Traitement numérique du signal

2019/2020

COMPOSANTE SIGNAL

Cours-Exercices-Tds-APP

Florence ROSSANT

SOMMAIRE

SOMMAIRE **3**

1. INTRODUCTION 5

1.1. Contexte *5*

1.2. Objectifs pédagogiques de le composante signal *6*

1.3. Organisation et évaluation des compétences *7*

1.3.1. Organisation générale 7

1.3.2. Evaluation des compétences 8

1.3.3. Planning général 8

2. SIGNAUX ECHANTILLONNES 10

2.1. Valeur moyenne, énergie, puissance, valeur efficace *10*

2.1.1. Signaux continus 10

2.1.2. Signaux échantillonnés 11

2.2. Amplification/atténuation d'un signal, gain *13*

2.3. Rapport signal à bruit *13*

2.4. Corrélation *14*

2.5. EXERCICES *16*

2.6. Travail en séances d'APP *18*

3. ANALYSE FREQUENTIELLE 21

3.1. Transformée de Fourier d'un signal continu *21*

3.2. Transformée de Fourier discrète *22*

3.3. Signaux stochastiques et densité spectrale de puissance *23*

3.4. Analyse spectrale *24*

3.5. EXERCICES *26*

3.6. Travail en séance d'APP *28*

4. NUMERISATION D'UN SIGNAL 30

4.1. Echantillonnage *30*

4.2. Interpolation *32*

4.3. Quantification *33*

4.4. En pratique… *34*

4.5. EXERCICES *34*

4.6. Travail en séances d'APP *36*

5. FILTRAGE NUMERIQUE 39

5.1. Fonction de transfert *39*

5.2. Principaux filtres *39*

5.3. Gabarits et synthèse *40*

5.4. EXERCICES *41*

5.5. Travail en séance d'APP *42*

6. IMPLEMENTATION MATERIELLE 46

6.1. EXERCICES *46*

6.2. Travail en séance d'APP *46*

1. INTRODUCTION

Un signal est une fonction de une ou plusieurs variables, engendrée par un phénomène physique. Les variations du signal sont porteuses d'information. Par exemple :

- un signal audio est une onde sonore qui se propage dans l'espace ; elle est transformée en un signal électrique par le microphone pour être stockée, transmise ou traitée.
- Le signal d'une télécommande infrarouge est généré par une diode électroluminescente (DEL). Le message numérique module la lumière IR émise ; ce signal est reçu par le capteur infrarouge puis décodé.

De nombreux signaux sont des fonctions 1D du temps, modélisables par une fonction de type $s(t)$. Mais il existe des signaux 2D fonctions de l'espace 2D, comme les images fixes, $I(x, y)$, des signaux vidéos, $V(x, y, t)$, des signaux vidéo 3D, $V(x, y, z, t)$, etc.

Les champs applicatifs du traitement du signal sont très vastes car notre monde regorge de capteurs qui fournissent des quantités de signaux à traiter pour être interprétés et utilisés. Tous les systèmes numériques développés par les ingénieurs traitent des signaux. **Connaître les notions fondamentales sur l'acquisition et le traitement numérique des signaux est donc indispensable à tout ingénieur du numérique !** Le but de cette composante APP est de vous amener à acquérir ces fondamentaux.

Le traitement du signal est donc omniprésent dans le domaine du numérique ; cette discipline s'appuie sur les domaines de la physique, des mathématiques, de l'informatique (algorithmique et programmation) et de l'électronique. Les objectifs du traitement du signal sont de :

- Transmettre des informations (télécommunications),
- Améliorer la qualité d'un signal,
- Compresser, coder les signaux,
- Interpréter, reconnaître, prédire….

1.1. Contexte

Le contexte de la mission signal est la réponse à l'appel d'offre de la société INFINITE MEASURES. Il s'agit, dans cette composante, de proposer des solutions à deux fonctionnalités au moins du cahier des charges : les deux fonctions indispensables qui sont la mesure du rythme cardiaque (FT1-1-2-1) et la mesure de la qualité de reconnaissance d'une tonalité (FT1-1-3-2).

Pour la fonction de mesure de rythme cardiaque (FT1-1-2-1), on travaillera sur des signaux acquis avec le capteur de la carte électronique.

Pour la fonction de mesure de la capacité de reproduction d'une tonalité (FT1-1-3-2), on adoptera le protocole d'évaluation suivant : le programme joue un son dont la fréquence a été tirée aléatoirement dans l'intervalle $[f_{min}, f_{max}]$, avec $f_{min} = 130Hz$ $f_{max} = 4000Hz$. L'utilisateur écoute ce son, et à l'aide d'un générateur de sons, tente de reproduire au mieux la fréquence jouée. Le programme fait l'acquisition de ce signal, l'analyse et attribue un score à l'utilisateur, fonction de l'écart entre la fréquence voulue et la fréquence générée.

Ces deux fonctionnalités seront mises au point sous Matlab et implémentées sur le microcontrôleur durant la composante signal.

Matlab est un environnement de calcul scientifique, largement utilisé dans le monde industriel, qui permet de résoudre des problèmes scientifiques et techniques. Nous utiliserons cet environnement pour simuler intégralement nos systèmes, afin de les mettre au point, de les optimiser et de les valider. Les algorithmes ainsi définis seront ensuite implémentés sur le microcontrôleur de la carte électronique. Les expérimentations menées sous Matlab devront donc prendre en compte les contraintes techniques imposées par le matériel, afin d'aboutir à une solution implémentable.

D'autres missions feront l'objet d'études sous Matlab, sans toutefois d'implémentation matérielle.

1.2. Objectifs pédagogiques de le composante signal

Compétences et savoir-faire en traitement du signal :

L'objectif est d'appréhender quatre notions fondamentales :

- L'analyse des signaux dans le domaine temporel,
- L'analyse des signaux dans le domaine fréquentiel,
- La numérisation des signaux,
- La notion de filtrage.

Les signaux numériques sont au cœur des systèmes électroniques et informatiques que nous utilisons au quotidien et que les ingénieurs conçoivent. Maîtriser ces quatre notions est donc absolument fondamental pour tout ingénieur dans le domaine des technologies du numérique.

Les compétences à acquérir lors de cette APP sont les suivantes :

- Estimation de la puissance d'un signal, mesure en Watt et en dBm
- Méthodes de détection d'un signal
- Définir les paramètres pour la numérisation d'un signal : fréquence de coupure du filtre anti-repliement, choix de la fréquence d'échantillonnage, choix du nombre de bits de quantification.
- Savoir calculer le débit d'un signal numérique et l'espace nécessaire à son stockage.
- Comparaison de signaux par corrélation.
- Analyse fréquentielle : calcul et interprétation d'une FFT.

- Définition du gabarit d'un filtre, avec prise en compte des contraintes matérielles
- Filtrage dans le domaine temporel et dans le domaine fréquentiel : méthodes de calcul et implémentation.

Les savoir-faire à acquérir sont :

- Algorithmique fondamentale
- Méthodologie de simulation et de validation de systèmes numériques
- Langage Matlab

Toutes ces compétences seront évaluées lors de l'APP et via un examen individuel.

Compétences générales

Enfin, une attention particulière sera donnée à la **rédaction des livrables**, qui doivent, **pour chaque problème** comporter les sections suivantes :

- Introduction : position du problème
- Description de la méthode proposée pour la résolution du problème : expliciter la méthode et les choix effectués par des phrases et **formaliser** la méthode par des équations. Aucun code ou pseudo code ne doit figurer dans le rapport.
- Résultats expérimentaux (illustrés et analysés).
- Conclusion

La rédaction doit être concise et efficace. Le lecteur doit pouvoir reproduire le système décrit, paramétrage compris, sous l'environnement de son choix.

Les présentations orales doivent suivre la même méthodologie.

1.3. Organisation et évaluation des compétences

1.3.1. Organisation générale

L'apprentissage du traitement du signal s'effectue dans le cadre de l'apprentissage par projet, à raison de 10 heures par semaine réparties sur 4 séances. L'essentiel du cours est donné au fur et à mesure de la progression dans ce livret. Le travail en séances d'APP consiste à s'approprier ce cours, effectuer les différents exercices demandés sous Matlab, résoudre les problèmes et préparer les livrables. Les exercices sont là pour vous aider à comprendre les concepts et vous aider dans la programmation en Matlab. Ils ne font pas l'objet de compte-rendu mais sont vérifiés durant les séances d'APP par le tuteur. Ils préparent la résolution des problèmes. Par conséquent, **le travail demandé doit être effectué séquentiellement dans l'ordre indiqué dans le livret, et aucune organisation tendant à paralléliser les tâches exercices/problèmes ne sera tolérée.**

En soutien des séances d'APP, 10 heures de **cours/Tds** sont prévues avec le tuteur, afin d'aider à la compréhension de la théorie. Les cours sont dispensés suivant l'approche **« classe inversée »** : les élèves étudient le cours et préparent les exercices d'application sur leur temps de travail individuel ; les séances servent aux questions et à la correction. Des exemples pratiques sous Matlab sont aussi effectués lors de ces séances. Il est indispensable de faire chaque semaine ce travail de préparation des cours/Tds pour en tirer un bénéfice.

Pour vous aider dans votre compréhension et maîtrise des concepts, des **quizz** et des **tests d'autoévaluation** sont disponibles sous Moodle. Ils servent aussi d'entrainement à l'examen individuel. Il est recommandé de faire individuellement ces tests à la fin de chaque séquence d'apprentissage.

