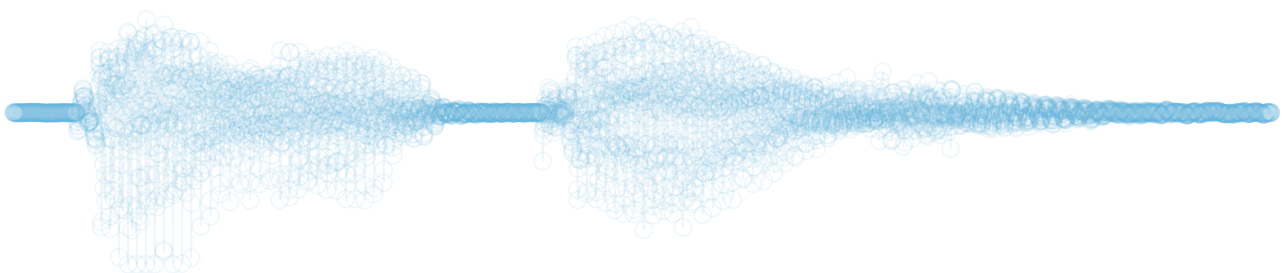




**Année universitaire
2020 - 2021**

Traitement numérique du signal – 3ETI
TP1

Signaux discrets et TFD



Eric Van Reeth

1 Présentation du TP

Ce TP est consacré à la génération de séquences et de signaux, à leur représentation temporelle et à leur représentation spectrale via la TFD calculée à l'aide d'algorithmes rapides. Il est réalisé sous Matlab. Un rappel sur des notions de base de Matlab est fourni en annexe.

1.1 Préparation

- Lire attentivement l'énoncé et les annexes avant la séance
- Avoir travaillé son cours sur la TFD et l'apporter
- Rédiger les réponses aux questions suivantes :

1. Qu'est-ce-qu'une fréquence dite « réduite » ou « normalisée » ? En quelle unité l'exprime-t-on ?
2. Comment sont graduées les abscisses (valeur minimale, maximale, incrément) d'une TFD- N points lorsqu'elle est affichée : 1) en fonction des indices, 2) en fréquence réduite, et 3) en Hz ?
3. Soit $s[k]$ une séquence à valeurs réelles. Calculer $S^*(f)$, conjugué de sa Transformée de Fourier à Temps Discret $S(f)$, et l'exprimer en fonction de $S(f)$.
4. Si en plus $s[k]$ est une séquence paire, calculer $S(f)$ en tenant compte des deux propriétés (réelle et paire). En déduire que sa TTFD est également réelle et paire.

1.2 Évaluation

Chaque binôme devra rédiger un compte-rendu. En plus du compte-rendu, **l'intégralité du code développé devra également être rendu.** Le code et le compte-rendu seront compressés dans **une archive (.zip) dont le nom contiendra tous les noms des membres du binôme.** L'archive sera remise sur le dépôt cpe-campus dans un **délai d'une semaine après le TP.**

Voici quelques règles à suivre pour la rédaction du compte-rendu :

- Les réponses à la préparation doivent être intégrées dans le compte-rendu
- La notation portera sur votre capacité à répondre à **chacune des questions posées**, mais surtout sur la qualité des **commentaires et justifications qui pourront représenter plus de la moitié des points**
- La présentation, la qualité du français et de l'orthographe seront prises en compte
- Les figures doivent être numérotées, et toutes mesures, justifications et commentaires concernant les figures doivent être rédigés dans le compte rendu en renvoyant précisément à la figure numérotée s'y rapportant
- Le **compte-rendu sera remis au format pdf**
- Le nom de l'archive doit contenir les explicitement les noms des membres du binôme

2 Génération de séquences et affichage

2.1 Exercice 1

Une des séquences de base en Traitement du Signal est le signal périodique à fréquence pure :

$$s[k] = A \sin(2\pi f_0 k + \phi) \quad \text{pour } k = k_d, \dots, k_f$$

Pour générer une telle séquence, il faut fixer les trois paramètres propres au signal (amplitude, fréquence et phase), ainsi qu'un indice de début k_d , et un indice de fin k_f .

