

# Filtrage Numérique

## SEATECH parcours IRIS

### TP 1 : Mise en oeuvre de filtres RIF et affichages

## Remarques introductives

Matlab propose un outil de visualisation des filtres numériques (`fvtool`), mais les conventions de tracé ne correspondent pas aux standards de la communauté, et pas non-plus aux conventions du cours.

Dans ce cours, on impose que les visualisations soient reprogrammées. Toutefois, cela ne concernent que la partie graphique, les calculs sous-jacents faisant toujours appels à des fonctions Matlab.

Au début du TP, on donne dans le sujet les figures que vous devez obtenir. Toutefois, lorsque les signaux comportent des composantes aléatoires, le rendu peut varier.

## 1 Filtre moyennneur

### 1.1 Génération et affichage du signal d'entrée

On définit un signal numérique par l'équation suivante :

$$x[n] = \sin(2\pi\nu n) + \sigma \mathcal{N}[n], \quad n \in [0 \cdots N - 1]$$

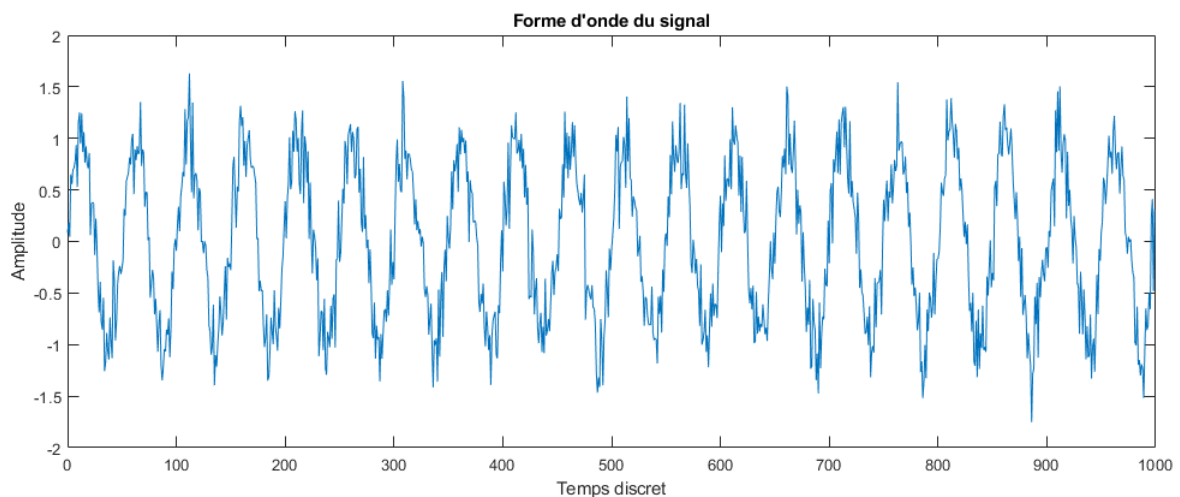
$$N = 1000$$

$$\nu = 0.02$$

$$\sigma = 0.25$$

$\mathcal{N}[n]$  désigne un bruit blanc gaussien normé centré (fonction `randn`).

Générer le vecteur signal, son vecteur support temporel, puis afficher la forme d'onde en tracé "semi-continu" (fonction `plot`). Ne pas oublier les légendes.



## 1.2 Génération et affichage du spectre du signal d'entrée

Pour calculer le spectre TFTD du signal  $x[n]$ , qu'on notera  $X(\nu)$ , on ne **doit pas** utiliser la FFT, car FFT et TFTD sont deux transformées distinctes qui ne coïncident que sous certaines conditions très particulières.