1.3.2. Evaluation des compétences

Toutes les compétences listées dans la section 1.2 seront évaluées en continu lors des séances d'APP et l'évaluation reportée dans l'onglet signal du livret de compétences. Les **quatre compétences techniques** seront aussi évaluées à la fin de la composante via un **examen individuel** (QCM), dont l'objectif est de vérifier que les concepts fondamentaux sont acquis. Les séances complémentaires de cours/Tds ainsi que les **exercices d'entrainement proposés (Quizz et tests en ligne sous Moodle)** vous aideront aussi dans l'acquisition de ces notions.

Attention : un échec au QCM à l'évaluation de l'une des quatre compétences techniques implique systématiquement l'évaluation « non acquis » de la compétence, et ce quel que soit le travail réalisé en séances d'APP.

Enfin, une attention particulière sera donnée à la rédaction des livrables. L'évaluation de ces compétences sera également consignée dans le livret de compétences (onglets compétences générales E-S et onglet signal).

1.3.3. Planning général

Le tableau ci-dessous résume le planning général des travaux à effectuer durant les séances d'APP et de cours/TDs sur les 5 semaines. Ajouter le travail personnel de compréhension du cours et les exercices d'entrainement.

	SEANCE	TRAVAIL A EFFECTUER
SEMAINE 1	S1	Initiation à Matlab. Etude des signaux échantillonnés dans le domaine temporel (signaux déterministes et stochastiques). Détection et analyse d'un signal (puissance, corrélation). **Résolution des problèmes I-A et I-B.** **Livrables** : rapport sur la résolution des problèmes I-A et I-B, codes matlab.
	S2	
	S3	
	S4	
SEMAINE 2	S1	
	S2	
	S3	Etude des signaux dans le domaine fréquentiel. TFD des signaux déterministes. Analyse fréquentielle des signaux stochastiques. **Résolution du problème II**.
	S4	
SEMAINE 3	S1	Etude de l'échantillonnage, théorème de Shannon, sous-échantillonnage. Quantification. **Résolution des problèmes III-A et III-B.**
	S3	
	S3	
	S4	
SEMAINE 4	S1	Etude du filtrage numérique. **Résolution du problème IV.** **Livrables** : rapport sur la résolution des problèmes I-A, I-B, II, III-A, III-B, IV. code Matlab pour le problème IV.
	S2	
	S3	
	S4	
SEMANE 5	S1	Implémentation sur le microcontrôleur des deux fonctionnalités indispensables. Démonstration au tuteur. **EXAMEN INDIVIDUEL portant sur toutes les notions vues en cours, TD et séances d'APP.**
	S2	
	S3	
	S4	

FIG. 1- Planning de la composante signal

2. SIGNAUX ECHANTILLONNES

Nous étudions dans cette section les signaux échantillonnés dans le domaine temporel.

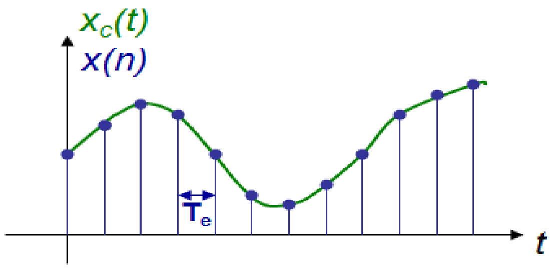


FIG. 2. Signal continu et signal échantillonné

Considérons un signal continu $x_c(t)$ fonction du temps t, la version échantillonnée (i.e. discrète) de ce signal est :

$$x(n) = x_c(nT_e)$$

T_e représente la période d'échantillonnage (en secondes) et $F_e = 1/T_e$ la fréquence d'échantillonnage (en Hz).

Autrement dit, on prélève à des instants réguliers les valeurs du signal continu. Les signaux échantillonnés sont donc représentés par un tableau de nombres, codant chacun l'amplitude du signal à un instant donné, avec, connue de manière sous-jacente, la période d'échantillonnage. Cette représentation est très pratique pour les traitements par ordinateur. On verra par la suite (Section 4.1), que, sous certaines conditions, les échantillons permettent de reconstruire tout le signal analogique (continu), c'est-à-dire de récupérer $x_c(t)$ à tout instant t. Le processus d'échantillonnage et de reconstruction sera étudié dans la section 4.

Dans le domaine temporel, on peut caractériser un signal par sa valeur moyenne, sa puissance, son énergie, sa valeur efficace (2.1). On peut comparer les puissances de deux signaux en introduisant les notions de gain et d'atténuation (2.2). Enfin, on peut mesurer une ressemblance entre deux signaux $x(t)$ et $y(t)$ grâce à la fonction de corrélation (2.3).

2.1. Valeur moyenne, énergie, puissance, valeur efficace

2.1.1. Signaux continus

Les définitions sont données pour un signal continu $x_c(t)$:

La **valeur moyenne** d'un signal, mesurée entre deux instants t_1 et t_2 est donnée par :

$$\bar{x}_c(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} x_c(t)\, dt$$

La **puissance instantanée** d'un signal $x_c(t)$, exprimée en Watt (W), est égale à son carré :

$$p(t) = x_c(t)^2$$

L'**énergie** d'un signal, exprimée en Joules (J), mesurée entre deux instants t_1 et t_2 est donnée par :

$$E(x, t_1, t_2) = \int_{t_1}^{t_2} x_c(t)^2 dt$$

La **puissance moyenne** dissipée entre ces deux instants est l'énergie dissipée divisée par la durée d'observation :

$$P(x, t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} x_c(t)^2 dt$$

La **valeur efficace** est la racine carrée de la puissance

$$A_{eff}(x, t_1, t_2) = \sqrt{\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} x_c(t)^2 dt}$$

2.1.2. Signaux échantillonnés

Pour les signaux échantillonnés, $x(n) = x_c(nT_e)$, les concepts restent les mêmes mais les intégrales sont transformées en sommes discrètes (approximation de Riemann). Par conséquent :

$$p(n) = x(n)^2$$

$$\bar{x}(n_1, n_2) = \frac{1}{n_2 - n_1 + 1} \sum_{n=n_1}^{n_2} x(n)$$

$$E(x, n_1, n_2) = T_e \sum_{n=n_1}^{n_2} x(n)^2$$

$$P(x, n_1, n_2) = \frac{1}{n_2 - n_1 + 1} \sum_{n=n_1}^{n_2} x(n)^2$$

$$A_{eff}(x, n_1, n_2) = \sqrt{\frac{1}{n_2 - n_1 + 1} \sum_{n=n_1}^{n_2} x(n)^2}$$

Signaux déterministes

Les signaux déterministes sont des signaux qui ont une forme mathématique connue. Par exemple, un forme d'onde sinusoïdale, une impulsion. Pour un signal périodique, on peut estimer la valeur moyenne, la puissance moyenne et la valeur efficace sur une période.

Signaux quelconques

Dans le cas général, les signaux générés par les capteurs ne sont pas déterministes et encore moins périodiques. Ils sont stochastiques : on ne peut pas calculer analytiquement leur valeur à chaque instant ; en revanche on peut les caractériser par des outils probabilistes ou statistiques.

Les **signaux stationnaires** sont des signaux dont les propriétés statistiques sont indépendantes du temps. Dans le cas général, les signaux n'ont pas cette propriété. Néanmoins, on peut souvent les considérer comme stationnaires sur des durées courtes. Par exemple, un signal de température ambiante est non stationnaire sur une durée de quelques heures mais quasi stationnaire sur une durée de 1mn, suffisamment longue pour estimer des propriétés statistiques.

Ainsi, pour un signal **quasi stationnaire**, on estimera sa moyenne et sa puissance sur des **fenêtres glissantes**, la taille de la fenêtre étant choisie de sorte que le signal puisse être considéré comme stationnaire sur cette durée.

La puissance d'un signal échantillonné $x(n)$ est estimée en chaque instant nT_e par

$$P(n) = \frac{1}{2K+1} \sum_{n-K}^{n+K} x(k)^2$$

avec $(2K+1)T_e$ la durée de la fenêtre temporelle pour l'estimation.

La puissance exprimée en **dBm** est la puissance rapportée à 1mW et exprimée en décibels :

$$P_{dBm}(n) = 10\, log_{10}\left(\frac{P(n)}{10^{-3}}\right)$$

2.2. Amplification/atténuation d'un signal, gain

Considérons un montage amplificateur dont le gain est g. L'amplitude du signal d'entrée est multipliée par un facteur g :

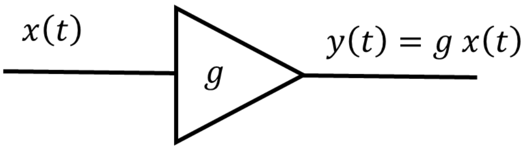


FIG. 3. Amplification

On définit le **gain** d'amplitude en décibels par :

$$g_{dB} = 20\log_{10}(g)$$

$$g = 10^{\frac{g_{dB}}{20}}$$

Par exemple, si $g = 100$, alors $g_{dB} = 40dB$; si $g = 0.1$ alors $g_{dB} = -20dB$. Un gain effectif $g \geq 1$ se traduit par une valeur positive ou nulle de g_{dB}. Une atténuation ($g < 1$) par une valeur négative de g_{dB}.

Soit P_x la puissance moyenne du signal x et P_y la puissance moyenne du signal x amplifié (de la forme $y(t) = g\,x(t)$). On calcule le **gain** de l'amplificateur en dB par :

$$g_{dB} = 10\log_{10}\left(\frac{P_y}{P_x}\right)$$

Bien noter le facteur 10 et non le facteur 20, quand on travaille sur des puissances, carrés de l'amplitude.

