TRAITEMENT DU SIGNAL

Cahier de Travaux Pratiques

3° année

AVANT-PROPOS

Les travaux pratiques de traitement du signal ont pour objectifs tout d'abord d'apprendre à utiliser le logiciel Matlab, logiciel très répandu dans l'industrie et le monde de la recherche, de mettre en pratique les notions théoriques du cours et enfin d'illustrer concrètement quelques applications.

Rappels concernant le logiciel Matlab

Si ces rappels ne s'avéraient pas suffisants, il faut aussi savoir que de très nombreux cours sur Matlab sont disponibles sur Internet et que l'aide de Matlab est très utile et très complète avec un descriptif de chaque fonction et de nombreux exemples.

Matlab (MATrix LABoratory) a été originalement écrit pour fournir un moyen simple de programmer avec des matrices. Ses principales utilisations concernent : mathématiques et calculs ; développements d'algorithmes ; modélisation, simulation et prototypage ; analyse de données, exploration et visualisation ; graphiques scientifiques ; applications incluant un outil graphique pour construire des interfaces graphiques.

Matlab est un langage interprété conçu pour les calculs techniques. Matlab combine des calculs efficaces, une visualisation conviviale des résultats et un environnement de programmation facile à utiliser.

Matlab est un élément interactif dont l'élément de base est un tableau qui ne nécessite pas de dimensionnement. Ceci permet de résoudre de nombreux problèmes techniques particulièrement ceux qui sont formulés avec des matrices et des vecteurs. Sous cette condition ce sera infiniment plus rapide que d'écrire ou exécuter un programme de type C.

Enfin, Matlab est un logiciel de la société Mathworks et il faut savoir qu'il existe des clones de ce logiciel comme par Scilab. L'un des principaux avantages de Matlab est de fournir une aide en ligne détaillée et de nombreuses librairies de fonctions, l'ensemble de ces librairies n'étant disponible qu'à condition de posséder l'ensemble des « toolboxes », boites à outils dédiées à un domaine d'applications particulier.

Programmer avec Matlab

Les matrices

```
>> A=[1 2 3 ; 4 5 6 ; 7 8 9]
A  =   1 2 3
       4 5 6
       7 8 9
```

séparateur de colonnes : blanc ( ) ou virgule (,)

séparateur de ligne : point virgule (;)

début et fin de liste : […]

Les constantes

pi = 3,14159265

i ou j = sqrt(-1), unité imaginaire

Les opérateurs

+ addition

- soustraction

* multiplication

/ division

^ puissance

' transposition conjuguée

not ~ NON

and & ET

or | OU

xor OU EXCLUSIF

.* multiplication élément par élément

./ division élément par élément

Structures de contrôles

```
if expression1
   fonction1 ;
elseif expression2
   fonction2 ;
else fonction3 ;
end
```

```
switch expression
  case cas1
     fonction1 ;
  case cas2
     fonction2 ;
  otherwise
     fonction3 ;
```

```
while i<=10
   fonction ;
   i=i+1
end
```

```
for i=1 :10
   fonction ;
end
```

Fonctions

Fonctions élémentaires (trigonométriques, exponentielles, complexes, …)

>> help elfun

Fonctions pour les matrices

>> help elmat

Fonctions mathématiques (spécilisées, transformation des coordonées, …)

>> help specfun

Scripts et fonctions

Deux types de fichiers .m

Scripts

N'acceptent pas d'arguments d'entrée et ne retournent pas d'arguments de retour

Travaillent directement sur les données de l'espace de travai

Matlab exécute simplement les commandes du fichier

Fonctions

Peuvent accepter et retourner des arguments

Les variables internes sont locales à la fonction

Travaillent sur leur propre espace de travail

Le nom du fichier .m doit être le même que celui de la fonction

Vectorisation

Il faut éviter l'usage de boucle si possible pour gagner en temps de calcul.

```
N=50 ;
for i=1 :N
  y(i)=sin(2*pi*i/N);
end
```

```
N=50;
i=1:N;
y=sin(2*pi*i/N);
```

Précision et format des données

char, short, int, long, float, …

%s, %5d, …

Visualisation des données

Affichage sur l'écran (ligne de commande) : disp, sprintf, …

Types de représentation graphiques : plot, stem, stairs,hist …

Paramètres de représentation graphiques : subplot, grid, title,gtext, axis …

Fonctions pré-programmées

ones, zeros, sin, square, sawtooth, rand, randn, eye, …

Ecriture et lecture de fichiers

save, load, …

Aide à la programmation

edit « nom de la fonction matlab » (edit exemplematlab ;) : édition du fichier .m correspondant

whos : affichage de l'ensemble des variables et de leurs dimensions (équivalent du « workspace »)

help « nom de la fonction matlab » : description rapide pour utiliser la fonction

Aide à la création d'interface graphique

guide

Pour plus d'informations, se référer en premier lieu à l'aide de Matlab.

