :

- vous pouvez travailler en binômes (dans un même groupe de TD) ;
- l'ensemble du TD se fait sous Matlab ;
- il faut déposer un seul document au format pdf détaillé sur ce **TD8** ainsi que la partie **TP C du TD7** avec les explications et les figures ainsi que le code Matlab bien commenté en annexe de ce rapport dans l'espace de dépôt dédié au Traitement du Signal sur DVO BrightSpace dans un délai maximal de 5 jours après cette séance.
- le nom du rapport doit être comme suivant:
NomEtudiant1_PrenomEtudiant1_NomEtudiant2_PrenomEtudiant2_TDXX
- si on constate des similitudes importantes (plagiat) entre deux rapports, une note de zéro sera attribuée aux deux rapports : les rapports de toute la promotion seront testés avec un détecteur automatique de plagiat.

L'objectif de ce TD est d'étudier plusieurs types de filtres numériques. Pour chaque filtre, vous devez visualiser les réponses indicielle, impulsionnelle et fréquentielle ainsi que les positions des pôles dans le plan en Z. Ces filtres seront utilisés dans la dernière partie pour démoduler un signal numérique codé en fréquence **FM - FSK**.

1 Filtre purement récursif

Soit le filtre purement récursif de premier ordre avec la fonction de transfert

$$H(Z) = \frac{1-a}{1-aZ^{-1}}$$

1) Déterminer la condition sur a pour que ce filtre soit stable.

2) Déterminer l'équation aux différences associée à ce filtre. On note $x(n)$ l'entrée et $y(n)$ la sortie.

3) Ecrire un programme qui pour chaque valeur de a :

- implémente l'équation aux différences jusqu'à n=200 ($n \geq 0$)
- crée un signal (vecteur) de dimension 200: $x_1(n) = \delta(n)$ qui correspond à une impulsion de dirac (attention dans Matlab l'indice le plus petit d'un vecteur est 1, décale tout de 1):

$$x_1(0) = 1 \; si \; n = 0$$
$$x_1(n) = 0 \; si \; n \neq 0$$

- calcule la réponse impulsionnelle $y_1(n)$ (réponse du filtre à $x_1(n)$) jusqu'à n=200 en utilisant l'équation aux différences
- crée un signal (vecteur) de dimension 200: $x_2(n) = u(n)$ qui correspond à un échelon ($n \geq 0$):

$$x_2(n) = 1$$

- calcule la réponse indicielle $y_2(n)$ (réponse du filtre à $x_2(n)$) jusqu'à n=200 en utilisant l'équation aux différences
- affiche le module de $H(f)$ en dB avec une échelle logarithmique sur l'axe des fréquences (utilisez la fonction*semilogx*). On prend une fréquence d'échantillonnage $f_e = 1\,Hz$ et on discrétise $f \in [-\frac{f_e}{2}\;\frac{f_e}{2}]$ sur $N = 200$ points. Pour rappel $df = f_e/N$; $z = e^{j2\pi f T_e}$ avec $T_e = 1/f_e$
- crée un signal (vecteur) de dimension 200: $x_3(n) = \sin(2\pi f_s n T_e)$ où $T_e = \frac{1}{f_e}$ est la période d'échantillonnage. On prend $f_e = 1\,Hz$ et $f_s = 0.1\,Hz$
- calcule la réponse du filtre à un signal sinusoïdal $y_3(n)$ (réponse du filtre à $x_3(n)$) jusqu'à n=200 en utilisant l'équation aux différences. On prend une fréquence d'échantillonnage $f_e = 1\,Hz$
- affiche tous les signaux calculés en fonction du temps $t = nT_e$ avec $f_e = 1\,Hz$ et $T_e = 1sec$
- affiche à nouveau le module de $H(f)$ en dB avec une échelle logarithmique sur l'axe des fréquences à partir de la réponse impulsionnelle y_1 du filtre en utilisant la fonction *fft* de Matlab
- affiche les pôles et les zéros de la fonction de transfert (utilisez les fonctions *tf(num, denum, Te)* et *pzplot*)

4) Tester le programme avec les valeurs de $a \in \{-0.8, 0.8, 0.95, 1.2\}$:

- Quel est le type du filtre dans chaque cas ?
- Est ce que le filtre est stable ?
- Quel est l'impact du filtre sur le signal $x_3(n) = \sin(2\pi f_s n T_e)$? Justifier ce résultat selon le type du filtre.