Soit $P_{x(dBm)}$ la puissance du signal x, exprimée en dBm, et g_{dB} le gain d'un amplificateur exprimé en dB. On calcule la puissance en dBm du signal amplifié par :

$$P_{y(dBm)} = P_{x(dBm)} + g_{dB}$$

2.3. Rapport signal à bruit

Le rapport signal à bruit (signal to noise ratio, SNR, en anglais) est le rapport entre la puissance du signal utile (i.e. qui contient l'information) et la puissance du bruit de fond, non significatif.

Soit P_s la puissance du signal utile et P_b la puissance du bruit, toutes les deux exprimées en Watt (W). Le rapport signal sur bruit, noté SNR, est une grandeur sans dimension définie par :

$$\mathrm{SNR} = \frac{P_s}{P_b}$$

Le rapport signal sur bruit, **exprimé en dB**, est défini par :

$$\mathrm{SNR}_{dB} = 10 \log_{10} \frac{P_s}{P_b}$$

Le bruit peut avoir des origines diverses : bruit thermique, perturbations électromagnétique, bruit engendré par le codage du signal, etc. Il est malheureusement toujours présent dans les signaux ! Le rapport signal à bruit permet de caractériser la qualité d'un signal. Par exemple, un signal audio de qualité CD a un bien meilleur rapport signal à bruit qu'un signal audio mp3 de fort facteur de compression.

2.4. Corrélation

La corrélation est une méthode de comparaison de deux signaux dans le domaine temporel.

Soient deux signaux $x(t)$ et $y(t)$. L'**intercorrélation** de $x(t)$ et $y(t)$ est la fonction définie par :

$$C_{x,y}(\tau) = \int_{-\infty}^{+\infty} x(t)\, y(t+\tau) dt$$

L'intercorrélation est donc une fonction du temps. C'est une mesure de ressemblance. Elle prend une valeur maximale pour les décalages τ pour lesquels les signaux $x(t)$ et $y(t+\tau)$ se « superposent » au mieux (FIG. 4).

Soit le signal $x(t)$. L'**autocorrélation** de $x(t)$ est la fonction définie par :

$$C_x(\tau) = \int_{-\infty}^{+\infty} x(t)\, x(t+\tau) dt$$

L'intercorrélation et l'autocorrélation se traduisent pour les signaux discrets en remplaçant les intégrales par des sommes discrètes :

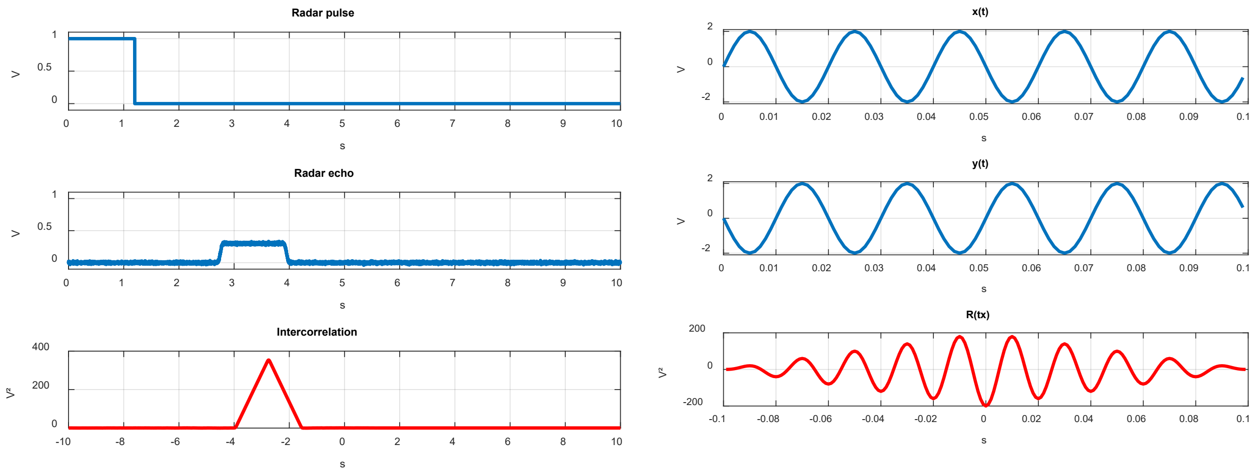
Soient deux signaux discrets $x(nT_e)$ et $y(nT_e)$ échantillonnés à la fréquence $f_e = \frac{1}{T_e}$.

La fonction d'**intercorrélation** de $x(nT_e)$ et $y(nT_e)$ est

$$C_{x,y}(kT_e) = \sum_{n=-\infty}^{+\infty} x(nT_e)\, y\big((n+k)T_e\big)$$

La fonction d'**autocorrélation** de $x(t)$ est la fonction définie par :

$$C_x(kT_e) = \sum_{n=-\infty}^{+\infty} x(nT_e)\, x\big((n+k)T_e\big)$$



(a) (b)

FIG. 4. *Fonction d'intercorrélation. (a) le maximum de la fonction d'intercorrélation indique le décalage temporel entre l'impulsion d'un radar et l'écho reçu. (b) Intercorrélation pour deux fonctions sinusoïdales de période $T_0 = 0.02s$ et déphasées de π (demi-période).*

En pratique, on ne peut observer le signal que sur une durée finie *D*. Dans ce cas, les intégrales sont calculées entre deux instants t_1 et t_2. Généralement, on normalise le calcul de corrélation par la durée d'observation, $t_2 - t_1$. Pour un signal discret, on dispose d'un nombre fini *N* d'échantillons et la normalisation consistera à diviser par *N* :

$$C_{x,y}(kT_e) = \frac{1}{N}\sum_{n=0}^{\mathrm{N}-1} x(nT_e)\, y\big((n+k)T_e\big)$$

2.5. EXERCICES

EXERCICE 2-1

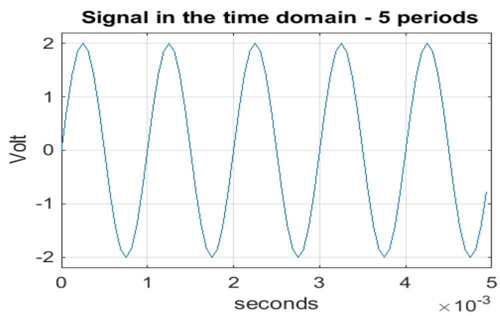
Prise en main de Matlab.

1. Générer un signal sinusoïdal x_1 de fréquence $f_1 = 1KHz$ de durée $D = 50ms$ et d'amplitude $A_1 = 1V$, avec un fréquence d'échantillonnage $F_e = 20KHz$. Afficher le signal avec une échelle appropriée (commande axis). Ajouter légendes et titre (title, xlabel, ylabel).
2. Extraire les périodes 5 à 10 de x_1 et appeler ce nouveau signal y. Afficher y sur le même graphe (:, hold on, find, min). Ajouter une légende (legend).
3. Générer un deuxième signal sinusoïdal x_2 de fréquence $f_2 = 100Hz$ et d'amplitude $A_2 = 0.3V$. Ajouter un offset de 2V. Afficher ce nouveau signal.
4. Faire la multiplication des signaux x_1 et x_2. Comprendre la différence entre le calcul matriciel et le calcul élément par élément avec la notation '.'. Montrer sur cet exemple comment les opérations sur les matrices permettent de supprimer des boucles for.

EXERCICE 2-2

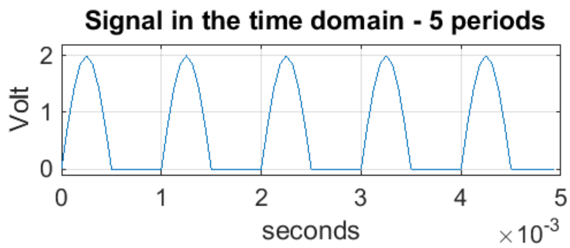
1. Calculer la valeur T_0 de la période, la valeur moyenne, l'énergie, la puissance moyenne et la valeur efficace des signaux analogiques suivants :

$$x(t) = A\sin(2\pi f_0 t) \text{ avec } f_0 = 1KHz$$



$$x_2(t) = \begin{cases} x(t) & si\ 0 \le mod(t, T_0) < \dfrac{T_0}{2} \\ 0 & \text{sinon} \end{cases}$$

$$\Leftrightarrow x_2(t) = \max(x(t), 0)$$

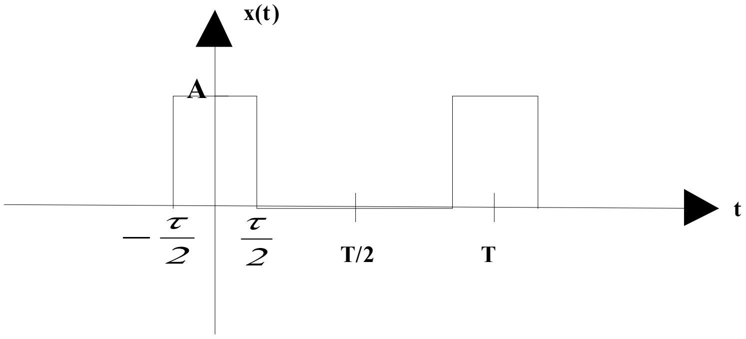


2. On échantillonne ces signaux à la fréquence $f_e = 16KHz$. Donner l'expression temporelle de chacun d'entre eux. Indiquer la valeur de l'échantillon $n = 347$. A quel instant a-t-il été prélevé ? Combien y a-t-il d'échantillons par période ?

3. Donner la valeur exacte ou une approximation des quantités calculées en 1, après échantillonnage des signaux (à faire sous Matlab).

EXERCICE 2-3

Soit le signal continu constitué d'un train d'impulsions de durée τ et de période T. On prendra $T = 20\mu s$, $\tau = 4\mu s$ et $A = 2.5V$.