Séance 0 : Introduction à Matlab

A faire à la maison avant le premier TP

1. Opération sur les vecteurs

1.1) créer un vecteur temps nommé t1 composé de 1001 échantillons de 0 à 1 secondes en tuilisant la fonction **linspace**.

1.2) Pour définir un vecteur avec un pas défini, il peut être utile d'utiliser la syntaxe suivante :

t2 = (0 : delta_t : N*delta_t) ;

Avec cette syntaxe, créer un vecteur t2 de 1001 échantillons avec un pas temporel de 10 ms.

1.3) Vérifier la taille du vecteur avec la fonction **size**. Est-ce un vecteur ligne ou un vecteur colonne ?

1.4) Créer les vecteurs : t3 = [1+j , 2+j , 3+j]   t4 = t3'   t5=t3.'

1.5) Quelles différences voyez-vous entre ces vecteurs (vous pouvez vous aider de la fonction size) ? Expliquer les fonctions ' et .'

1.6) Réaliser la multiplication : t3*t3. Expliquez la signification de l'erreur renvoyée par Matlab (Il est très important de bien lire et comprendre les erreurs données par le logiciel. Cela vous aidera à corriger votre code rapidement)

1.7) Réaliser les multiplications suivantes : A1=t3*t4 ; A2=t3.*t3

1.8) Expliquer les différences ? A quoi sert le point avec le signe multiplié : .*

1.9) Concaténation de deux vecteurs : B = [t1 t2]

1.10) Selection d'une partie d'un vecteur : C=t1(1:20) ; % on selectionne les 20 premières valeurs.

1.11) Fonction find : grâce à l'aide de matlab, expliquer le fonctionnement de la fonction find à l'aide d'un exemple.

2. Génération d'un signal numérique et de son spectre

2.1) Créer un signal numérique S1 de N=101 échantillons représentant une cosinusoïde de fréquence f_0 = 2 kHz, la fréquence d'échantillonnage étant fixée à 8 kHz.

2.2) Afficher le signal grâce à la fonction **plot**. Grâce à l'aide, familiarisez-vous avec les options de cette fonction. Pour cela utilisez l'aide de Matlab.

2.3) Avec la fonction **axis**, afficher une seule période de la fonction.

2.4) Calculer l'énergie du signal en vous aidant de la fonction **sum** ainsi que la puissance.

2.5) A l'aide de la fonction FFT, calculer le spectre du signal et tracer le **module** du spectre. Grâce au **data cursor** afficher la fréquence des pics observés.

2.6) Si vous enlevez la fonction **abs** lors de l'affichage du spectre, quel warning voyez-vous apparaitre? Expliquez en la signification.

3. Formules du cours

3.1) Démontrer les formules de linéarité, de translation, de modulation, de dilatation/contraction.

3.2) Créer un signal carré de fréquence 50 Hz et visualiser le spectre.

Séance 1 : Introduction au traitement du signal numérique

L'objectif de ce travail est d'aborder la manipulation de signaux discrets en utilisant le logiciel Matlab ; les exemples traités sont simples et illustrent les représentations temps et fréquence des signaux ainsi que les problèmes liés à l'échantillonnage. A l'issue de la séance, il faut notamment être capable d'écrire un programme Matlab qui génère tout type de signal, calcule sa transformée de Fourier, et trace les différentes représentations associées ET il faut aussi savoir analyser et comprendre une courbe obtenue dans un contexte donné !

1. Génération d'un signal numérique

1.1) Générez un signal numérique, x1, représentant N = 100 échantillons d'une cosinusoïde de fréquence f_0 = 1 kHz, la fréquence d'échantillonnage étant fixée à 8 kHz.

1.2) Tracez ce signal en faisant apparaître en abscisse le temps. Pour cela aidez vous de la fonction « plot ».

1.3) Déterminez l'énergie du signal : Comment varie l'énergie lorsque N augmente ? Commenter ce résultat.

