2021

Traitement de Signal
TD6
Transformée en Z et filtres numériques

Ce TD a pour but d'étudier la transformée en Z et le filtrage numérique d'un signal numérique.

Un document contenant le **code Matlab** et une **copie des figures** est à remettre au plus tard une semaine après cette séance sur l'espace de dépôt dédiée au TD6 sur DVO.

Ex1 : Transformée en Z d'un signal numérique

Déterminer la TZ de $x(n)$ ainsi que son domaine de convergence.

$$x_1(n) = \left(\frac{1}{5}\right)^n u(n)$$

où

$$u(n) = \begin{cases} 1 \; si \; n \geq 0 \\ 0 \quad si \; n < 0 \end{cases}$$

Ex2 : Fonctions de transfert des filtres numériques et TZ

Soient les fonctions de transfert suivantes :

$$H_1(z) = \frac{0.5}{1 - 0.5z^{-1}} = \frac{0.5z}{z - 0.5}$$

$$H_2(z) = \frac{0.36}{1 + 0.64z^{-2}} = \frac{0.36z^2}{z^2 + 0.64}$$

$$H_3(z) = 0.25 - 0.5z^{-1} + 0.25z^{-2} = \frac{0.25z^2 - 0.5z + 0.25}{z^2}$$

Pour chacun de ces trois filtres :

a. Déterminer les pôles du système. En déduire la stabilité du système.
b. Déterminer l'expression de la transformée de Fourier à temps discret $TFTD$ de la réponse impulsionnelle. On note T_e la période d'échantillonnage.
c. En utilisant le résultat de la partie b), tracer le spectre en amplitude (en dB) du filtre pour $f \in \left[-\frac{f_e}{2} + \frac{f_e}{2}\right]$ où $f_e = \frac{1}{T_e}$ (sur Matlab). On prend $f_e = T_e = 1$. En déduire le type de chaque filtre. En déduire la fréquence de coupure ou la bande passante (en Hz) en comparant le spectre à « la valeur maximal du spectre -3dB ».
d. Déterminer l'expression de la réponse impulsionnelle $h(n)$. En déduire le type de $h(n)$.
e. Donner l'équation aux différences permettant de calculer la sortie $y(n)$ pour une entrée $x(n)$ quelconque. Est-ce que ce filtre est causal ?

Traitement de Signal
TD6
Transformée en Z et filtres numériques

TP1 : Filtre numérique et Matlab

En utilisant **Matlab**, tracer le diagramme de Bode et la réponse impulsionnelle pour les trois filtres de l'exercice 3. On prend une fréquence d'échantillonnage Fe = 1 Hz.
On utilise les instructions suivantes de Matlab ***tf(), impulse(), bodeplot() .***

Vérifier les résultats théoriques ainsi que la réponse impulsionnelle calculée en exercice 3 pour chaque filtre.

Attention :

- *la fonction bodeplot() de Matlab affiche la fréquence en rad/s. Pour la transformer en Hz, il faut diviser par* 2π.
- *La commande doc (doc + nom de la commande Matlab) affiche l'aide au format HTML dans le "Help Broswer".*

TP2: Filtrage d'un bruit aigu dans un signal

Lors de l'enregistrement d'un signal audio, des bruits sonores indésirables à des fréquences variables sont parfois présents dans le signal numérique. L'objectif de ce travail est de supprimer un bruit très aigu présent dans un signal audio.

Nous avons un signal enregistré avec des bruits sinusoïdaux : une séquence de musique « Mozart_Bruit.wav »

A – Etude du signal « Mozart_Bruit.wav »

1. Charger le fichier : « **Mozart_Bruit.wav** » et écouter ce signal et déterminer sa période et sa fréquence d'échantillonnage.
2. Tracer les représentations **temporelles** et **fréquentielles** ce signal. Pour la représentation fréquentielle, on trace le spectre en dB ($20log_{10}(,)$) sur l'intervalle [0 f_e] où f_e est la fréquence d'échantillonnage.
3. Que peut-on conclure dans la forme du spectre de ce signal ?
4. Déterminer la valeur des fréquences indésirables entre $0\ et\ f_e$ qui génèrent le bruit.

B-Filtrage numérique en post-traitement (filtre idéal-non causal)

1. Quel est le type du **filtre idéal** qu'il faut utiliser pour supprimer ce bruit ?
2. Définir un vecteur $H(k)$ qui représente ce filtre dans le domaine fréquentiel
3. Appliquer ce filtre au signal original.
4. Générer, écouter et enregistrer sur votre PC le signal filtré

C– Filtrage numérique en temps réel (filtre IIR causal)

1. Quelles bandes ce filtre devra coupée ?

On utilise des filtres de type Butterworth de premier ordre pour éliminer chaque fréquence indésirable. Pour chaque fréquence indésirable, couper une bande de largeur 50 Hz centrée sur cette fréquence.

2. Utiliser la fonction ***butter()*** pour trouver les coefficients de chaque filtre
3. Filtrer le signal d'origine en cascade : ***filter().***
4. Reconditionnement : ramener le signal filtré entre -1 et 1. ***max() ; abs()***
5. Tracer sur les mêmes figures les représentations temporelles et fréquentielles du signal d'origine et du signal filtré. Conclure.
6. Ecouter et enregistrer sur le PC le signal filtré sous le nom "MozartFiltre.wav".
7. Déterminer la fonction de transfert du filtre équivalent : ***tf()***
8. Tracer le diagramme de bode du filtre équivalent : ***bodeplot()***
9. Tracer la réponse impulsionnelle du filtre équivalent. Conclure sur la stabilité de ce filtre ? ***impulse()***