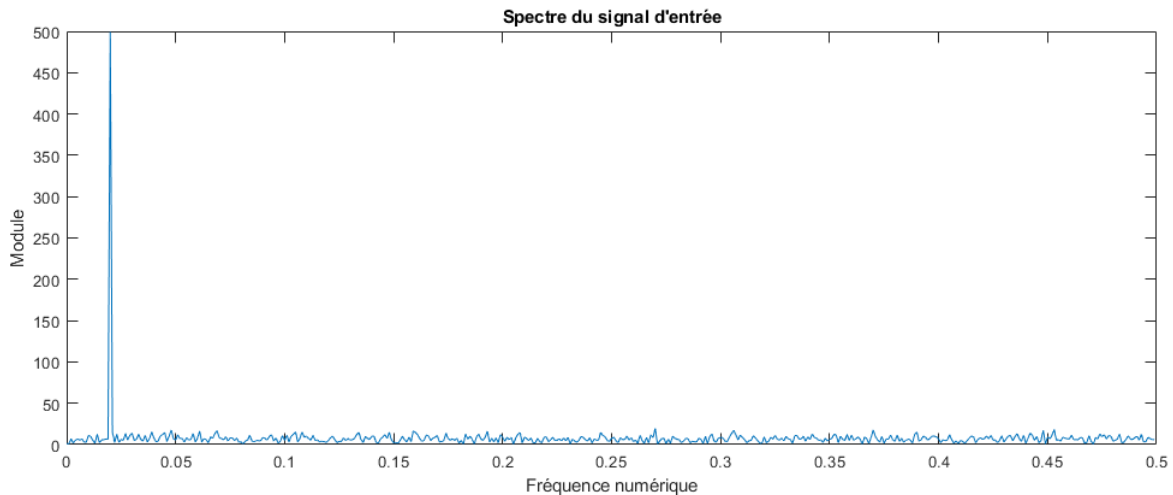
Sous Matlab, on peut calculer facilement un spectre TFTD et le vecteur support associé avec la fonction `freqz`, sur laquelle on reviendra plus en détails dans la suite. Pour calculer le spectre TFTD d'un vecteur signal `x`, la syntaxe est la suivante :

```
[X,w] = freqz(x,1,K)
```

`X` correspond au vecteur TFTD (**complexe**), `w` est le vecteur support fréquentiel et `K` est le nombre de points en fréquence. Par convention, il est supposé que le vecteur support temporel de `x` commence à  $n = 0$ .

Le vecteur support fréquentiel `w` est exprimé en pulsation (radians par secondes, entre 0 et  $\pi$ ). Pour le convertir en fréquence numérique (entre 0 et 0.5), on divise par  $2\pi$ .

Calculer et afficher le module du spectre TFTD du signal  $x[n]$  avec 500 points en fréquence, toujours en tracé "semi-continu". Interpréter.



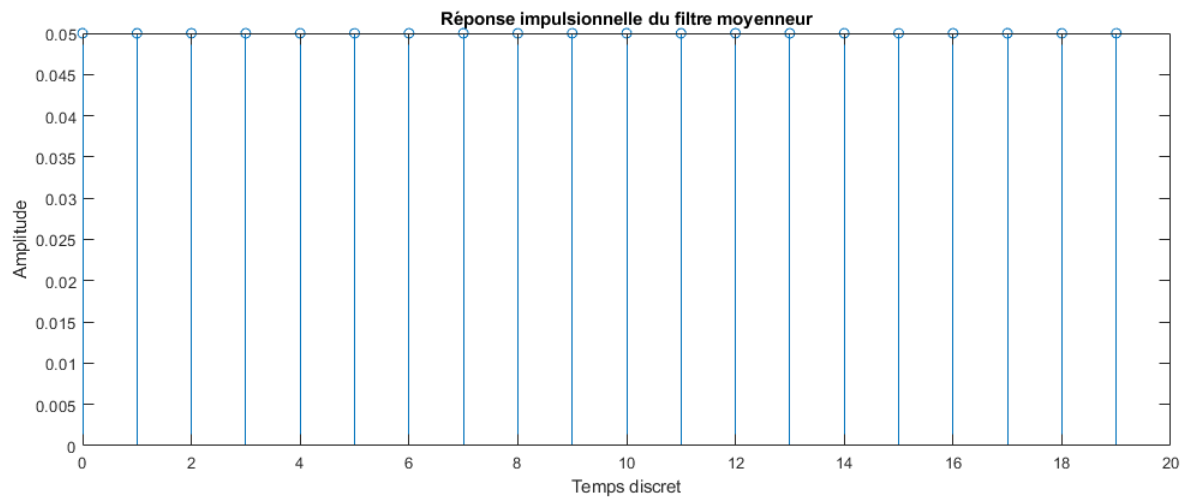
## 1.3 Génération et application du filtre numérique

On considère un filtre RIF dit "moyenneur" défini par sa réponse impulsionnelle :

$$h[n] = 1/P \quad n \in [0 \cdots P-1]$$
$$h[n] = 0 \quad \text{ailleurs}$$

$$P = 20$$

Générer le vecteur RI, son vecteur support temporel, puis afficher la RI en tracé "discret" (fonction `stem`).