1. Calculer la valeur moyenne de ce signal
2. Calculer la puissance moyenne de ce signal en Watt et en dBm
3. Même question si on ajoute une composante continue $A_0 = 0.5\ V$

EXERCICE 2-4

1. On mesure la puissance d'un signal Wifi reçu sur un smartphone : $Pr = -63\ dBm$. Quelle est la puissance du signal reçu en Watt ?
2. On mesure la puissance du signal émis par la borne Wifi : $P_e = 2W$. Quelle est la puissance émise en dBm ?
3. Quelle est l'atténuation du signal entre l'émetteur Wifi et le récepteur, en valeur linéaire et en dB ?

EXERCICE 2-5

Soit deux signaux échantilonnés x_n et y_n avec

- $x_n = y_n = 0$ pour $n < 0$
- $x_0 = 1;\ x_1 = 2; x_2 = 1;\ x_n = 0$ pour $n > 2$
- $y_0 = 0;\ y_1 = 0; y_2 = 3; y_3 = 5; y_4 = 3; y_5 = 0;\ y_n = 0$ pour $n > 5$

Calculer la corrélation entre les deux signaux. Interpréter.

EXERCICE 2-6

On considère deux signaux $x_1(t) = A_1 \cos(2\pi f_1 t)$ et $x_2(t) = A_2 \cos(2\pi f_2 t + \varphi)$.

1. Calculer l'intercorrélation $C_{x_1,x_2}(0)$ de ces deux signaux, estimée sur l'intervalle temporel $[0, D]$. On normalisera le résultat par la durée D.
2. Etudier le cas particulier où $f_2 = f_1$. On prendra $D = \frac{n}{f_1}, n \in N$.
3. Sous Matlab, **programmer** l'intercorrélation $C_{x_1,x_2}(\tau)$ (avec la normalisation) et retrouver les résultats. On prendra :
   - $f_1 = 500Hz,\ f_2 = 550Hz,\ A_1 = 0.5V\ ,\ A_2 = 1.5V,$
   - fréquence d'échantillonnage $F_e = 10KHz$.
   - $\tau \in \left[0, \frac{2}{f_1}\right]$
   - $\varphi = 0\ rad$
4. Prendre $f_2 = f_1$ et faire varier φ. Interpréter.

2.6. Travail en séances d'APP

Prise en main de Matlab

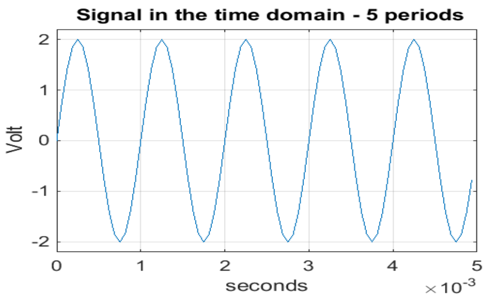
Se reporter au tutoriel.

Exercices sous Matlab

A. Créer un signal sinusoïdal discret $x(nT_e) = A \sin(2\pi f_0 n T_e)$ dont les caractéristiques sont les suivantes :

- Fréquence du sinus $f_0 = 1\ KHz$, période T_0
- Amplitude $A = 2V$
- Fréquence d'échantillonnage $F_e = \frac{1}{T_e} = 16\ KHz,$
- Durée : $D = 2s.$

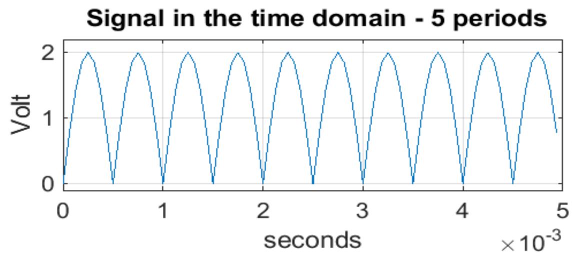
Afficher les 5 premières périodes. Bien indiquer les échelles sur les deux axes. La figure ci-contre montre le graphe que vous devez obtenir.



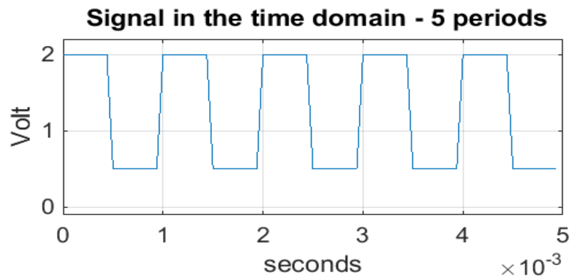
B. Faire une estimation de la valeur moyenne, de la puissance moyenne (en W et en dBm) et de la valeur efficace de ce signal (échantillonné), en choisissant correctement la durée de l'estimation. Comparer avec les valeurs trouvées dans les exercices. Augmenter la fréquence d'échantillonnage. Conclure.

C. Refaire le même travail sur les signaux suivants :

$$x_3(t) = |x(t)|$$



$$x_4(t) = \begin{cases} 2 & si\ 0 \leq mod(t, T_0) < \frac{T_0}{2} \\ 0.5 & \text{sinon} \end{cases}$$



D. Générer un signal y discret, échantillonné également à $F_e = 16\ KHz$:
$y(nT_e) = B\sin(2\pi f_0 nT_e + \varphi)$ tel que

- La puissance du signal y est égale à 32 dBm
- $\varphi = \frac{\pi}{3} rad$

E. Programmer l'autocorrélation du signal x et l'intercorrélation des signaux x et y. Afficher les résultats et interpréter.

F. Estimer la puissance instantanée des signaux de parole (*BonneJournee.wav*) et de musique (*JBTheme.wav*, *Allegretto.wav*). Ces signaux peuvent être chargés à l'aide de la commande `audioread`. Utiliser l'aide en ligne pour connaître les paramètres d'entrée et de sortie. Attention, certains signaux peuvent être en stéréo et il faut alors les ramener à un signal mono, par moyennage des canaux gauche et droite (`mean`). Ecouter les signaux à l'aide de la fonction `sound`. Justifier la longueur de la fenêtre définie pour chaque type de signal.

Problèmes à résoudre

Résoudre les problèmes I-A et I-B et commencer à rédiger le premier livrable.

Problème I-A : Proposer un algorithme qui permet de détecter la présence/absence d'un signal audio. On supposera que les 50 premières millisecondes du signal ne contiennent pas de signal utile (parole, musique…) mais uniquement le bruit ambiant. Appliquer votre algorithme sur les signaux *BonneJournee.wav*, *Atone.wav*, *Allegretto.wav*, *JBTheme.wav*. Justifier <u>tous</u> les paramètres.

Problème I-B : Proposer un algorithme qui permet de comparer les fréquences de deux sons audio purs (donc de forme $A\cos(2\pi f t + \varphi)$).

On considèrera des fréquences dans l'intervalle [130Hz,4000Hz] avec une fréquence d'échantillonnage de 16KHz. Les paramètres des signaux (amplitude, fréquence et phase) seront générés aléatoirement. Votre algorithme doit renvoyer un résultat binaire : 1 si les deux fréquences sont les mêmes, à une marge d'erreur près, 0 sinon.

<u>Optionnel</u> : la fréquence de référence doit prendre aléatoirement une les valeurs discrètes gamme tempérée sur les octaves de l'intervalle fréquentiel spécifié (FIG. 5). On vérifiera l'égalité des fréquences à ½ ton près.

NB : les problèmes I-A et I-B sont des premiers éléments de résolution de la fonction FT1-1-3-2 qui sera finalisée dans le problème IV

Fréquences des notes (en hertz) dans la gamme tempérée

Note/octave	-1	0	1	2	3	4	5	6	7	8	9
do ou *si*♯	16,35	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01	8 372,02	16 744,04
do♯ ou *ré*♭	17,33	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92	8 869,84	17 739,68
ré	18,36	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64	9 397,28	18 794,56
ré♯ ou *mi*♭	19,45	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03	9 956,06	19 912,12
mi ou *fa*♭	20,60	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04	10 548,08	21 096,16
fa ou *mi*♯	21,83	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65	11 175,30	22 350,60
fa♯ ou *sol*♭	23,13	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91	11 839,82	23 679,64
sol	24,50	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93	12 543,86	25 087,72
sol♯ ou *la*♭	25,96	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88	13 289,76	26 579,52
la	27,50	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00	14 080,00	28 160,00
la♯ ou *si*♭	29,14	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	7458,62	14 917,24	29 834,48
si ou *do*♭	30,87	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13	15 804,26	31 608,52

FIG. 5. Fréquences des notes dans le tempérament égal.
Le passage d'une note à la note suivante, distante d'un demi ton, se fait par multiplication par $2^{\frac{1}{12}}$.

Livrables

Commencer à rédiger le premier livrable (voir le planning en **FIG. 1**). **Vérifier les consignes données** dans la section 1.2 de l'introduction de ce document (page 6 et suite).

3. ANALYSE FREQUENTIELLE

Les signaux peuvent être étudiés dans le domaine fréquentiel, en calculant la transformée de Fourier.

3.1. Transformée de Fourier d'un signal continu

Considérons un signal continu $x_c(t)$. La **transformée de Fourier** et la **transformée de Fourier inverse** sont données par les équations suivantes :

$$TF\{x_c(t)\} = X_c(f) = \int_{-\infty}^{+\infty} x_c(t)\, e^{-2\pi jft}\ dt$$

$$TF^{-1}\{X_c(f)\} = x_c(t) = \int_{-\infty}^{+\infty} X_c(f)\, e^{+2\pi jft}\ df$$

La variable *f* représente la **fréquence** exprimée en **Hertz (Hz)**. La transformée de Fourier est biunivoque. Elle permet une interprétation des signaux dans le domaine fréquentiel. $x_c(t)$ and $X_c(f)$ sont deux manières différentes de représenter la même information. La transformée de Fourier inverse donne la signification physique de l'analyse de Fourier : elle montre que tout signal peut être vu comme la combinaison linéaire de fonctions exponentielles complexes $e^{+2\pi jft}$ de fréquences f différentes ; le coefficient $X_c(f)$ pondère chacune de ces fonctions élémentaires.

