TP 2 : Identification, débruitage et analyse temps-fréquence

Il est indispensable d'avoir préparé le TP avant la séance.
Les listings des programmes doivent être joints au rapport.
Tous les résultats et courbes doivent être commentés.

1 Présentation du TP

Ce TP est constitué de deux parties :

— Dans la première partie, on cherche à caractériser un filtre inconnu en estimant sa réponse impulsionnelle $h[n]$ ou sa réponse en fréquence $\widehat{h}(f)$ à partir de la sortie $y[n]$ du filtre, éventuellement perturbée, mesurée pour une entrée $x[n]$. On étudiera pour cela des méthodes expérimentales s'appuyant sur l'estimation de la corrélation des signaux. On verra ensuite l'utilisation de ce type de méthodes pour le débruitage des signaux à partir d'une référence de bruit.

— Dans la seconde partie, on illustrera le problème de l'analyse temps fréquence des signaux, en particulier par la transformée de Fourier à fenêtre glissante. Pour cela, on commencera par bien comprendre la signification des paramètres de cette représentation sur des signaux simulés et bien appréhender les problèmes de localisation en temps et en fréquence. Ensuite, on tentera d'analyser des signaux réels.

Les questions dont le numéro est suivi du signe "†" sont des questions théoriques, ne nécessitant pas l'utilisation de Matlab, qui doivent impérativement être résolues avant la séance de TP.

Première partie

2 Identification de la réponse impulsionnelle d'un filtre

On considère un signal $e[n]$ mis en entrée d'un filtre de réponse impulsionnelle $h[n]$. On dispose de l'entrée et d'une sortie perturbée $s[n]$, la perturbation étant notée $b[n]$ (*cf* Figure 1).

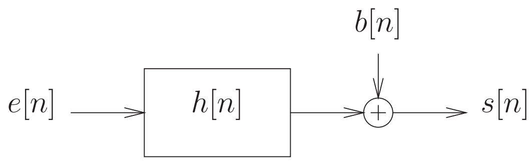

FIGURE 1 – Relation entrée sortie d'un filtre

2.1 Étude du filtre

Dans toute cette partie, on considérera le filtre de fonction de transfert

$$H(z) = \frac{1 - 0,5z^{-1} + z^{-2}}{1 - 0,729z^{-3}}.$$

1. En utilisant la fonction Matlab `freqz`, tracer la réponse en fréquence de ce filtre.
2. Tracer sa réponse impulsionnelle $h[n]$. **Dans la suite du TP on considérera que cette réponse impulsionnelle est finie, et qu'elle peut être estimée par un filtre RIF d'ordre** $P = 50$. Commentez cette approximation...
3. † Donner la relation liant l'intercorrélation sortie-entrée $R_{se}[m]$ à l'autocorrélation de l'entrée $R_e[m]$ et à l'intercorrélation perturbation-entrée $R_{be}[m]$. Que devient cette relation si $e[n]$ et $b[n]$ sont non-corrélés et que $b[n]$ est centré ?
4. † La relation de la question 3 peut s'écrire sous forme matricielle $\Gamma h = c$, avec Γ matrice carrée, en rangeant les coefficients de la réponse impulsionnelle dans un vecteur colonne. A quoi correspondent les éléments de la matrice Γ et du vecteur c ?
5. Dans une expérience, on a placé un signal $e[n]$ à l'entrée du système de Fig. 1 et on a mesuré la sortie $s[n]$. Les signaux $e[n]$ et $s[n]$ sont disponibles dans le fichier *sig.mat*. Construire la matrice Γ et le vecteur c en utilisant ces signaux et les corrélations estimées. Résoudre le système pour estimer la réponse impulsionnelle du filtre et sa réponse en fréquence. Tracer ces réponses et les comparer aux réponses théoriques. Conclusions...
6. † Comment s'écrit la relation de la question 3 dans le domaine fréquentiel (on notera S_{se}, S_e et S_{be} les densités spectrales et inter-spectrales utilisées) ?
7. † Comment peut-on estimer la réponse en fréquence du filtre grâce à cette dernière relation ? Quelles sont les conditions nécessaires pour que cette estimation se passe *sans problèmes* ?
8. Utiliser cette méthode pour estimer la réponse en fréquence puis la réponse impulsionnelle du filtre. Tracer ces réponses et les comparer aux réponses théoriques. Conclusions...