1.4) Déterminez la puissance du signal : Comment varie la puissance lorsque N augmente ? Est-ce en accord avec la théorie ?

2. Transformation de Fourier Discrète

2.1) Calculez le spectre du signal x1 avec la fonction FFT pour N = 100.

2.2) Tracez le module du spectre correspondant en fonction de la fréquence. Commenter le spectre obtenu. Est-il en accord avec ce que vous vous attendiez à trouver ? Quelle est la fréquence exacte du signal indiquée par le spectre ?

2.3) Tracez à nouveau le spectre mais pour N = 1000. Commenter le spectre obtenu. Quelle est la fréquence du signal indiquée par le spectre ? Conclure sur l'influence du nombre de points dans la déduction de la fréquence du signal.

2.4) Tracez à nouveau le spectre mais pour N = 104 et N = 405. Quelle est la fréquence du signal indiquée par les spectres ? Est-ce en accord avec la conclusion de la question 23) ? Expliquer précisément pourquoi nous observons cela et conclure sur ce cas particulier.

2.5) Comment varie l'amplitude des raies pour N = 1000 et N = 2000. Justifier cette variation.

2.6) Déterminez l'énergie du spectre pour N = 100. Comparer avec l'énergie obtenue à la question 1.3. (en discret, l'énergie du spectre est : **1/N***sum(abs(spectre).^2)).

3. Transformée de Fourier inverse - Reconstruction

A partir du spectre du signal x1, reconstruire le signal grâce à la transformée de Fourier inverse (fonction ifft). Comparer le signal reconstruit à l'aide de la FFT inverse, au signal original de la question 11).

Quelle condition doit-on respecter pour effectuer cette opération ?

4. Théorème de Shannon

4.1) Tracez les spectres des signaux numériques sinusoïdaux de fréquence 1 kHz et 7 kHz, échantillonnés à la fréquence de 8 kHz avec N = 100 échantillons. Expliquer les résultats obtenus.

4.2) D'une façon générale, quelle est la fréquence maximale autorisée sans qu'apparaisse un phénomène de repliement de spectre ? Si cette fréquence est dépassée, quelle est la fréquence apparaissant en réalité dans le spectre échantillonné ?

4.3) Générez deux sinusoïdes de fréquence 200 Hz échantillonnée à la fréquence F_{e1} = 1000 Hz avec N1 = 80 échantillons et échantillonnée à la fréquence F_{e2} = 250 Hz avec N2 = 20 échantillons.

4.4) Tracer sur une même figure ces deux signaux en faisant apparaître les échantillons.

4.5) Grâce à la fonction interp de matlab, interpoler les deux signaux d'un facteur 5. Tracez sur une même figure les signaux obtenus. Expliquez les résultats : pour cela vous pouvez mesurer la fréquence de chaque signal et la comparer avec celle obtenue par la fonction FFT. Conclure sur l'intérêt du la limite de Shannon.

5. Exercice d'approfondissement

Réalisez un algorithme qui calcule directement la Transformée de Fourier Discrète (TFD) du signal généré en utilisant la formule théorique du cours. Ecrire le programme correspondant et comparer le résultat avec la fonction FFT de matlab.

Séance 2.1 : Observation spectrale

L'objectif principal de ce TP est d'étudier plus particulièrement la représentation fréquentielle des signaux. Il s'agit de déterminer les bonnes (voire les meilleures) conditions d'observation du contenu spectral des signaux. Trois études de cas illustrent des exemples permettant de mettre en relief les méthodes d'observation et les options possibles. A l'issue de la séance, il faut être capable de déterminer les paramètres permettant d'observer correctement le spectre d'un signal, mettre en œuvre la procédure correspondante et avoir compris les notions théoriques associées.

Avant-propos : afin d'examiner plus précisément les caractéristiques des spectres des signaux, deux caractéristiques très importantes sont la précision et la résolution fréquentielle. La précision est rattachée à la mesure d'une fréquence. Par exemple, lorsque le spectre d'une sinusoïde est déterminé par une TFD, le tableau des valeurs obtenues donne une valeur approchée de la fréquence : son exactitude ou précision est directement liée au nombre de points. La précision ne doit pas être confondue avec la résolution fréquentielle qui est la capacité à détecter des fréquences distinctes contenues dans un même signal.

1. Observation d'une sinusoide

Soit la suite x(n) = cos(2π.F0.n.Te) obtenue par échantillonnage du signal cos(2πF0t). La fréquence F0 sera prise égale à 1 kHz, la fréquence d'échantillonnage à Fe = 8 kHz et la suite sera composée de Ns = 85 échantillons.