Pour les signaux réels (non complexes), la propriétés $X_c(f) = X_c^*(-f)$ implique que tout signal réel peut se décomposer comme une combinaison linéaire de fonctions *cos* et *sin* de fréquences différentes. De plus, pour les signaux réels périodiques, de période $T_0 = 1/f_0$, on peut montrer que les fréquences possibles (i.e. $f\ /\ |X_c(f)| \neq 0$) sont uniquement les multiples de la fréquence fondamentale f_0, ce qui fait le lien avec la décomposition en série de Fourier.

Le résultat de la transformée de Fourier est complexe. On calcule donc le module et l'argument par :

$$X_c(f) = A_c(f)e^{j\varphi_c(f)} \text{ avec } \begin{cases} A_c(f) = |X_c(f)| = \sqrt{Re\{X_c(f)\}^2 + Im\{X_c(f)\}^2} \\ \varphi_c(f) = \tan^{-1}\left(\frac{Im\{X_c(f)\}}{Re\{X_c(f)\}}\right) \end{cases}$$

où $Re\{z\}$ et $Im\{z\}$ désignent la partie réelle et la partie imaginaire du nombre complexe z.

Généralement, on s'intéresse au module de la transformée de Fourier, exprimé en dB. Il représente la répartition énergétique en fonction de la fréquence (voir sections 3.3 et 3.4).

$$|X_c(f)|_{dB} = 20\ \log_{10}(|X_c(f)|)$$

3.2. Transformée de Fourier discrète

Le passage de la transformée de Fourier « analogique » à la Transformée de Fourier Discrète (TFD) se fait (1) en remplaçant la variable t par nT_e et l'intégrale par une somme discrète, (2) en discrétisant l'espace des fréquences. On rappelle ces étapes dans ce qui suit.

Soit $x(n) = x_c(nT_e)$ un signal discret échantillonné à la fréquence $F_e = \frac{1}{T_e}$.

(i) La transformée de Fourier de ce signal discret est :

$$X(f) = T_e \sum_{n=-\infty}^{+\infty} x(n) e^{-2\pi j f n T_e}$$

Noter que *f* est une variable continue et définie sur l'intervalle $[0, F_e]$[1] ou, au choix, $\left[-\frac{F_e}{2}, +\frac{F_e}{2}\right]$.

(ii) Considérant maintenant une suite discrète de *N* échantillons ($n \in [0, N-1]$), l'équation précédente devient :

$$X(f) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-2\pi j f n T_e}$$

(iii) Enfin, on discrétise l'espace des fréquences en posant $f = k\frac{F_e}{N}$, $k \in [0, N-1]$:

$$X\left(k\frac{F_e}{N}\right) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-2\pi j k \frac{F_e}{N} n T_e}$$

Ainsi :

La **Transformée de Fourier Discrète (TFD)** d'une suite de *N* échantillons ($n \in [0, N-1]$) est définie par :

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-2\pi j \frac{nk}{N}}$$

où $X(k)$, $k \in [0, N-1]$, représente la transformée de Fourier à la fréquence $f = k\frac{F_e}{N}$

La **Transformée de Fourier Discrète Inverse (TFDI)** est donnée par :

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{+2\pi j \frac{nk}{N}}$$

On a donc discrétisé l'espace des fréquences $[0, F_e[$. Le **pas d'échantillonnage** est égal à $\Delta f = \frac{F_e}{N}$, c'est aussi la précision fréquentielle de la TFD. On remarque donc que plus on a d'échantillons (i.e. plus *N*

[1] Ce point sera précisé dans la partie relative à l'échantillonnage. On peut déjà remarquer que la transformée de Fourier discrète est périodique de période F_e, i.e. $X(f + F_e) = X(f)$, et que, par conséquent, la connaissance de $X(f)$ sur l'intervalle $[0, F_e[$ est suffisante.

est grand), plus la TFD est précise. En pratique, on augmente la précision d'une TFD en ajoutant des échantillons égaux à 0 en bout de signal : cette opération est appelée « zero-padding » en anglais. On augmente artificiellement le nombre de points sans changer le contenu spectral du signal.

L'algorithme rapide qui permet de calculer la TFD est appelé **FFT** (**Fast Fourier Transform**). La fonction Matlab correspondante est la fonction `fft`.

3.3. Signaux stochastiques et densité spectrale de puissance

La transformée de Fourier est calculable analytiquement pour les signaux déterministes ayant une forme mathématique connue. En revanche, la formule n'est pas applicable pour les signaux stochastiques dont on ne connaît pas la forme analytique. On doit donc décrire la structure fréquentielle du signal d'un point de vue statistique. Un descripteur très utilisé est la **densité spectrale de puissance (DSP)**, notée $S_x(f)$. Cette quantité mesure la puissance moyenne du signal en la fréquence f ; elle est donc exprimée en W/Hz. Ainsi, intégrée sur tout le domaine, la densité spectrale de puissance conduit à la puissance totale du signal (**théorème de Parseval**).

Soit le signal $x(t)$ de densité spectrale de puissance $S_x(f)$. La puissance du signal $x(t)$ est donnée par :

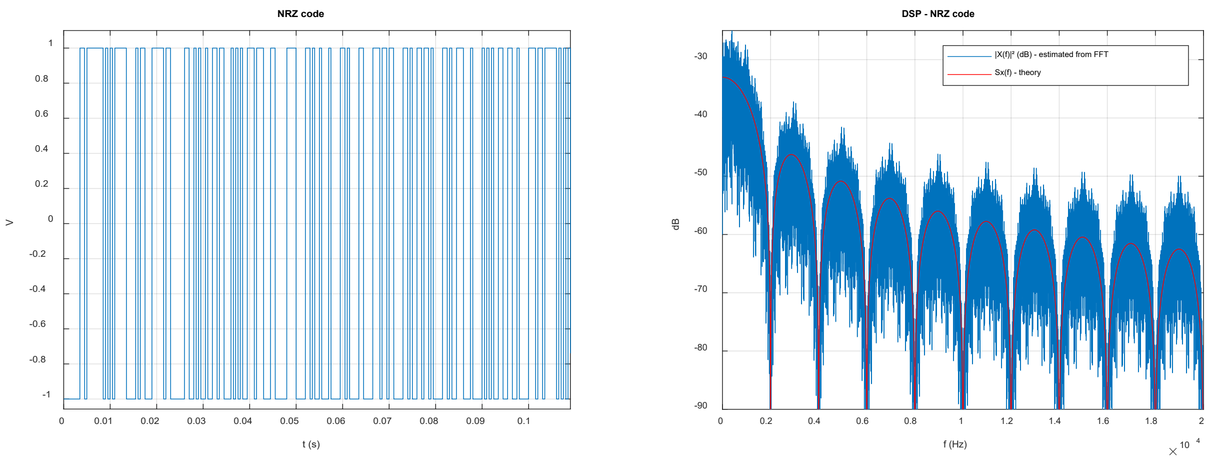
$$P_x = \int_{-\infty}^{+\infty} S_x(f)\, df$$

Théorème de Wiener-Khintchine :

Soit le signal $x(t)$. La densité spectrale de puissance $S_x(f)$ du signal $x(t)$ est définie comme la transformée de Fourier de sa fonction autocorrélation $C_x(\tau)$:

$$S_x(f) = TF\{C_x(\tau)\} = \int_{-\infty}^{+\infty} C_x(\tau)\, e^{-2\pi j f \tau} d\tau$$

En communications numériques, ce théorème permet de calculer mathématiquement l'encombrement spectral des signaux transmis : en effet, si la forme d'onde est connue analytiquement (par exemple, bit 1 codé par un créneau et bit 0 codé par un créneau inversé, FIG. 6a), le message numérique qui la module est, quant à lui, aléatoire (succession aléatoire de 1 et de 0). Le signal transmis est donc de nature stochastique et c'est le théorème de Wiener-Khintchine qui permet d'obtenir la forme mathématique de sa densité spectrale de puissance (FIG. 6b, courbe rouge), et donc la connaissance de la bande passante nécessaire à sa transmission. Nous ne développerons pas ce point dans le cadre de cette APP : ce théorème n'est mentionné ici que pour faire le pont avec les notions fondamentales de communications numériques.



(a) (b)

FIG. 6. *Estimation de la densité spectrale de puissance (DSP) d'un signal NRZ ; (a) signal dans le domaine temporel : un bit 1 est codé par un créneau de 1V, un bit 0 par un créneau de -1V, avec un débit de 2kbits/s; (b)Densité spectrale de puissance (DSP) : estimation par le module au carré de la TFD du signal (a) et courbe théorique calculée par le théorème de Wiener Khintchine.*

En pratique, comment estimer la densité spectrale de puissance d'un signal échantillonné, $x(nT_e)$, $n = 0, \cdots, N-1$?

Un **estimateur de la densité spectrale de puissance** (DSP) d'un signal $x(nT_e)$ est le module de sa transformé de Fourier Discrète (TDF) au carré. La représentation de la DSP en fonction de la fréquence f est appelée **spectre du signal**.

