Initiation au traitement du signal et applications
Séance 6

École des Mines de Nancy

Frédéric Sur
http://www.loria.fr/~sur/enseignement/signal/
sur@loria.fr

Consigne : Il vous est demandé de **rédiger les travaux pratiques**. Ouvrez en parallèle un document Word dans lequel vous allez coller les différents résultats obtenus, les commandes qui vous ont permis de les obtenir, et les réflexions que cela vous inspire. L'évaluation de ce cours sera basée sur ces comptes rendus.

1 Transformée de Fourier discrète à fenêtre (STFT) avec Matlab

1.1 Les fonctions fournies

Pour calculer les transformées de Fourier à fenêtre, on utilisera les fonctions `dgt.m` et `idgt.m` faisant partie de la *Linear Time-Frequency Analysis Toolbox*[1]. Commencez par copier ces deux fonctions dans votre répertoire de travail, ainsi que les fonctions `candual.m` et `pgauss.m`. Elles appellent des fonctions annexes qui sont dans l'archive `comp.zip`. Décompressez cette archive et copiez le répertoire correspondant dans votre répertoire de travail.

Puis réglez le chemin de Matlab en entrant la commande :

```
>> path(path,'U:\chemin\vers\votre\repertoire\comp')
```

Vous pouvez aussi utiliser l'onglet `File -> Set Path`.

1.2 Syntaxe de `dgt.m` et `idgt.m`

Rappel : l'aide peut être obtenue par `help nom_fonction`.

La formule pour la STFT discrète figurant dans le cours permet de calculer la STFT de taille $N \times N$ (N positions temporelles de la fenêtre et N fréquences) pour un signal de longueur N. En fait on peut se contenter de ne pas calculer la STFT pour toutes les positions et toutes les fréquences mais en calculer un sous-échantillonnage. Ceci accélère les calculs, et si on garde suffisamment de coefficients la représentation reste assez précise.

La fonction `dgt.m` calcule la STFT ou *Discrete Gabor Transform* d'un signal. Elle peut prendre quatre arguments qui sont dans l'ordre :

1. le signal à analyser ;
2. la fenêtre ;
3. le pas entre les positions temporelles calculées ;
4. le nombre de fréquences calculées.

[1] `http://sourceforge.net/projects/ltfat/`

Le signal à analyser sera un vecteur Matlab. Les pas en espace et en fréquence seront des diviseurs de sa longueur N.

On ne considérera ici que des fenêtres gaussiennes. La fonction `pgauss.m` permet de calculer les coefficients de cette fenêtre. Elle peut prendre deux arguments :

1. la longueur du signal à analyser
2. le rapport entre le support temporel et le support fréquentiel de la fenêtre.

Plus ce rapport est faible (proche de 0), plus la fenêtre est localisée en temps, plus il est grand (>5 par exemple), plus la fenêtre est large.

Exemple : on cherche à analyser une signal `Y` de longueur 8000.

```
>> G=pgauss(length(Y),0.2);  % essayez d'autres valeurs que 0.2
>> plot(fftshift(G))  % pour visualiser la largeur de G
>> dgtY=dgt(Y,G,10,1000);  % valide car 10 et 1000 divisent 8000
>> imagesc(abs(dgtY))  % permet de visualiser le module du spectre
```

Vous pouvez également visualiser le logarithme du module du spectre.

La fonction `idgt.m` fait la transformation inverse. Ses trois arguments sont :

1. le spectre à inverser ;
2. la fenêtre conjuguée à la fenêtre d'analyse ;
3. le pas entre les positions temporelles calculées.

La fenêtre conjuguée s'obtient par la fonction `candual.m`. Le pas entre positions temporelles est le même que celui utilisé par `dgt.m`.

Exemple :

```
>> Gdual=candual(G,10,1000);
>> idgtW=dgt(Y,Gdual,10);
```

2 Mise en évidence du principe d'incertitude

On considère le signal discret suivant :

```
>> X=[.005:.005:20];
>> Y=[cos(20*pi*X(1:500)), cos(50*pi*X(501:800)), ...
     cos(100*pi*X(801:2500)), cos(150*pi*X(2501:4000))];
```

Observez le spectre pour différentes tailles de la fenêtre d'analyse :

```
>> G=pgauss(4000,a);
>> plot(fftshift(G));
>> dgtY=dgt(Y,G,5,500);
>> figure, imagesc(abs(dgtY));
```

où `a` prend les valeurs suivantes : 0.05, 0.5, 2, 5, 10.

Comme chaque colonne correspond à la transformée de Fourier du signal sur une fenêtre, on observe la symétrie habituelle. On peut se contenter de visualiser le spectre sur la moitié des lignes :

```
>> adgtY=abs(dgt(Y,G,5,500));
>> figure, imagesc(adgtY(1:size(adgtY,1)/2,1:size(adgtY,2)));
```

Comment se traduit ici le principe d'incertitude ?