1.1) Représenter le module de la Transformée de Fourier à discrète X(f) de la suite x(n), en utilisant la fonction **fft**.

1.2) Calculer et représenter le spectre de X(f) en utilisant **fft(x,Nfft)** avec Nfft = 2000. Comparer le résultat avec celui de la question précédente.

En vous aidant de l'aide de matlab, expliquer l'opération effectuée par l'option Nfft. Expliquer alors le résultat obtenu.

1.3) Calculer et représenter le spectre de X(f) avec Ns = 2000. Comparer le résultat avec celui de la question précédente.

1.4) Ajoutez au signal x(t) un signal 0.001*cos(2π.F1.t) de fréquence F1 égale à 2 kHz. Représentez le module du spectre en linéaire et en décibels. Quel est l'intérêt de la représentation en dB? (attention sous matlab : la fonction log correspond au ln et la fonction log10 correspond au log).

2. Effets du fenêtrage sur la résolution

3.1) Générer le signal x(n) obtenue par échantillonnage du signal exp(2πjF_0*n*Te) à la fréquence d'échantillonnage F_e = 1/Te pour une fréquence normalisée $f_0 = F_0 / F_e = 0.2$. Vous prendrez un nombre d'échantillons N égal à 2000. Astuce : remplacez F_0 par son expression dans la formule de x(n)

3.2) Tracer le spectre du signal x(n) en fonction de la fréquence normalisée $f_{norm} = f / F_e$. Quelle est la fréquence du signal mesurée sur ce spectre ?

L'objectif de l'exercice est d'extraire une partie du signal de taille Ns=32 grâce à une opération de fenêtrage. Pour le moment, vous avez surtout utilisé la fenêtre de type rectangle pour réaliser cette opération. En pratique, il existe beaucoup de fenêtre disponible. Nous allons à présent comparer la fenêtre rectangle à une autre fenêtre très connue, la fenêtre de Hamming.

Le fait de limiter à Ns le nombre d'échantillons d'un signal peut être vu comme la multiplication terme à terme de la totalité du signal x(n) (de longueur N) par une fenêtre (de longueur Ns). Chaque échantillon est multipliée par $c \cdot \omega(n)$, où c est une constante et $\omega(n)$ désigne :

- soit la fenêtre rectangulaire définie par : $\omega_R(n) = 1 \ \forall \ n \in [0 ; L-1]$, et 0 ailleurs ;

- soit la fenêtre de Hamming définie par : $\omega_{HL}(n) = 0{,}54 - 0{,}46\cdot\cos(\frac{2\pi n}{L-1})\ \forall\ n \in [0\ ;\ L-1]$, et 0 ailleurs.

3.3) A faire à la maison avant le TP : Pour une fenêtre quelconque, déterminer **théoriquement** la constante c telle que l'amplitude maximale de la TFD du signal fenêtré en f_0 soit égale à 1. Montrer que c = 1/L pour une fenêtre rectangle et 1/(0.54*L) pour celle de Hamming

3.4) Ecrire un programme qui affiche le module en décibels de la Transformée de Fourier de x(n) pour la pondération rectangulaire et la pondération de Hamming.

3.5) Pour les deux fenêtres, relever la largeur du lobe principal et la hauteur du premier lobe secondaire (la hauteur sera prise en décibels par rapport au lobe principal).

3.6) Illustrer vos conclusions à partir de signaux simples comme la somme de signaux sinusoïdaux de fréquences voisines ou/et d'amplitudes différentes. Montrer en particulier l'importance du choix de la fenêtre d'apodisation en fonction du signal à étudier.

3. Exercice supplémentaire :

Observation d'un signal modulé en amplitude

Soit le signal m(t), à temps continu, de spectre M(f) : $m(t) = \cos(2\pi f_1 t) + 1{,}8.\cos(2\pi f_2 t) + 0{,}9.\cos(2\pi f_3 t)$ avec $f_1 = 2310$ Hz, $f_2 = 3750$ Hz et $f_3 = 4960$ Hz.

Pour transmettre par ondes radiofréquences ce signal, une modulation d'amplitude avec porteuse est effectuée telle que le signal modulé s'écrit : $s(t) = (1 + k.m(t)).\cos(2\pi f_0 t)$ avec $f_0 = 50$ kHz et k = 0,25.