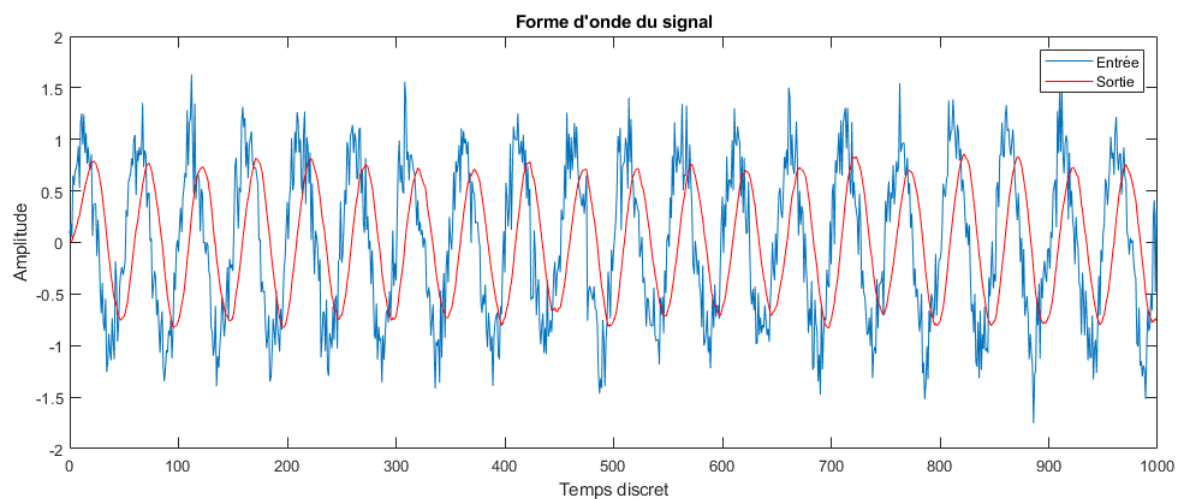


Pour appliquer la relation entrée-sortie du filtre, en l'occurrence la convolution entre  $x$  et  $h$ , on utilise la fonction `filter`, sur laquelle on reviendra plus en détails dans la suite. Dans le cas d'un filtre RIF, la syntaxe est la suivante :

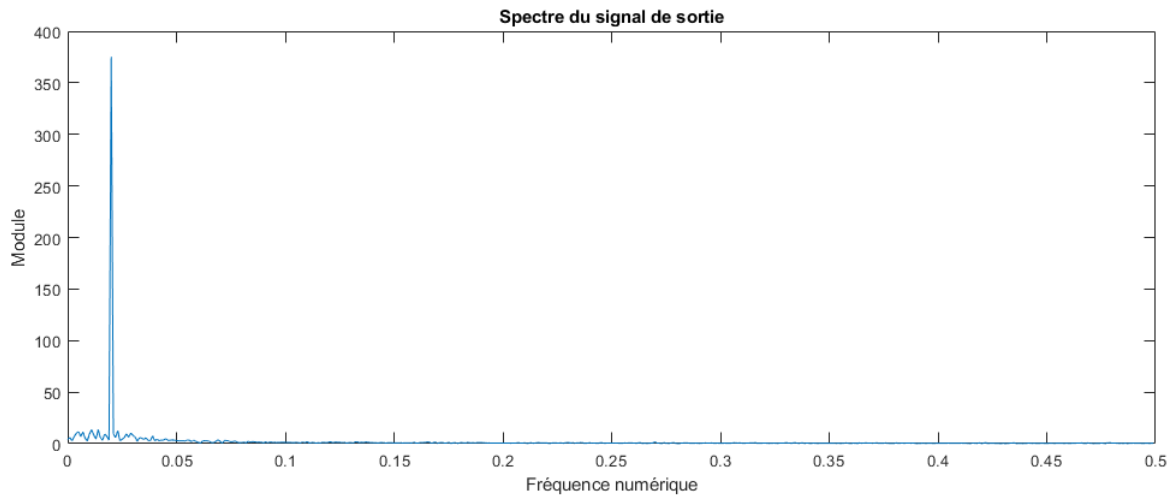
```
y = filter(h,1,x)
```

Par convention, le vecteur  $y$  a la même dimension que le vecteur  $x$ , et tous les vecteurs supports temporels commencent à  $n = 0$ , ce qui implique qu'on suppose le filtre causal.