1. Créer une fonction qui permettra de générer des séquences $s[k]$ du type ci-dessus de longueur finie. Les paramètres d'entrée doivent être : A , f_0 , ϕ en radians, k_d et k_f . Cette fonction doit ressortir : k le vecteur indices, s le vecteur signal. Attention, en Matlab, lorsqu'une fonction est écrite dans un nouveau script, elle doit porter le même nom que le script.
2. Écrire un script principal qui appelle la fonction précédente pour générer les séquences ci-dessous.

$$\begin{aligned} s_1[k] &= 2 \sin\left(\frac{\pi}{17}k\right) && \text{pour } 0 \leq k \leq 26 \\ s_2[k] &= -4 \cos\left(\frac{\pi}{13}k\right) && \text{pour } -13 \leq k \leq 13 \\ s_3[k] &= \sin\left(\frac{3\pi}{16}k + \frac{\pi}{4}\right) && \text{pour } -10 \leq k \leq 10 \\ s_4[k] &= \cos\left(\frac{\pi}{\sqrt{23}}k\right) && \text{pour } 0 \leq k \leq 50 \end{aligned}$$

Dans ce script, réaliser le tracé des séquences à l'aide de `stem` avec en abscisse les indices corrects et ce, dans une seule fenêtre avec quatre « imagettes » distinctes (fonction `subplot`).

2.2 Exercice 2

Fréquemment, un signal à temps discret est produit par échantillonnage d'un signal à temps continu. Considérons le signal :

$$s(t) = A_1 \sin(2\pi\nu_1 t + \phi_1) + A_2 \sin(2\pi\nu_2 t + \phi_2)$$

Pour générer une séquence $s[k]$ correspondant à l'échantillonnage de $s(t)$ pendant une durée finie, il faut fixer les paramètres du signal (A_1 , ν_1 , ϕ_1 , A_2 , ν_2 , ϕ_2) ainsi qu'une date de début t_d et une date de fin t_f , plus une cadence d'échantillonnage ν_e , soit neuf paramètres.

1. Créer une nouvelle fonction qui permettra de générer une séquence $x[k]$ correspondant aux échantillons de $x(t)$ entre t_d et t_f . Cette fonction doit respecter les exigences suivantes :
 - Elle doit faire appel à la fonction élaborée dans l'exercice 1
 - Les indices k doivent être des valeurs entières
 - La séquence générée doit impérativement posséder une valeur en $t = 0$
 - Les échantillons situés aux extrémités doivent être à l'intérieur de l'intervalle temporel imposé

Les paramètres d'entrée doivent être : (A_1 , A_2), les amplitudes, (ν_1 , ν_2) en Hz, (ϕ_1 , ϕ_2) en degrés, t_d et t_f en secondes, ν_e fréquence d'échantillonnage en Hz. Les grandeurs de sortie doivent être : t un vecteur dates, x le vecteur contenant la séquence et K la longueur de la séquence.

2. Dans un script principal, faire exécuter cette nouvelle fonction avec le jeu de paramètres suivant :

$$\begin{aligned} A_1 &= 8 & \nu_1 &= 271 \text{ Hz} & \phi_1 &= 25^\circ \\ A_2 &= 5 & \nu_2 &= 1147 \text{ Hz} & \phi_2 &= 38^\circ \\ t_d &= -8.1 \text{ ms} & t_f &= 12.05 \text{ ms} & \nu_e &= 5000 \text{ Hz} \end{aligned}$$

Puis tracer les échantillons de x en fonction des dates $t = kT_e$ exprimées en millisecondes. Vérifier que la séquence affichée correspond aux contraintes exigées.

3. Expliquer les transformations des paramètres opérées pour les fournir à la fonction de l'exercice 1.

3 Calculs de TFD

La TFD N -points X d'une séquence x de longueur K (avec $K \leq N$) peut être obtenue à l'aide de la commande $X = \text{fft}(x, N)$;

X est en général une grandeur complexe. Pour visualiser le résultat, on peut s'intéresser à quatre grandeurs : partie réelle, partie imaginaire, module et phase.

