ELEC-H-313 : Instrumentation

Laboratoire 5: Filtres numériques

Information préliminaire

Le laboratoire se déroule au sein d'un "LiveScript" MATLAB. Ce dernier contient du code qu'il faut compléter afin de répondre aux différents exercices.

Si vous n'êtes pas familier avec l'éditeur MATLAB et les commandes usuelles, il est conseillé de réaliser le tutoriel MATLAB Onramp avant le laboratoire.

Toolboxes requises

Cette séance nécessite les toolboxes suivantes:

- Signal Processing Toolbox
- Control System Toolbox

Objectifs du laboratoire

- Dimensionner un filtre numérique en utilisant la transformation bilinéaire à partir d'un prototype de filtre analogique.
- Réaliser le filtre sous Matlab.
- Observer les problèmes de quantification.

Rappels

Nous avons vu dans la dernière séance comment dimensionner un prototype de filtre analogique afin de respecter un gabarit. Il arrive également que l'on doive réaliser un filtre numérique (dans le code d'un microcontrôleur par exemple). Il existe plusieurs méthodes afin de dimensionner un filtre numérique. La méthode la plus utilisée est de dimensionner un filtre analogique et de le transposer en un filtre numérique via différentes méthodes. De cette manière, on peut réutiliser tous les résultats connus pour les filtres analogiques, par exemple les solutions de type Butterworth, Chebyshev, elliptique, ...

Deux méthodes classiques permettent de transformer un filtre analogique en filtre numérique :

- La transformation bilinéaire (bilinear transform)
- L'échantillonnage de la réponse impulsionnelle (impulse invariance)

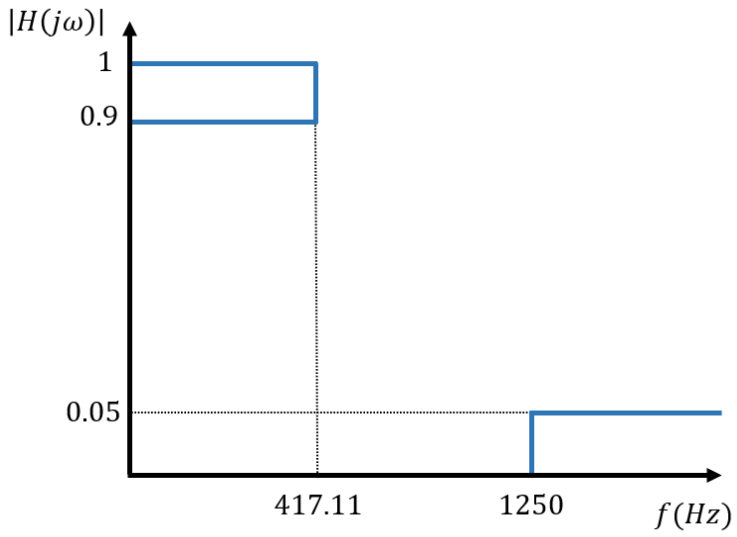
Bien que les deux méthodes soient utilisables et ont leurs avantages et inconvénients, nous verrons ici la transformation bilinéaire qui est la plus courramment utilisée car elle conserve la stabilité des filtres, empêche le repliement spectral et qu'elle peut s'appliquer à n'importe quel type de filtre.

L'annexe "*Transformation bilinéaire*" décrit la méthode.

```
clearvars %clear the workspace
clc
cd(fileparts(matlab.desktop.editor.getActiveFilename)); %Changes location to script directory
```

Exercice 1: Transformation bilinéaire

En utilisant la transformation bilinéaire, déterminez un filtre numérique (en z) respectant le gabarit ci-dessous.



1. Déterminez l'ordre et la pulsation de coupure du **filtre analogique** respectant ce gabarit ainsi que sa fonction de transfert $H(p)$. La fonction Matlab buttord réalise l'approximation de Butterworth **en respectant la bande d'arrêt exactement**. Par défaut, cette fonction travaille dans le domaine numérique (fréquences normalisées par rapport à la fréquence de Nyquist). Vérifiez la syntaxe requise pour travailler dans le domaine analogique. La fonction butter permet d'obtenir le numérateur et le dénominateur de la fonction de transfert du filtre analogique.

```
%% Transformation bilineaire
gainPassant = mag2db(0.9);
gainStop = mag2db(0.05);
% [n,wc] =   % buttord returns min order and cut-off w
fc = wc/(2*pi)
%compare the obtained fc with results from the previous lab and explain

% [numA,denA] =  ;%find the TF
LPa = tf(numA,denA)

figure('Name', 'Bode plots')
bode(LPa)
```

2. Sans la résoudre ou la simplifier, exprimez la fonction de transfert du filtre **numérique** équivalent (sur papier).

3. A l'aide de l'approximation bilinéaire (bilinear), déterminez la fonction $H(z)$ du filtre numérique équivalent, sachant que le système numérique possède une fréquence d'échantillonage de 10kHz.

```
% Fs =  ; %Sampling frequency
% [NumD,DenD]= ;
LPd=tf(NumD,DenD,1/Fs)
```

```
figure('Name', 'Bode plots')
bode(LPa)
hold on
bode(LPd)
legend('LPAnalog','LPDigital')
```

4. La fonction butter permet également d'obtenir directement le numérateur et le dénominateur de la fonction de transfert numérique. Il est également possible d'utiliser la fonction c2d (continuous to digital), en spécifiant la méthode "Tustin" (autre nom de la transformation bilinéaire), pour transformer une fonction de transfert en p en une fonction de transfert en z. Comparez les fonctions de transfert et les courbes de Bode obtenues.

```
[NumDbis,DenDbis] = butter(n,(wc/(2*pi))/(Fs/2),'low'); %this method works in normalized freque
LPdbis = tf(NumDbis,DenDbis,1/Fs)
LPdter = c2d(LPa,1/Fs,'tustin') %continuous to digital transformation, tustin = bilinear metho

figure('Name', 'Bode plots')
bode(LPa)
hold on
bode(LPd)
bode(LPdbis)
bode(LPdter)
legend('LPAnalog','LPDigital', 'LPDbis', 'LPDter')
```

5. Comparez les réponses en fréquences des filtres numériques et du filtre analogique et concluez sur la distorsion due à l'approximation bilinéaire.

6. Proposez une solution pour que le filtre numérique ait la même atténuation que le filtre analogique pour $f = 3.2\text{kHz}$. Implémentez cette solution via les options (c2dOptions) de la fonction c2d.

```
% options = c2dOptions( ); %sets specific options for c2d
LPDquad = c2d(LPa,1/Fs,options) %using the option set above to implement the solution
bode(LPDquad)
legend('LPAnalog','LPDigital', 'LPDbis', 'LPDter', 'LPDquad')
```