3 Débruitage d'un signal avec référence de bruit

On va exploiter une des méthodes précédentes pour identifier la réponse impulsionnelle d'un filtre dans une configuration de débruitage d'un signal avec référence de bruit telle que représentée Figure 2. On suppose disposer de deux capteurs : le premier mesure le signal bruité (Mesure) et le deuxième mesure la référence de bruit b[n].

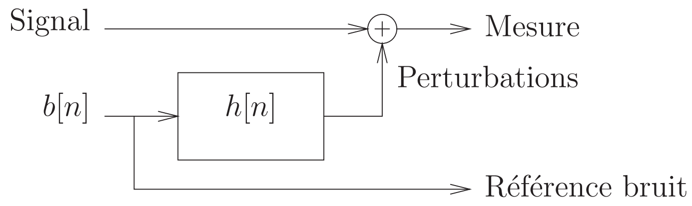

FIGURE 2 – Configuration pour le débruitage avec référence de bruit

1† Montrer que l'on peut bien exploiter la corrélation pour débruiter le signal.

Dans la suite, on applique la méthode ci-dessus dans deux applications différentes.

3.1 Application biomédicale

Le premier exemple est une application biomédicale qui concerne la mesure de l'électrocardiogramme d'un fœtus dans le ventre de sa mère. Un capteur, situé sur le ventre de la mère donne le signal électrique du cœur du fœtus (signal) additionné au signal électrique issu du cœur de la mère (perturbations). Un autre capteur situé près du cœur de la mère donne le signal électrique du cœur de la mère (référence de bruit). On modélise la relation entre la référence de bruit et les perturbations par un système linéaire de réponse impulsionnelle $h[n]$. Une fois estimée la réponse impulsionnelle $h[n]$ il est aisé de débruiter le signal à partir de la référence de bruit.

Choisissez parmi les méthodes précédentes exploitant la corrélation celle qui vous parait donner les meilleurs résultats. Les mesures `mes`, la référence de bruit `ref` et le signal `sig` sont disponibles dans le fichier `estim.mat`.

1. Estimer et tracer la réponse impulsionnelle du filtre.
2. A partir de la réponse impulsionnelle estimée, en déduire une estimation du signal débruité. Tracer et comparer les signaux estimé et original. Conclusions...

3.2 Application audio

Le deuxième exemple est une application audio. On suppose que le signal utile (signal) est la voix d'une personne qu'on veut enregistrer avec un microphone "A" situé près de cette personne. En pratique, cet enregistrement est pollué par une perturbation, résultat de la propagation d'une source de bruit lointain et fort jusqu'au microphone "A". En modélisant la propagation de ce bruit par un effet de filtrage, nous pouvons utiliser la technique ci-dessus en plaçant un deuxième microphone "B" près de la source du bruit (référence de bruit). On retrouve donc le modèle de la figure ci-dessus[1].

1. On suppose ici que le signal utile est faible devant le bruit. Si ce n'est pas le cas, on doit également tenir compte de la propagation du signal utile jusqu'au microphone B, ce qui nécessite l'utilisation des techniques de débruitage plus sophistiquées.

Le fichier z2.wav contient un signal sonore enregistré par le microphone “A”. Le fichier b2.wav contient une mesure de ce bruit par le microphone “B”.

1. Ecouter ces deux signaux.
2. En utilisant la méthode expliquée ci-dessus, estimer la réponse impulsionnelle du filtre de propagation et en déduire le signal utile.
3. Ecouter le signal utile.

Deuxième partie

4 Analyse temps-fréquence

L’analyse temps-fréquence a été abordée en cours au travers de la transformée de Fourier à fenêtre glissante (ou TF à court terme – *short term Fourier Transform* en anglais). L’idée de base de cette représentations est de décomposer le signal sur une famille de sinusoïdes fenêtrées.
Pour une fenêtre $g(t)$ donnée, les coefficients d’une telle décomposition se calculent par :

$$C_x(t,f) = \int_{-\infty}^{+\infty} x(\tau) g_{t,f}^*(\tau) \mathrm{d}\tau \text{ avec } g_{t,f}(\tau) = g(\tau - t) e^{2j\pi f \tau}$$

Il est alors immédiat de montrer que :

$$C_x(t,f) = \int_{-\infty}^{+\infty} x(\tau) g^*(\tau - t) e^{-2j\pi f \tau} \mathrm{d}\tau = \mathrm{TF}\{x(\tau) g^*(\tau - t)\}$$

d’où le nom de Transformée de Fourier à fenêtre glissante ($g^*(\tau - t)$ correspondant à une fenêtre centrée en t). En pratique, il est procédé à une discrétisation de cette relation en temps et en fréquence et le calcul est effectué par Transformée de Fourier Rapide (FFT).
Le spectrogramme d’un signal est défini comme le module au carré de la TF à court terme.
Dans une telle représentation, il se pose la question de localisation en temps et en fréquence :

1.† Pour une fenêtre de support temporel ΔT et de support fréquentiel ΔF, donnez le domaine du plan temps-fréquence influencé par un événement intervenant à un instant t_0.