3.1 Exercice 3

1. Générer et tracer une nouvelle séquence $x[k]$ avec la fonction de l'exercice 2 en utilisant le jeu de paramètres ci-dessous :

$$\begin{array}{lll} A_1 = 8 & \nu_1 = 271 \text{ Hz} & \phi_1 = 25^\circ \\ A_2 = 5 & \nu_1 = 1147 \text{ Hz} & \phi_1 = 38^\circ \\ t_d = 0 \text{ ms} & t_f = 18.41 \text{ ms} & \nu_e = 5000 \text{ Hz} \end{array}$$

2. Calculer $X_1[n]$, sa TFD K -points où K est la longueur de la séquence $x[k]$ obtenue
3. En utilisant les fonctions `real`, `imag`, `abs`, `angle`, tracer dans une même fenêtre les parties réelle et imaginaire de $X_1[n]$ ainsi que son module et sa phase :
 - avec des abscisses graduées en indice de 0 à $N - 1$ (ici $N = K$)
 - avec des abscisses graduées en fréquence réduite
 - avec des abscisses graduées en fréquence réelle en kHz
4. Calculer $X_2[n]$, sa TFD 512-points de la séquence $x[k]$
5. Tracer le module de $X_2[n]$ (en utilisant la fonction `plot`) avec des abscisses graduées en indice de 0 à $N - 1$, avec des abscisses graduées en fréquence réduite, avec des abscisses graduées en kHz, le tout sur la même figure.
6. En considérant le module de $X_1[n]$, relever les mesures des fréquences des quatre composantes maximales en indiquant à quel indice elles se trouvent, en mesurant leurs positions en fréquence réduite, puis en donnant les fréquences correspondantes en Hz
7. Vérifier les relations qui lient les composantes dans les trois modes de graduation
8. Expliquer la présence de quatre composantes sur chaque module d'une TFD et expliquer le lien entre ces raies
9. Comparer les fréquences mesurées avec les fréquences programmées. Expliquer les différences constatées.
10. Refaire ces mesures sur $X_2[n]$. Comparer avec les résultats précédents. Interpréter les différences.
11. Expliquer la présence de lobes latéraux autour de chaque composante.

3.2 Exercice 4

On souhaite réaliser le calcul de la TFD d'une séquence $x[k]$ issue de la fonction de l'exercice 2 obtenue avec le jeu de paramètres ci-dessous :

$$\begin{array}{lll} A_1 = 8 & \nu_1 = 271 \text{ Hz} & \phi_1 = 90^\circ \\ A_2 = 5 & \nu_1 = 1147 \text{ Hz} & \phi_1 = 90^\circ \\ t_d = -10.12 \text{ ms} & t_f = 10.12 \text{ ms} & \nu_e = 5000 \text{ Hz} \end{array}$$

1. Générer la séquence temporelle x et l'afficher. Cette séquence doit être réelle et paire.

2. En vue du calcul d'une TFD-512 points, organiser les valeurs contenues dans x pour les présenter correctement à l'algorithme de calcul rapide utilisé par la fonction `fft`. Justifiez la réorganisation et tracer la nouvelle séquence $y[k]$ pour vérification.
3. Réaliser le calcul de la TFD-512 points
4. Tracer partie réelle, partie imaginaire, module et phase graduées en Hz

Annexes

Rappels généraux sur Matlab Puisque un des types de données numériques valides sous Matlab est une matrice de dimensions $M \times N$, les signaux sont représentés par des vecteurs. Il s'agit, soit de matrices de dimension $M \times 1$ si ce sont des vecteurs colonnes, soit de matrices de dimension $1 \times N$ si ce sont des vecteurs lignes. Sous Matlab, tous les signaux ou séquences doivent être de durée finie. Ceci contraste fortement avec la résolution de problèmes analytiques où l'on peut utiliser une formule mathématique pour représenter un signal de durée infinie.

