

Traitement du signal temps-réel

SEATECH IRIS 2ème année

TP 1 : Mise en oeuvre de filtres numériques

1 Squelette du traitement et affichages

En utilisant les éléments décits en cours, mettre en place un programme de traitement qui recopie le signal d'entrée sur le signal de sortie (traitement transparent). Inclure un affichage de forme d'onde et un affichage spectral. Les deux affichages doivent superposer le signal d'entrée et le signal de sortie sur les même figures.

On utilise les objets suivants :

- `dsp.AudioFileReader` avec des blocs de 512 échantillons
- `dsp.AudioFileWriter`
- `timescope`
- `dsp.SpectrumAnalyzer`

Tester le programme avec le fichier audio `Suzanne_Vega_44_mono.wav` fourni. Vérifier sur les affichages et par écoute que le contenu du fichier de sortie est indentique au contenu du fichier d'entrée.

Convertir le programme pour fonctionner avec des E/S sur la carte son. Mettre en place une alerte d'overrun en ligne de commande (fonction Malab `disp`). Fixer la durée totale d'exécution à une valeur raisonnable (typiquement 10 s). On utilise les objets suivants :

- `audioDeviceReader` remplace `dsp.AudioFileReader`
- `audioDeviceWriter` remplace `dsp.AudioFileWriter`

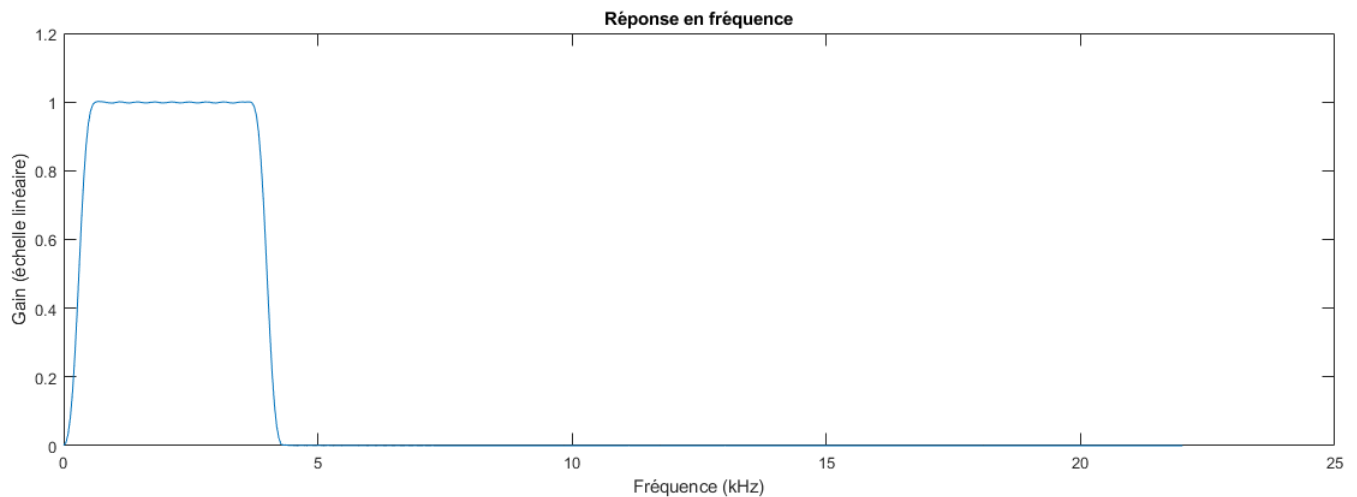
Tester le programme sur le même fichier audio en utilisant un PC comme lecteur, et un autre comme machine de traitement. Vérifier sur les affichages et par écoute que le contenu du fichier de sortie est indentique au contenu du fichier d'entrée.

2 Implémentation d'un filtre RIF passe-bande

En utilisant comme base les programmes précédents, implémenter un traitement qui simule le passage dans un canal téléphonique. Pour cela, on implémente un filtre RIF basse-bande :

- Fréquence de coupure basse : 300 Hz
- Fréquence de coupure haute : 4000 Hz
- Ordre 256

Calculer les fréquences de coupures numériques du filtre, synthétiser sa RI en utilisant `fir1` et afficher sa réponse en fréquence en utilisant `freqz`. On doit obtenir ceci :



Insérer le filtre dans la boucle de traitement (E/S sur des fichiers puis sur la carte son) en utilisant `filter`. Vérifier sur l’affichage spectral que le filtre se comporte comme attendu.