2.1) Déterminer **théoriquement** l'expression du spectre de m(t), M(f).

2.2) Déterminer **théoriquement** l'expression du spectre de s(t), S(f). En déduire les fréquences contenues dans le signal modulé.

2.3) Choisir **en justifiant** une fréquence d'échantillonnage puis en déduire la durée minimale d'observation du signal qui permet de distinguer par analyse de Fourier, les fréquences présentes dans le spectre de s(t).

2.4) Quel est le nombre de points de la FFT nécessaire, si l'on souhaite lire le spectre avec une précision de 10 Hz ?

2.5) Ecrire un programme qui trace le spectre du signal modulé avec une résolution de 10 Hz.

Séance 2.2 : représentation par spectrogrammes

1. Spectrogramme d'un signal

1.1) Afficher un signal avec une fréquence variant quadratiquement puis linérairement avec le temps. La fréquence commence à 1hz et atteind 400 hertz à t = 2 secondes. Ces signaux s'appellent généralement "chirp".
En utilisant la fonction "chirp" de matlab, réaliser ces signaux et visualiser le signal et le spectre.
1.2) L'objectif est à présent d'utiliser une représentation de type spectrogramme pour mettre en evidence la variation temporelle de la fréquence. Un spectrogramme consiste à afficher la variation du spectre du signal en fonction du temps en utilisant une Short time Fourier Transform. La fonction « spectrogram » vous permettra de faire cela.
En vous aidant de l'aide, expliquer le fonctionnement de la fonction spectrogram. Pour cela, vous pourrez vous appuyer sur des schémas.
1.3) Effet du type de fenêtre : Faire le spectrogramme pour nfft = 320, une fenêtre de 20 points et un overlap de 10 points.
Comparer le spectrogramme de la fenêtre rectangle avec la fenêtre hamming.
1.4) Effet de nfft : Faire le spectrogramme pour nfft = 20 et 320, une fenêtre de hamming de 20 points et un overlap de 10 points.
1.5) Quel est l'intérêt du spectrogram comparé à la FFT standard.

2. Spectrogramme d'un signal audio

2.1) Charger les fichiers audio electro, rap, metal1, metal2 et guitare à l'aide de la fonction audioread.
2.2) Les fichiers peuvent être écouté avec la fonction « sound ». Faire varier le paramètre Fs de la fonction et expliquer les différences que vous entendez.
2.3) A partir des informations renvoyées par audioinfo, estimer la taille du fichier rap.wav. Comparé avec la valeur donnée par windows. D'où peuvent venir les petites différences ?
2.4) Les fichiers étant très long, nous allons les tronquer pour alléger les traitements. Enregistrer les versions tronquées de n = 2000e3 à 2800e3.
2.5) En vous basant sur l'étude de l'exercice 1, afficher sur le spectrogrammes de l'extrait rap avec une fenêtre de hamming de 100, un overlap de 50 et un nFFT de 320.

3. Recherche d'un extrait musical

Le fichier extrait.mat contient un extrait d'une des musiques précedentes. Proposer et programmer une méthode permettant de localiser facilement l'extrait. Dans quel fichier et à quelle position se trouve-t-il? Pour charger l'extrait, utilisez la fonction « load ».

4. Exercice supplémentaire :

effet sonore simple : reverse

4.1) Charger le fichier piano_d2.wav. A quelle note de musique correspond ce son?
4.2) Effet de renversement temporel. C'est un effet très simple qui est réalisé en inversant l'ordre des échantillons. Il permet par exemple de creer des sons montant progressivement en volume, chose difficile à réaliser avec des instruments classiques. A l'aide des fonctions fliplr et flipud,

appliquez cet effet au fichier piano_d2.wav. Montrer l'effet du renversement à l'aide d'un spectrogramme. D'un point de vu sonore quel effet entendez-vous?

Séance 3 : Synthèse d'un filtre à réponse impulsionnelle finie

La fréquence d'échantillonnage est choisie égale à F_e = 8 kHz.

1. Synthèse d'un filtre

1.1. A faire avant le TP : En utilisant la méthode de la fenêtre (décrite dans le cours), déterminer **théoriquement** la suite h(n) des N = 35 coefficients d'un filtre RIF passe-bas idéal dans la bande [0, 1000 Hz].