Un point important est le domaine d'indexation associé à un vecteur signal. Matlab suppose qu'un vecteur x est indexé de 1 à N si N est la longueur du vecteur. Il faut y prendre garde car les formulations utilisées en T.N.S. sont souvent indexées de 0 à $N-1$, voire comportent des indices négatifs. Cela ne pose pas en général de problème majeur. Il faut cependant en tenir compte lors des calculs et tout particulièrement lors des tracés quand on veut afficher des coordonnées correctes (en particulier pour les abscisses). Cette gestion des coordonnées est indispensable en Traitement du Signal pour ne pas commettre ultérieurement d'erreurs d'interprétation.

Un dernier point concerne l'utilisation de la notation matricielle pour générer des signaux. Cette notation de haut niveau est un des points forts de Matlab : les boucles de type `for` sont la plupart du temps inutiles. Par exemple, pour créer une sinusoïde, il faut appliquer la fonction `sin` à un vecteur contenant les dates auxquelles on souhaite disposer des échantillons.

Comment programmer une fonction ? Une fonction est forcément décrite dans un script de type m-file (par exemple appelé `fexo.m`). La première ligne doit impérativement comporter l'instruction `function` sous la forme suivante :

```
function [vs1,vs2,...] = fexo2(ve1,ve2,...);
```

où `ve1`, `ve2`, ..., sont les noms des variables d'entrée que vous utiliserez dans la fonction et `vs1`, `vs2`, ..., les noms des variables de sortie. Lorsque la saisie de vos commandes est terminée, enregistrez votre m-file sous le nom `fexo.m`.

Si la fonction est définie dans le m-file par : `function [vs1,vs2]=fexo(ve1,ve2,ve3)` par exemple, et que l'on souhaite, dans le script principal (ou dans la fenêtre de commande Matlab), récupérer les valeurs renvoyées dans le premier paramètre dans un vecteur nommé `signal`, et les valeurs renvoyées dans le second paramètre dans un vecteur nommé `indices`, faire exécuter la fonction en utilisant :

```
[signal,indices]=fexo(3 paramètres dans l'ordre défini dans fexo et séparées par une virgule);
```

Rappels sur les tracés

- Pour ouvrir la figure n°7, utilisez `figure(7)` (à mettre avant les tracés que l'on veut faire).
- Les deux fonctions principales de tracé, utiles pour ce TP, sont `stem` et `plot`.

- Pour obtenir plusieurs tracés sur la même figure, il faut utiliser avant chaque tracé, la commande `subplot(m, n, p)`, qui découpe la fenêtre figure en une matrice de dimension $m \times n$ de petites figures et sélectionne la p -ième figure pour le tracé suivant. Les tracés seront numérotés horizontalement comme le montre l'exemple ci-dessous.
`subplot(2, 3, p)`

p=1	p=2	p=3
p=4	p=5	p=6

Conseil : utilisez `subplot(2, 2, p)` pour afficher 4 figures plutôt que `subplot(4, 1, p)`

- À la suite d'un tracé, on peut ajouter sur la figure :
 - un titre, avec la commande : `title('commentaire de titre')`
 - un libellé d'abscisse avec : `xlabel('commentaire d"abscisse')`

Pour obtenir l'affichage de lettres grecques dans les commentaires, utilisez `\` suivi du nom de la lettre : Exemple : `\alpha` fournira α .

- Il est possible d'afficher des valeurs de variables dans un titre comme le montre l'exemple suivant pour l'affichage de la longueur M d'une séquence : `title(['s(k) a ', num2str(M), 'échantillons'])`

Opérations sur les vecteurs

- Portion d'un vecteur : `y= x(n1:n2)`
- Le produit `*` de deux vecteurs donne une matrice.
- Le produit terme à terme de deux vecteurs de même taille s'effectue par l'opération `.*` (de même `./` ou `.^` pour l'élévation à une puissance).
- Concaténation de plusieurs vecteurs lignes à la suite : `y=[x1 x2 x3]`