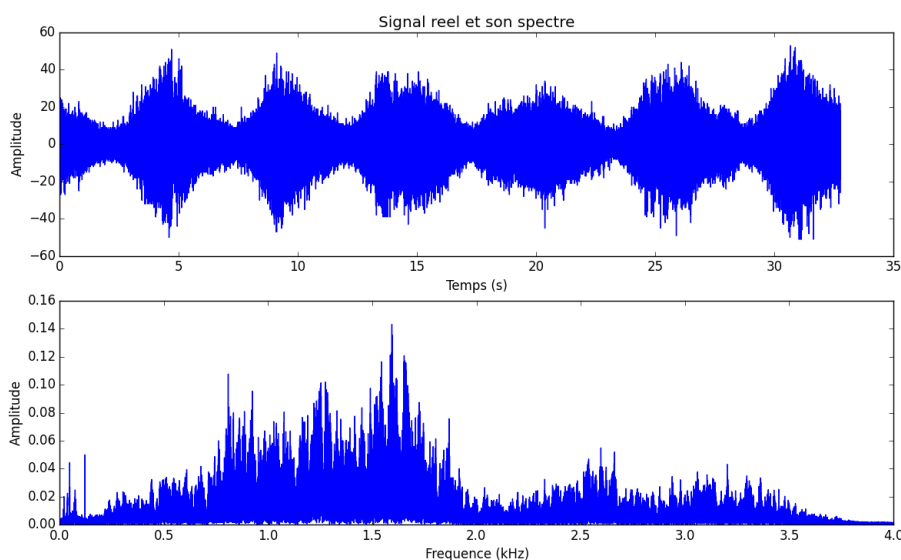


## Informatique S6 Projet: traitement du signal

Ce projet se place dans le contexte du traitement du signal sonore et de l'analyse de Fourier. On considère un signal sonore  $x(t)$ . Le projet va consister à écrire les procédures qui permettent d'échantillonner des signaux, les manipuler et appliquer des traitements tel que les filtres. On devra aussi discuter différentes propriétés à l'aide d'exemple.

On fournira le code ainsi qu'un petit rapport avec des exemples de résultats et l'analyse rapide des résultats ainsi que la description du code.



## Signal discret

A partir du cours numéro 3.

un signal discret dit finis peut être interprété comme un tableau à valeur dans les complexes

$$x(k), \quad k \in \{0, n-1\}$$

La première partie du projet consistera à écrire une classe détaillée de type signal discret : avec toutes les méthodes d'accès/modifications, tous les constructeurs et destructeurs, opérateurs dont on pourrait avoir besoin. On ajoutera à cela un certain nombre d'opérations :

- L'addition, la multiplication, la soustraction, le conjugué, le produit scalaire et l'énergie (on peut trouver cela facilement dans un cours de traitement du signal)

- La transformée de Fourier, la convolution circulaire et la transformée de Fourier inverse (essentiel)
- La translation :

$$t(k_0) = x(k - k_0), \quad k \in \{0, n - 1\}$$

- La modulation :

$$m(k_0) = e^{ik_0} x(k), \quad k \in \{0, n - 1\}$$

On utilisera la classe complexe fournie. La classe doit fournir des tests unitaires du maximum de méthodes et opérateurs. Des fonctions pour sortir les données et les tracer ensuite seront fournies.

## Signal continu

**A partir** du cours numéro 3.

On construira ensuite une classe de signal continue qui contiendra au moins :

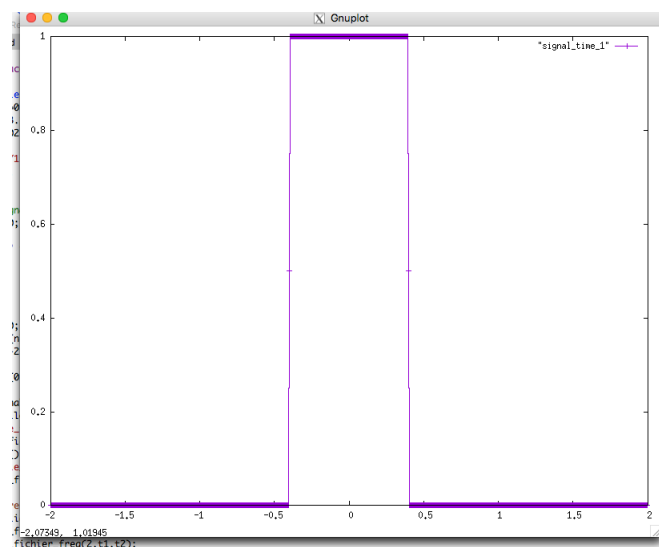
- le temps de début et de fin,
- un paramètre de variance du bruit,
- un paramètre de fenêtrage du signal (noté "s"),
- un pointeur vers une fonction  $f(t)$  qui sera la base du signal,
- une fonction échantillonnage dépendant de  $n$  qui donne un signal discret échantillonnant :