2 Filtre de premier ordre

On considère un filtre avec la fonction de transfert suivante:

$$H(Z) = \frac{1 - bZ^{-1}}{1 - aZ^{-1}}$$

Ecrire un programme qui pour chaque valeur de a et b:

1) Utiliser la fonction *filter* de Matlab pour calculer et afficher la réponse impulsionnelle et la réponse indicielle de ce filtre (utiliser les mêmes signaux $x_1(n)$ et $x_2(n)$ déjà définis)

2) En déduire et afficher la réponse en fréquence de ce filtre en utilisant la fonction *fft* de Matlab.

3) On utilise ($a = 0.8$ ou $a = -0.8$) et ($b = 1$ ou $b = -1$). Pour les 4 cas possibles, étudier la réponse en fréquence de ce filtre. Quels types de filtres on obtient ?

3 Filtre de second ordre

On considère le filtre de second ordre avec la fonction de transfert suivante:

$$\begin{aligned} H(Z) &= \frac{1}{1 - 2r\cos(2\pi f_0 T_e)Z^{-1} + r^2 Z^{-2}} \\ &= \frac{1}{(1 - re^{2j\pi f_0 T_e} Z^{-1})(1 - re^{-2j\pi f_0 T_e} Z^{-1})} \end{aligned}$$

avec

$$\begin{cases} 0 \leq f_0 \leq \frac{f_e}{2} \\ r > 0 \end{cases}$$

- Déterminer l'expression théorique des pôles de ce filtre. En déduire la condition sur r pour que ce filtre soit stable.
- Utiliser le programme précédant pour réaliser ce filtre: pour chaque valeur de r et f_0:

- trouver et afficher la réponse impulsionnelle et indicielle ainsi que la réponse fréquentielle. On prend $f_e = 1$ Hz.
- affiche les pôles et les zéros de la fonction de transfert (utilisez les fonctions *tf(num, denum, Te)* et *pzplot*).

Application: type du filtre et rôle de f_0 et r:

1) On prend $f_0 = 0.1$ Hz. Varier r entre 0 et 2 avec un pas de 0.5. Que peut-on déduire sur la stabilité de ce filtre ?

2) On prend $f_0 = 0$ Hz. Varier r entre 0 et 1. Quel est le type de ce filtre et quel est l'impact de r ?

3) On prend $f_0 = 0.5$ Hz. Varier r entre 0 et 1. Quel est le type de ce filtre et quel est l'impact de r ?

4) On prend $f_0 = 0.25$ Hz. Varier r entre 0 et 1. Quel est le type de ce filtre et quel est l'impact de r sur la bande passante de ce filtre ?

4 Application : FSK (Frequency Shift Keying) modulation

Un message binaire est transmis en mode FSK selon le protocole suivant:

- Un 1 est transmis en émettant un signal sinusoïdal avec une fréquence f_1 durant un temps T_s.
- Un 0 est transmis en émettant un signal sinusoïdal avec une fréquence f_2 durant le même temps T_s.
- Ici $T_s = 1$ ms et la fréquence d'échantillonnage est $f_e = 100$ kHz.

1) Charger et afficher le signal *signal.mat* qui contient un signal FSK codé avec $f_1 = 40$ kHz et $f_2 = 20$ kHz. Utiliser la fonction Matlab *load*

2) En utilisant la fonction *fft* de Matlab, afficher et vérifier le contenu fréquentiel de ce signal

3) Filtrer ce signal en utilisant le filtre de second ordre (de la partie 3) qui permet de retrouver le signal d'origine: pour retrouver une fréquence f_i on prend $r = 0.9$ et $f_0 = f_i$ => filtre passe bande. Tracer la réponse en fréquence du filtre.

4) Afficher le signal filtré et retrouver la séquence de 0 et 1 du signal d'origine (avant modulation). Donner l'équivalent de cette séquence binaire en décimal.

5) Mêmes questions (1, 2, 3 et 4) pour le signal *signal2.mat* qui contient un signal FSK codé avec $f_1 = 35$ kHz et $f_2 = 40$ kHz.