3.4. Analyse spectrale

Supposons que l'on veuille estimer expérimentalement la densité spectrale de puissance d'un signal de N échantillons $x(nT_e)$= $x(n)$. On applique une TFD (en pratique une FFT),

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)\, e^{-2\pi j\frac{kn}{N}},$$

puis on calcule le module au carré $|X(k)|^2$ de chaque coefficient complexe obtenu. Chaque $|X(k)|^2, k = 0, \dots, N-1$, est une estimée de la puissance du signal à la fréquence $k\frac{F_e}{N}$, l'unité étant le W/Hz. La représentation graphique des $|X(k)|^2$ en fonction de la fréquence $k\frac{F_e}{N}$ montre le spectre sur l'intervalle $\left[0, \frac{N-1}{N}F_e\right]$, donc grosso-modo $[0, F_e[$. On peut en déduire dans quelles bandes de fréquence le signal a de l'énergie (**encombrement spectral**). Le plus souvent, on adopte une représentation en décibels en calculant le spectre en dB :

Spectre du signal $x(n), n \in [0, N-1]$ en dB :

$$X(k) = TFD\{x(n)\}, k \in [0, N-1]$$

$$10log_{10}(|X(k)|^2) = 20log_{10}(|X(k)|)$$

On verra dans la section 4 que le spectre des signaux réels (à valeurs dans $\mathbb{R}$) échantillonnés doit être interprété sur l'intervalle $\left[0, \frac{F_e}{2}\right]$. En effet :

- Le spectre est périodique de période F_e: voir la formule de la TFD qui implique $X(k+N) = X(k)$.
- Un signal ne peut être correctement échantillonné à la fréquence F_e que s'il ne possède pas d'énergie aux fréquences supérieures à $\frac{F_e}{2}$ (théorème d'échantillonnage de Shannon, voir section 4.1): il n'y a donc pas lieu de regarder au-delà de $\frac{F_e}{2}$.
- Pour les signaux réels , on a $|X(-k)| = |X(k)|$; autrement dit, le spectre est symétrique par rapport à la fréquence 0.

Pour toutes ces raisons, seul l'intervalle $\left[0, \frac{F_e}{2}\right]$ a un sens physique, tout le reste de l'axe fréquentiel se déduit par symétrie et recopie, comme le montre la figure ci-dessous.

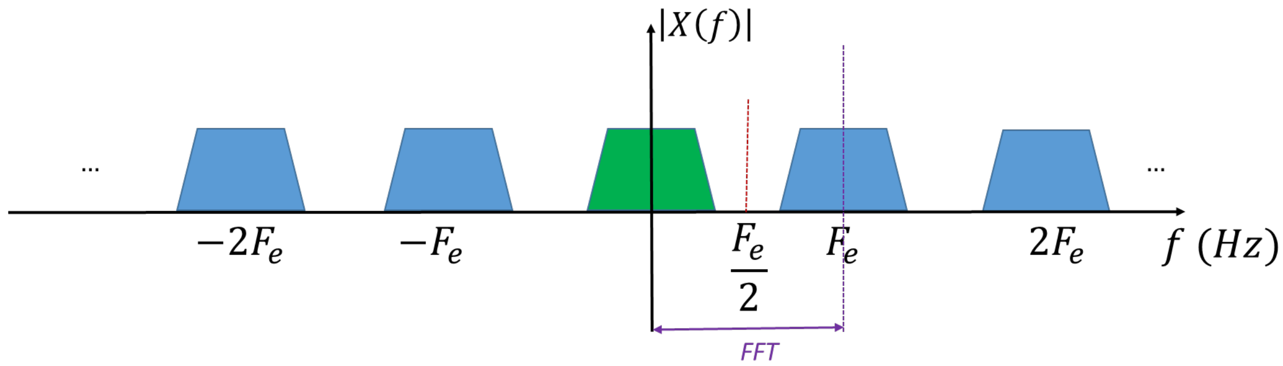


FIG. 7. *Interprétation spectrale d'un signal échantillonné à la fréquence F_e. En vert le spectre du signal analogique réel $x(t)$, symétrique par rapport à $f = 0Hz$. En vert et bleu le spectre du signal échantillonné $x(nT_e)$, qui correspond à la périodisation du spectre du signal $x(t)$. La FFT donne le spectre entre 0 et F_e. A cause de la symétrie et de la périodisation, seul l'intervalle $[0, F_e/2]$ doit être pris en compte pour l'interprétation.*

Dans la FIG. 6, la DSP du signal a été représentée sur l'intervalle $\left[0, \frac{F_e}{2}\right]$ $(F_e = 40\ KHz)$.

3.5. EXERCICES

EXERCICE 3-1

On rappelle les propriétés suivantes :

- $\int_{-\infty}^{+\infty} e^{-2\pi jft}\, dt = \delta(f)$
- $\int_{-\infty}^{+\infty} e^{2\pi jft}\, df = \delta(t)$

où δ dénote l'impulsion de Dirac.

5. Calculer la transformée de Fourier d'un signal constant d'amplitude A.
6. Calculer la transformée de Fourier du signal $x(t) = Ae^{2\pi jf_0t}$.
7. Calculer la transformée de Fourier du signal $Asin(2\pi f_0 t)$.
8. Calculer la transformée de Fourier du signal $Acos(2\pi f_0 t)$.

Pour chacune des fonctions précédentes, représenter graphiquement le module de la TFD et donner la signification physique du résultat trouvé.

EXERCICE 3-2

Sous Matlab, calculer la transformée de Fourier d'un signal sinusoïdal x_1 (`fft`) :

- Fréquence $f_1 = 500Hz$
- Amplitude $A_1 = 2V$
- Fréquence d'échantillonnage $F_e = 8KHz$
- Durée : $D = 50ms$.

1. Afficher le module de la fft en dB, en fonction de la fréquence (attention à bien définir le vecteur des fréquences sur l'axe des abscisses). Quel est le pas Δf de la fft ? Expliquer la forme du résultat obtenu.
2. Ajouter un signal sinusoïdal de fréquence $f_2 = 600Hz$ et d'amplitude $A_2 = 1V$. Commenter le résultat obtenu.

EXERCICE 3-3

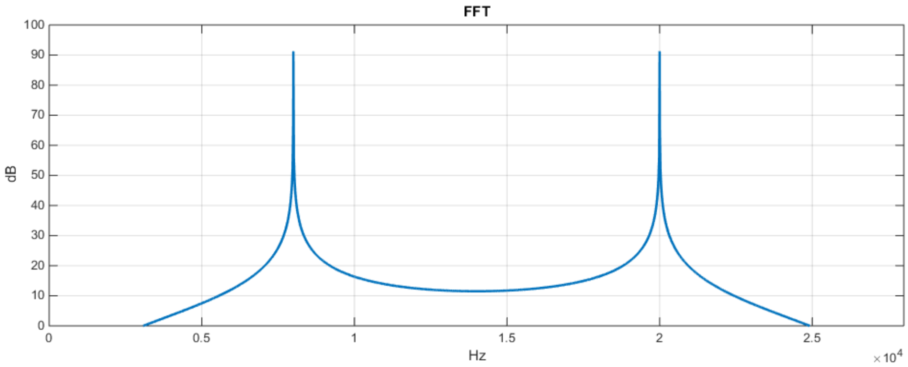
Calculer la transformée de Fourier d'une porte d'amplitude A et de durée τ :

$$x(t) = \begin{cases} A\ si\ t \in [0, \tau] \\ 0\ sinon \end{cases}$$

Relier la première annulation de la transformée de Fourier à la durée τ de la fonction porte.

EXERCICE 3-4

On a calculé le module de la transformée de Fourier discrète d'un signal discret $x(nT_e)$ échantillonné à la fréquence $F_e = 28\ KHz$. La figure suivante montre le résultat obtenu :



Donner l'expression temporelle du signal $x(nT_e)$ en précisant les valeurs des fréquences.

EXERCICE 3-5

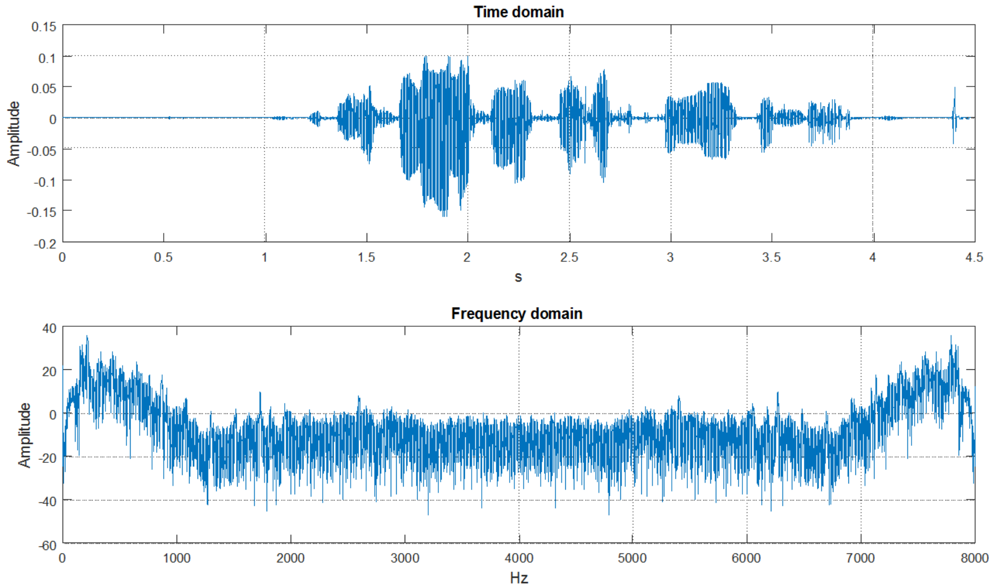
Soit le signal numérique constitué des échantillons : $x_0 = 2, x_1 = 3, x_2 = 5, x_3 = 3, x_4 = 2$. Tous les autres échantillons sont nuls.