3 Analyse d'un chirp

Un *chirp* (gazouillis, pépiement) est un signal modulé en fréquence, de la forme $f(t) = a(t)\sin(\phi(t))$, avec a et ϕ des fonctions réelles.

On considère le chirp suivant :

```
>> X=[0.0001:0.0001:10];
>> Y=sin(3.^X);
>> plot(X,Y);
```

Observez le résultat de l'analyse par STFT et expliquez-le. En particulier, expliquez l'évolution de l'épaisseur de la « courbe » obtenue. Quel est le phénomène mis en évidence dans la partie « droite » du spectre ?

Quel est le spectre obtenu par la transformée de Fourier discrète sur tout le signal ? Expliquez en quoi la STFT est dans ce cas plus adaptée que l'analyse de Fourier « classique ».

4 Modification de tempo sans changement de ton

On travaillera sur l'échantillon sonore : `foryoublue.wav`

On veut ralentir cet échantillon de manière à doubler sa durée.

Une manière naïve de faire est diviser par deux la fréquence d'échantillonnage passée à la fonction `wavplay` : on ralentit ainsi sa « lecture ». Que remarquez-vous ? Savez-vous l'expliquer ?

Une autre manière de faire est de procéder par duplication sur le spectrogramme.

On commence par extraire le spectrogramme (en complétant la longueur du signal `W` à 500000 dans `dgt`, pour avoir des diviseurs simples : c'est le rôle du cinquième argument) :

```
>> [W,Fs,nbits]=wavread('foryoublue.wav');
>> G=pgauss(500000,1);
>> dgtW=dgt(W,G,500,1000,500000);  % par exemple (visualisez dgtW)
```

et on crée un nouveau spectrogramme `dgtW2` où on copie dans les colonnes paires et impaires le spectrogramme.

```
>> size(dgtW)
>> dgtW2=zeros(1000,2*1000);
>> dgtW2(1:1000,1:2:2*1000)=dgtW;
>> dgtW2(1:1000,2:2:2*1000)=dgtW;
```

On procède ensuite à la transformée à fenêtre inverse pour récupérer un signal de durée deux fois plus grande que le signal original :

```
>> G=pgauss(1000000,1);
>> Gdual=candual(G,500,1000);
>> idgtW2=idgt(dgtW2,Gdual,500);
```

Comparez à la méthode précédente.

Proposez une démarche équivalente pour augmenter la vitesse, et implantez la.

Rappel : les fonctions Matlab pour le traitement des fichiers `wav` sont décrites dans le TP1.

5 Débruitage

5.1 Signal synthétique

Générez un signal sinusoïdal de fréquence 10 Hz, d'amplitude 1, sur une seconde, avec un pas de 1 ms.
Ajoutez-lui un bruit blanc gaussien d'écart-type 0.5 (fonction `randn`).
Calculez le spectre du signal bruité, et annulez tous les coefficients dont le module est en dessous de $S = 10\%$ (par exemple) du module maximal.
Reconstruisez le signal correspondant à ce spectre, et comparez avec le signal initial.
Discutez de l'influence de l'amplitude du bruit, et du seuil S.

5.2 Signal sonore

On considère le signal bruité : `foryoubluenoise.wav`
Commencez par visualiser le spectrogramme.
Vous allez tester les deux méthodes (assez naïves) de débruitage vues en cours.

1. *Filtrage linéaire "brutal"*. Mettez à 0 les coefficients de la STFT au dessus d'une certaine fréquence. Faites différents essais selon la valeur de la fréquence limite.
2. *Filtrage non linéaire* (seuillage dur). Mettez à 0 les coefficients dont le module est en dessous d'une certaine valeur. Faites différents essais.

Que constatez-vous ?
Quel phénomène finit par devenir audible quand trop de coefficients sont mis à zéro ?

Questions subsidiaires, pour les plus rapides...
Remplacez la fenêtre rectangulaire utilisée dans le filtrage linéaire par une fenêtre Gaussienne par exemple. (Quel est l'intérêt ?)
Comment s'adapte le filtre de Wiener (séance 3) dans le cas du débruitage ? Faites des essais.
Le seuillage dur (*hard thresholding*) consiste à mettre à 0 les coefficients de module inférieur à un seuil S. Ceci crée une discontinuité dans le spectre. Le seuillage doux (*soft thresholding*) consiste à mettre à 0 les coefficients de module inférieur à S et les autres à : signe$(x)(|x| - S)$. Testez également le seuillage doux.
Ajoutez un bruit blanc gaussien (fonction `randn`) à un échantillon musical de votre choix, et tentez de le débruiter.

6 Anecdote sur le spectrogramme

Après les passages de chansons à écouter à l'envers pour entendre des messages cachés, encore plus tordu :

`http://www.bastwood.com/aphex.php`