3 Implémentation d’un filtre RII réjecteur

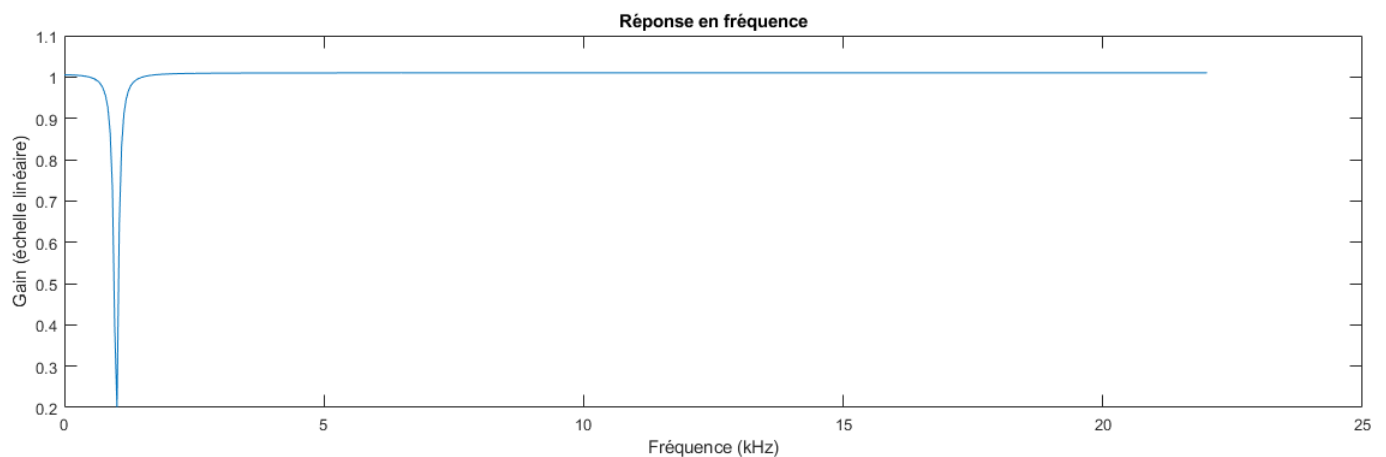
L’objectif est d’implémenter un filtre RII permettant de supprimer une certaine fréquence qui pourrait perturber le signal. Ce type de traitement est utile pour supprimer les remontées du courant secteur (50 Hz et ses harmoniques : 100 Hz, 150 Hz, ...) ou pour réaliser des stabilisateurs de systèmes bouclés (par exemple anti-larsen).

Une cellule réjectrice RII est un filtre ARMA d’ordre 2 défini par ses pôles et ses zéros :

- Zéros : $z = e^{\pm j2\pi\nu_r}$
- Pôles : $p = \rho e^{\pm j2\pi\nu_r}$

où ν_r représente la fréquence de réjection numérique. Le paramètre ρ est supposé proche de 1 par valeur inférieure.

En utilisant les méthodes vues dans le cours de filtrage numérique, calculer sous Matlab les polynômes numérateur (B) et dénominateur (A) de la fonction de transfert du filtre, pour une fréquence de réjection de 1000 Hz et pour $\rho = 0.99$. Afficher la réponse en fréquence en utilisant `freqz`. On doit obtenir ceci :



Tester le programme avec le fichier audio `Suzanne_Vega_1000Hz.wav` fourni (d'abord avec E/S sur des fichiers, puis sur la carte son). Vérifier que le traitement fonctionne correctement.

Lorsque la fréquence de la sinusoïde à supprimer n'est pas rigoureusement égale à la valeur théorique, le traitement ne fonctionne pas bien. Vérifier cela avec le fichier audio `Suzanne_Vega_1000Hz_variable.wav`.

Proposer et implémenter une solution qui améliore les performances du filtre, en associant plusieurs cellules réjectrices en série. Pour pré-calculer le filtre ARMA résultant, il sera utile d'utiliser la fonction `conv` qui implémente, entre autres, le produit de deux polynômes (voir aide de Matlab pour les détails).