1. Calculer la transformée de Fourier discrète de ce signal. Le résultat est-il réel ?
2. Montrer que la TFD prend des valeurs réelles si on translate temporellement le signal de *-2T*, *T* étant la période d'échantillonnage.

EXERCICE 3-6

On a calculé la transformée de Fourier discrète (TFD) d'un signal audio $x(nT_e), n = 0,1,\cdots,N-1$ avec $N = 1024$. La TFD a été calculée sur les N échantillons, sans zéro-padding. La fréquence d'échantillonnage est $F_e = \frac{1}{T_e} = 8\ KHz$. La figure ci-dessous montre le module de la TFD en dB : $10log\left(\left|X\left(k\frac{F_e}{N}\right)\right|^2\right)$

1. Combien de coefficients $\left|X\left(k\frac{F_e}{N}\right)\right|$ a-t-on obtenu ? Quel est le pas d'échantillonnage de l'axe fréquentiel ? Que représente le 151ème coefficient $\left|X\left(k\frac{F_e}{N}\right)\right|^2$? Quelle est la grandeur mesurée et son unité ?
2. Quelle est l'encombrement spectral de ce signal ?
3. Quel est le rapport signal à bruit en dB ?



3.6. Travail en séance d'APP

Transformée de Fourier Discrète de signaux déterministes

A. Soit le signal :

$$y(n) = \begin{cases} 1V \text{ si } 0 \leq nT_e \leq 5ms \\ 0V \text{ sinon} \end{cases}$$

Générer ce signa sous Matlab et afficher sa FFT. On prendra une fréquence d'échantillonnage $F_e = 8KHz$. Relier le résultat obtenu au résultat théorique de l'EXERCICE 3-3.

Translation en fréquence

B. Créer deux signaux sinusoïdaux de fréquences $f_1 = 1440Hz$ et $f_2 = 2000Hz$, de durée $D = 1s$. On choisira une fréquence d'échantillonnage $F_e = 8000Hz$. Multiplier les deux signaux et faire l'analyse spectrale du résultat. Justifier le résultat obtenu par le calcul.

Analyse fréquentielle de signaux stochastiques

On travaillera dans cette partie sur des signaux audio.

C. Faire une analyse fréquentielle de sons de musique (1 seule note) pour plusieurs instruments : fichiers piano.wav, flute.wav, flute2.wav, violon.wav, violon2.wav. Pour chaque fichier, indiquer la note qui a été jouée. Caractériser qualitativement le timbre des trois instruments. Que pensez-vous du choix de la fréquence d'échantillonnage dans les différents fichiers son ?

D. Charger le fichier *2notes.wav*. Faire l'analyse fréquentielle de ce signal. Que peut-on en déduire ?

Avec la méthode précédente, l'information temporelle est totalement perdue. Pour une analyse plus fine, on calcule la FFT sur des fenêtres glissantes (spectrogramme). La taille de la fenêtre est choisie de telle sorte que le signal puisse être considéré comme stationnaire sur la durée d'observation.

E. Programmer le calcul de la FFT sur des fenêtres glissantes sans recouvrement, en choisissant correctement la durée des fenêtres. Faire l'analyse spectrale du fichier *2notes.wav*.

Problème à résoudre

Problème II : Proposer **deux algorithmes** permettant d'estimer le rythme cardiaque à partir des signaux ECG stockés dans les fichiers 100.dat, 101.dat, …, et 109.dat (Fonctionnalité FT1-1-2-1 du cahier des charges). Le résultat de ces algorithmes doit être une fréquence cardiaque mesurée au cours du temps (pas un résultat moyen sur tout le fichier .dat).

Appliquer les algorithmes proposés sur <u>tous</u> les signaux (une méthode qui fonctionne sur un seul cas n'a aucun intérêt !).

Discuter les différentes approches possibles en termes de précision de l'analyse, de robustesse et de complexité algorithmique. Indiquer quelle approche est à favoriser pour une implémentation sur le microcontrôleur et quelle approche conduit aux résultats les plus robustes et les plus précis.

NB : Utiliser la fonction Open_dat.m pour charger les signaux stockés dans les fichiers .dat.

Remarque 1 : éviter les heuristiques fondées sur la définition de multiples paramètres, par exemple pour détecter des maxima. Il est possible de résoudre ce problème sans aucune définition de seuil !

Livrables

Transmettre le premier livrable conformément au planning (**FIG. 1**). **Vérifier le respect des consignes données** dans la section 1.2 de l'introduction de ce document (page 6 et suite). Transmettre à votre tuteur, par mail, le rapport <u>au format pdf</u> avec les programmes Matlab associés.

Commencer à rédiger le second livrable (problème II). Ce livrable devra aboutir à une discussion comparative sur les différentes méthodes possibles en termes de précision de l'analyse, de robustesse et de complexité algorithmique.

4. NUMERISATION D'UN SIGNAL

Nous avons directement travaillé sur des signaux échantillonnés. Or la plupart des capteurs délivrent des signaux analogiques. Il est temps maintenant de s'interroger sur la méthode de numérisation des signaux. Comment doit-on procéder pour numériser correctement un signal de telle sorte que ce signal puisse être restitué sans distorsion ? Quelle est l'influence des différents paramètres sur la qualité du signal ?

La numérisation d'un signal repose sur deux étapes : l'**échantillonnage** et la **quantification**. L'échantillonnage consiste à prélever la valeur du signal analogique à des instants régulièrement espacés. La quantification consiste à coder les amplitudes des échantillons sur un nombre fixé de valeurs prédéfinies et représentables par des nombres entiers.

4.1. Echantillonnage

Formellement, l'échantillonnage correspond à la multiplication du signal continu $x_c(t)$ par un peigne de Dirac $W_{T_e}(t)$ de période T_e (FIG. 8) :

$$x_e(t) = W_{T_e}(t) \,.\, x_c(t)$$

On montre alors que, dans le domaine fréquentiel, cela correspond à une périodisation du spectre (FIG. 8) :

$$X_e(f) = \frac{1}{T_e} \sum_{k=-\infty}^{+\infty} X_c\left(f - \frac{k}{T_e}\right) = F_e \sum_{k=-\infty}^{+\infty} X_c(f - kF_e)$$

Le spectre du signal échantillonné est périodique de période F_e, où F_e est la fréquence d'échantillonnage.

Dans la suite, on notera la suite des échantillons $x(n) = x_e(nT_e) = x_c(nT_e)$ ou encore x_n, pour désigner l'échantillon prélevé à l'instant nT_e.

La question est de savoir s'il y a perte d'information lors de l'opération d'échantillonnage, ou, autrement dit, s'il est possible ou non, à partir des échantillons x_n, de reconstruire intégralement et sans erreur le signal continu $x_c(t)$ pour tout *t* (FIG. 9) ?

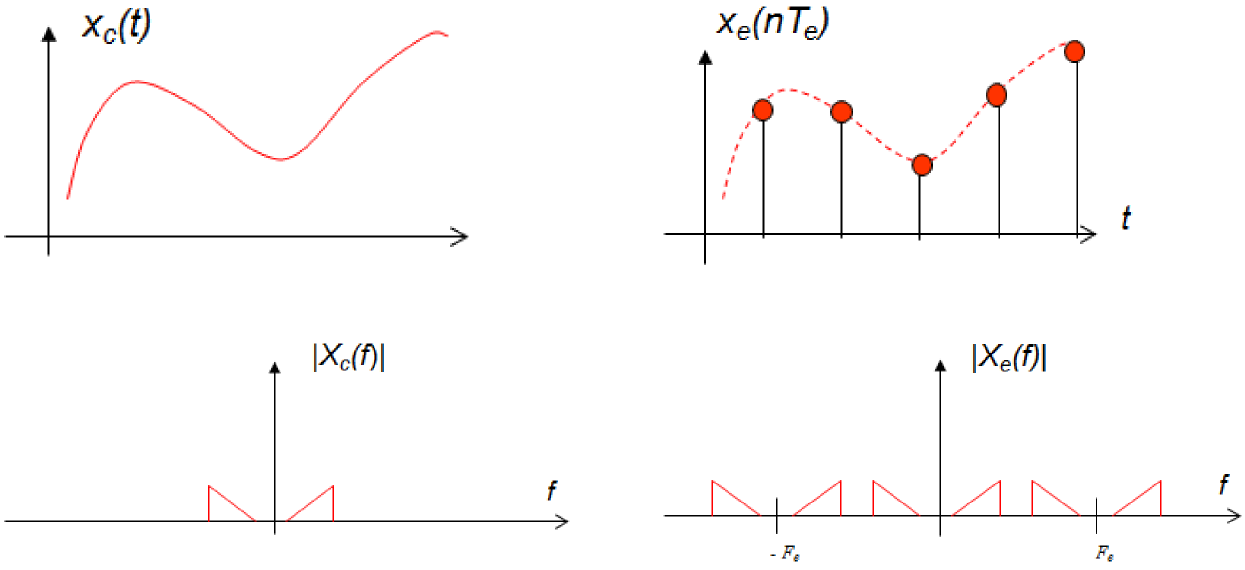


FIG. 8. Principe de l'échantillonnage

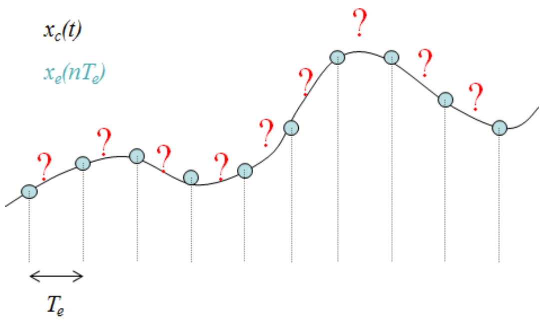


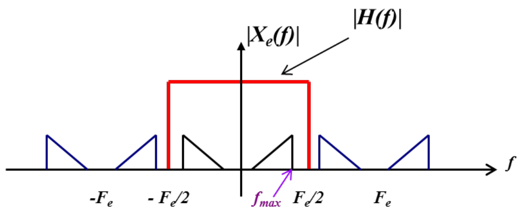
FIG. 9. Problématique de reconstruction du signal continu $x_c(t)$ à partir des échantillons $x_n = x_e(nT_e)$

La réponse a été apportée par Shannon, ingénieur mathématicien (1916-2001), qui a ouvert la voie vers le numérique, en énonçant « **le théorème d'échantillonnage de Shannon** ». Considérant un signal continu, dont l'occupation spectrale est bornée et comprise entre 0 et f_{max}, le théorème est le suivant :

Théorème d'échantillonnage

Un signal qui ne comporte pas de composantes à des fréquences supérieures ou égales à une valeur f_{max} est entièrement déterminé par la suite de ses valeurs à des instants régulièrement espacés d'une durée $T_e = 1/F_e$ à condition d'avoir $F_e \geq 2f_{max}$.