$$(f(t) + \epsilon b(t))g(t)$$

avec  $f(t)$  le signal de base  $b(t)$  un bruit Gaussien et la fonction fenêtrage (elle permet de mettre zéro en dehors d'une certaine fenêtre).  $g(t)$  est donné par

$$g(t) = \left( \frac{1}{2} \tanh(-r(t - s * T)) + \frac{1}{2} \right) * \left( \frac{1}{2} \tanh(r(t + s * T) + \frac{1}{2} \right)$$

avec  $T$  la durée du signal et  $r$  un paramètre (autour de 1000). Exemple pour  $s = 0.1$ .



Une fois cette classe et la précédente validées, discutez le problème du sampling/sous-échantillonnage. Donnez des exemples.

## Filtre

**A partir** du cours numéro 6-7.

Un filtre est un traitement sur un signal. En général on passe en fréquence (à l'aide de Fourier) pour obtenir un signal

$$\hat{x}(w), \quad w \in \{0, n-1\}$$

Les  $n/2$  premiers éléments représentent les fréquences de 0 à  $(n-1)/(2T)$ . Les  $n/2$  éléments suivants représentent les mêmes fréquences de  $(n-1)/(2T)$  à 0. Une fois la description fréquentielle obtenue on applique un filtre  $F(w)$  en multipliant  $\hat{x}(w)$  par  $F(w)$ . Par transformée inverse on récupère le signal filtré.

On construira ensuite une classe mère "filtre" continue qui contiendra au moins :

- le signal discret sur lequel on applique le filtre,
- le signal discret qui stocke le filtre,
- la durée du signal,
- une fonction qui applique le filtre,
- une fonction virtuelle qui construit le filtre.

On construira ensuite des classes dérivées pour chaque exemple de filtre avec leur fonctions, constructeurs propres et leurs implémentations de la fonction qui construit le filtre. On implémentera plusieurs exemples :

- filtre passe bas/passe haut idéal
- filtre passe bas/passe haut 1er ordre
- filtre passe bas/passe haut second ordre

On construira aussi des classes "filtre coupe bande 1er ordre" héritée des classes "filtre passe bas 1er ordre" et "filtre passe haut 1er ordre".

On comparera les différents filtres passe bas sur des exemples de débruitage d'un signal et on utilisera le filtre coupe bande pour donner un exemple où on retire une fréquence particulière.

## STL

**A partir** du cours numéro 11.

Réécrire une classe "signal discret" en remplaçant l'allocation dynamique par la classe "vector" ou "valarray" de la STL. Ecrire les classes filtres avec des templates permettant d'utiliser votre classe signal original ou la nouvelle. Comparer les deux implémentations en matière de temps de calcul.

## Appendice

Dans cet exemple de fonction pour sortir les données, on note  $v$  un tableau contenant le signal.

```

void sortie_fichier_freq(int nb,double t1,double t2){

    char string[255];
    FILE *fileresult=NULL;
    int i;
    double w=0.0, freq;

    sprintf(string,"%s%d","signal_freq_",nb);
    fileresult=fopen(string,"w");
    w = (n-1)/(2.0*(t2-t1));
    freq = 0.0;
    for(int i=0;i<n/2;i++){
        fprintf(fileresult,"%e %e\n",freq, v[i].module());
        freq = freq + w;
    }

    fclose(fileresult);
}

void sortie_fichier_time(int nb,double t1,double t2){

    char string[255];
    FILE *fileresult=NULL;
    int i;
    double dt=0.0, time;

    sprintf(string,"%s%d","signal_time_",nb);
    fileresult=fopen(string,"w");
    dt = (t2-t1)/n;
    time = t1;
    for(int i=0;i<n;i++){
        fprintf(fileresult,"%e %e\n",time, v[i].module());
        time = time + dt;
    }

    fclose(fileresult);
}

```