2.† Pour une telle fenêtre, donnez le domaine du plan temps-fréquence influencé par un événement de fréquence f_0.

3.† Sur quels paramètres de la fenêtre faut-il jouer pour obtenir une meilleure localisation en temps ? en fréquence ? Peut-on avoir simultanément une bonne localisation en temps et en fréquence ?

4.1 Compréhension sur des signaux simulés

Il va s'agir principalement ici de bien comprendre l'utilisation des programmes calculant le spectrogramme (voir l'annexe sur l'utilisation de la fonction `spectrogram` de Matlab). Pour tous les signaux générés, on prendra une fréquence d'échantillonnage de $F_e = 8$ kHz et un nombre d'échantillons de $N = 1024$.

1. Tracer le spectrogramme d'une impulsion unité (Kronecker) pour différents types de fenêtres. Conclure sur la localisation temporelle de la TF à court terme suivant la fenêtre…
2. Tracer le spectrogramme d'une sinusoïde (faire varier la fréquence) pour différents types de fenêtres. Conclure sur la localisation fréquentielle de la TF à court terme suivant la fenêtre et la fréquence…
3. Tracer le spectrogramme de signaux comportant à la fois des événements ponctuels et des fréquences pures. Conclure sur le compromis à régler entre localisation temporelle et fréquentielle…
4. Poursuivre par l'analyse de signaux synthétiques de votre choix (chirp, sinusoïdes par morceaux, signaux bruités, etc.) Commentez vos résultats…

4.2 Analyse de signaux réels

Vous disposez d'un ensemble de fichiers de signaux qu'il vous faut analyser. Pour cela, vous pouvez tracer leur représentation temporelle, leur représentation fréquentielle, mais également leur spectrogrammes. Commentez vos résultats d'analyse sur quelques-uns de ces signaux…

5 Annexe

De nombreuses fonctions Matlab vous seront utiles dans ce TP.

Dans la première partie, les fonctions `randn`, `freqz`, `filter`, `xcorr` et `toeplitz` pourront être utilisées. Il est prudent de faire appel à l'aide en ligne avant de les utiliser…

Dans la deuxième partie, l'analyse temps-fréquence sera effectuée grâce à la fonction `spectrogram` de Matlab décrite ci-dessous.

5.1 Annexe : Fonction Matlab `spectrogram`

```
spectrogram Spectrogram using a Short-Time Fourier Transform (STFT).
   S = spectrogram(X) returns the short-time Fourier transform of the
   signal specified by vector X in the matrix S. By default, X is divided
   into eight segments with 50\% overlap, and each segment is windowed with
   a Hamming window. The number of frequency points used to calculate the
   discrete Fourier transforms is equal to the larger of 256 or the next
   power of two greater than the segment length.
```

```
    S = spectrogram(X,WINDOW), when WINDOW is a vector, divides X into
    segments of the same length as WINDOW, and then windows each segment
    with the vector specified in WINDOW.  If WINDOW is an integer, the
    function divides X into segments of length equal to that integer value
    and windows each segment with a Hamming window.  If WINDOW is not
    specified, the default is used.

    S = spectrogram(X,WINDOW,NOVERLAP) specifies NOVERLAP samples of
    overlap between adjoining segments. NOVERLAP must be an integer smaller
    than WINDOW if WINDOW is an integer.  NOVERLAP must be an integer
    smaller than the length of WINDOW if WINDOW is a vector.  If NOVERLAP
    is not specified, the default value is used to obtain a 50\% overlap.

    S = spectrogram(X,WINDOW,NOVERLAP,NFFT) specifies the number of
    frequency points used to calculate the discrete Fourier transforms.
    If NFFT is not specified, the default NFFT is used.

    S = spectrogram(X,WINDOW,NOVERLAP,NFFT,Fs) specifies the sample rate,
    Fs, in Hz. If Fs is specified as empty, it defaults to 1 Hz. If it is
    not specified, normalized frequency is used.

spectrogram(...) with no output arguments plots the PSD estimate for
    each segment on a surface in the current figure.
```