1.2. Représenter la réponse impulsionnelle déterminée. (Attention à la fonction sinc de Matlab. Voir la section « more about » de l'aide… oui c'est important donc regardez encore si vous ne trouvez rien)

1.3. Vérifier le résultat obtenu en traçant la réponse en fréquence du filtre obtenu. Comparer gabarie intial du filtre, commenter et conclure.

1.4. Représenter en fonction de la fréquence, le module en décibels et la phase en radians du filtre calculé. Commenter et vérifier que le filtre synthétisé est un filtre à phase linéaire. Quel peut être l'avantage d'une telle caractéristique ?

2. Génération d'un signal composite

2.1. Créer un signal sinusoïdal composite noté « sig » d'une durée totale de 2 secondes tel que :

si t ∈ [0 ; 0,3 s], « sig » d'amplitude 3 et de fréquence 500 Hz ;

si t ∈ [0,3 s ; 0,8 s], « sig » d'amplitude 1 et de fréquence 800 Hz ;

si t ∈ [0,8 s ; 1,2 s], « sig » d'amplitude 2 et de fréquence 1200 Hz ; si t ∈ [1,2 s ; 1,7 s], « sig » d'amplitude 2 et de fréquence 400 Hz ;

si t ∈ [1,7 s ; 2 s], « sig » d'amplitude 3 et de fréquence 1500 Hz.

2.2. Représenter ce signal ainsi que le module de son spectre.

2.3. Représenter le spectre sous la forme d'un spectrogramme.

3. Filtre à fenêtre rectangle

3.1. Appliquer le filtre synthétisé à la question 1 au signal de la question 2 en utilisant la fonction « filter ».

3.2. Représenter sur une même figure, d'une part le signal avant et après filtrage et d'autre part, le module du spectre du signal avant et après filtrage. Analyser et commenter. Vérifiez que l'amplitude en sortie du filtre à 1200 Hz et 1500 Hz correspond bien à l'amplitude d'entrée multiplié par la fonction de transfert.

3.3. Filtre RIF à fenêtre de Hamming

Répéter les questions précédentes avec le filtre issu de la pondération par la fenêtre de Hamming :

Une variante du filtre calculé consiste à appliquer à la suite h(n) une fenêtre de Hamming dont l'équation a pour expression : $\omega_{HL}(n) = 0{,}54 - 0{,}46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), \forall\, n \in [0\,;\,N-1]$ et 0 ailleurs.

Remarque : par défaut, la fenêtre est une fonction rectangulaire.

a) Représenter cette fenêtre. Commenter.

b) Représenter sur une même figure pour les deux filtres obtenus (sans et avec fenêtre de Hamming) : les réponses impulsionnelles et les réponses en fréquence.

c) Appliquer le filtre.

4.3 Filtre RII vs filtre RIF

Dimensionnez un filtre RII de type Butterworth de fréquence de coupure 1 kHz. L'ordre du filtre doit être choisi pour avoir la même atténuation que le filtre RIF de la question 4.1 à la fréquence de 1.24 kHz.

a) Quelle différence voyez-vous au niveau de l'ordre du filtre entre un filtre RIF et RII ?

b) Comment est la phase et le retard du filtre RII ?

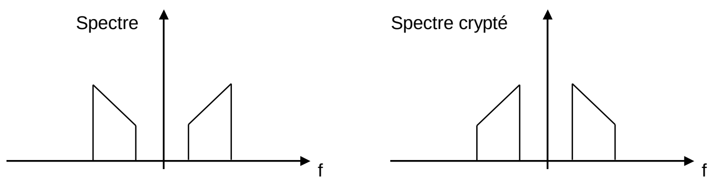
c) Conclure sur les avantages et inconvénients du filtre RII par rapport au filtre RIF

d) Exportez le filtre dans le « workspace » de Matlab (File => Export) puis appliquez le au signal de la question 2.

Approfondissement

5. Cryptage type CANAL +

Le cryptage du son sur Canal + utilise la technique dite du "retournement" de spectres, qui consiste à faire pivoter les parties du spectre dans les fréquences positives et négatives d'un demi-tour autour d'un certain axe. En pratique, on suppose que le spectre du son est situé (dans les fréquences positives) entre 100 Hz et 10 kHz.



1. Montrer que l'opération de cryptage correspond à une translation du spectre d'une fréquence f_0 suivie d'un filtrage passe-bas.

2. En quoi consiste l'opération de décryptage ?

3. Ecrire un programme qui effectue le cryptage et le décryptage d'un signal. Prendre comme signal de parole, une des musiques du TP spectrogramme.