Générer le signal de sortie et son vecteur support temporel. L'afficher en le superposant au tracé du signal d'entrée. Décrire l'effet du filtre dans le domaine temporel.



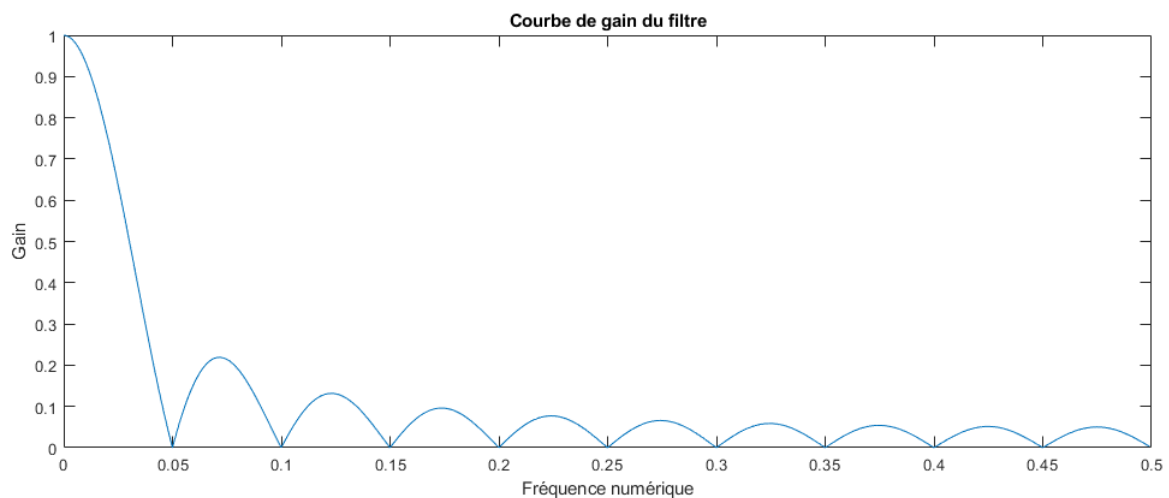
Calculer et afficher le module du spectre TFTD du signal  $y[n]$  avec 500 points en fréquence, toujours en tracé "semi-continu". Décrire l'effet du filtre dans le domaine fréquentiel.



## 1.4 Réponse en fréquence du filtre numérique

Par définition, la réponse en fréquence est la TFTD de la réponse impulsionnelle. Pour la calculer, on utilise donc la même méthode que précédemment avec la fonction `freqz`.

Générer la courbe de gain (le module de la réponse en fréquence) du filtre moyennneur, et l'afficher en tracé "semi-continu". Expliquer l'effet du filtre dans le domaine fréquentiel à partir de la courbe de gain.



## 2 Filtre passe-bas symétrique

Dans cette deuxième partie, on reprend les méthodes décrites précédemment en les appliquant à un autre signal d'entrée et un nouveau filtre. On ne donne pas les figures à obtenir.

### 2.1 Génération d'un signal d'entrée créneaux

On utilise maintenant comme signal d'entrée un signal créneaux. Chaque période est constituée de 32 échantillons de valeur +1 puis 32 échantillons de valeur -1. Le signal total comprend 20 périodes (soit 1280 échantillons au total).

Générer le signal d'entrée, son support temporel, le spectre du signal et son support temporel puis les afficher. Décrire le spectre du signal en terme d'analyse harmonique.

## 2.2 Importation du filtre

La réponse impulsionnelle du filtre est fournie dans l'archive `Filtre_RIF_TP1.mat`. Pour importer la RI, utiliser la fonction `load`.

Afficher la RI et la courbe de gain du filtre. Quelle est la fréquence de coupure du filtre ?

## 2.3 Application du filtre

Calculer le signal de sortie ainsi que son spectre et les afficher. Décrire l'effet du filtre sur le signal et les justifier par la courbe de gain.