Ce théorème se comprend très bien dans le domaine fréquentiel. Rappelons que le spectre du signal échantillonné est le spectre du signal continu auquel s'ajoutent des recopies centrées sur les multiples de la fréquence d'échantillonnage. Si on filtre le spectre du signal échantillonné avec un filtre passe-bas $H(f)$[2] qui supprime toutes les recopies du spectre et ne garde que la partie autour de *f=0*, alors on retrouve le spectre du signal continu et donc $x_c(t)$ (puisque la transformée de Fourier est biunivoque). Mais pour cela, il faut qu'il n'y ait pas de **repliement de spectre** (**aliasing** en anglais), c'est-à-dire que $F_e \geq 2f_{max}$. La FIG. 10 illustre cela.

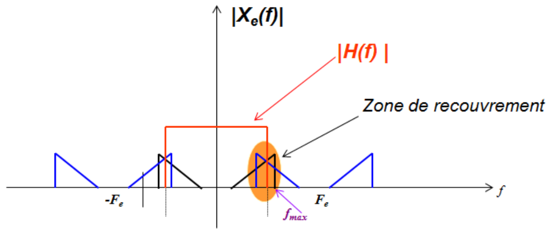


Le théorème de Shannon est respecté puisque $F_e \geq 2f_{max}$

Il n'y a pas de superposition des recopies du spectre $X_c(f)$ (partie en noire sur la figure) et on peut restaurer $X_c(f)$ en filtrant $X_e(f)$ avec un filtre passe-bas $H(f)$ tel que :

$$H(f) = \begin{cases} \frac{1}{F_e} & \text{si } f \in \left[-\frac{F_e}{2}, \frac{F_e}{2}\right] \\ 0 \text{ sinon} & \end{cases}$$

Le signal $x_c(t)$ est parfaitement reconstruit dans ce cas



Le théorème de Shannon n'est pas respecté : $F_e < 2f_{max}$.

Il n'est pas possible de réaliser un filtre qui permet de séparer les recopies du spectre (en bleu) du spectre initial (en noir).

Dans ce cas, on ne pas restaurer $X_c(f)$ donc $x_c(t)$ sans distorsion.

FIG. 10. Illustration de l'aliasing et de la reconstruction du signal continu $x_c(t)$ à partir des échantillons $x_e(nT_e)$

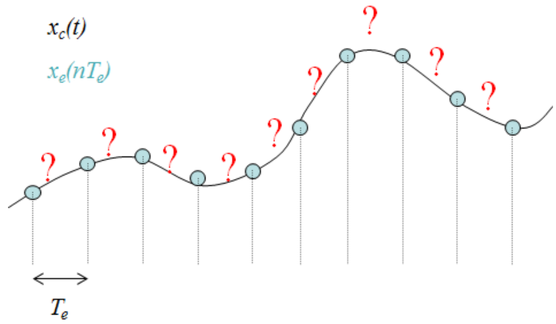
4.2. Interpolation

Lorsqu'on exprime la fonction de filtrage passe-bas $H(f)$ dans le domaine temporel, on aboutit à la formule d'interpolation :

[2] La notion de filtrage est abordée dans la section 5. Un filtre est un système linéaire qui supprime certaines fréquences dans le signal mis à son entrée.

$$\hat{x}_c(t) = \sum_{-\infty}^{+\infty} x(n) \frac{\sin\left(\pi\left(\frac{t}{T_e} - n\right)\right)}{\pi\left(\frac{t}{T_e} - n\right)}$$

avec $x(n) = x_e(nT_e)$



Dans l'équation précédente, $\hat{x}_c(t)$ désigne le signal analogique reconstruit en tout instant *t*. En pratique, on ne dispose que de signaux de durée finie, c'est-à-dire une suite finie d'échantillons $x(n)$, $n \in [0, N-1]$. On approxime donc la formule d'interpolation en considérant les échantillons disponibles autour de l'instant *t* considéré. L'approximation la plus simple est l'approximation linéaire.

4.3. Quantification

La seconde étape de la numérisation d'un signal est la quantification. Les échantillons sont approximés par des valeurs discrètes multiples du **pas de quantification** *q*. Chaque valeur discrète (*k.q*, k entier) est codée par un nombre entier représenté sur *b* bits. Il y a donc $N = 2^b$ niveaux de quantification. La figure FIG. 11 illustre cette étape pour un signal $x_e(nT_e)$ dont l'amplitude varie entre –*A*/2 et +*A*/2. Le signal quantifié est noté $x_q(nT_e)$.

Le **débit binaire** d'un signal numérique échantillonné à la fréquence F_e et codé sur b bits est :

$$D = F_e.b \text{ (bits/s)}$$

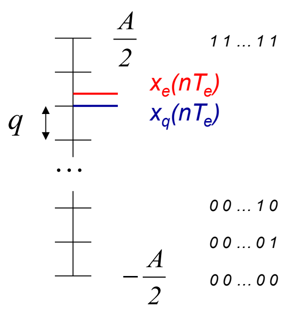


FIG. 11. Quantification d'un signal échantillonné

La quantification des échantillons conduit à une erreur puisque les échantillons sont approximés sur une grille de valeurs discrètes (FIG. 12). On crée donc un bruit appelé **bruit de quantification**. Le bruit de quantification est la différence $e(nT_e) = x_e(nT_e) - x_q(nT_e)$.

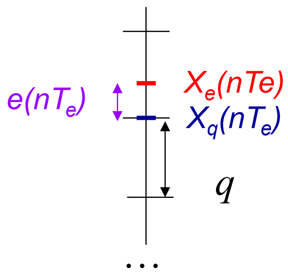


FIG. 12. Bruit de quantification

La puissance moyenne du bruit de quantification est

$$P_e = \frac{q^2}{12} \text{ (W)}$$

4.4. En pratique…

Numériser un signal implique donc de définir une fréquence d'échantillonnage, compte-tenu de son contenu spectral, et de définir un nombre de bits de quantification, compte-tenu du rapport signal à bruit de quantification admissible. Il faut aussi prendre en compte le débit final admissible pour l'application visée. Noter que l'échantillonnage est toujours précédé d'un filtre anti-repliement, de fréquence de coupure inférieure à $F_e/2$, afin de garantir que les conditions du théorème de Shannon sont respectées.

4.5. EXERCICES

EXERCICE 4-1

On considère le signal défini par :

$$x(t) = A_0 + A_1 \sin(2\pi f_0 t) + A_2 \sin(4\pi f_0 t),$$

avec $f_0 = 1000 Hz, A_0 = 1, A_1 = 2, A_2 = 0.5$

1. Calculer la transformée de Fourier de ce signal.

On échantillonne le signal x*(t)* à la fréquence F_e*=5000 Hz*.

2. Le théorème de Shannon est-il respecté ?
3. Calculer les 5 premiers échantillons.

4. Tracer le spectre du signal échantillonné.

EXERCICE 4-2

Soit le signal défini par $x_0 = Acos(2\pi f_0 t)$, avec $f_0 = 750$ Hz. On prélève des échantillons de ce signal tous les $T = 0.1$ ms.

1. Donner l'expression du signal échantillonné
2. Donner l'expression de la transformée de Fourier du signal échantillonné. Tracer le spectre.
3. Comment reconstruire le signal ? Le système satisfait-il au critère de Shannon ?

On change la période d'échantillonnage, qui devient $T = 10^{-3}$ s.

4. Tracer le nouveau spectre. Le théorème de Shannon est-il toujours respecté ?

On filtre le signal échantillonné par un filtre passe-bas idéal de fréquence de coupure $f_c = 500$ Hz.

5. En raisonnant dans le domaine spectral, écrire l'expression du signal en sortie du filtre.
6. Comment expliquer cela dans le domaine temporel ? Faire un schéma explicatif…

EXERCICE 4-3

Soit un signal sinusoïdal d'amplitude $A = 1V$, de fréquence $f_0 = 1KHz$, échantilllonngé à la fréquence $F_e = 16KHz$. $b = 8$ bits.

1. Calculer le rapport signal à bruit de quantification.
2. Vérifier sous Matlab. Pour quantifier le signal, on appliquera la formule suivante :

$$x_q = \frac{round\left(x_e . 2^{(b-1)}\right)}{2^{(b-1)}}$$

Justifier cette formule.

EXERCICE 4-4

On considère un signal musical stéréo échantillonné à la fréquence $f_e = 44$ KHz et numérisé sur 16 bits.

1. Quel est le débit ?
2. Quelle quantité de données, en octets, représente 30 minutes de musique ?

Afin de réduire la quantité de données, on numérise sur 12 bits.

3. Quelle est la variation du rapport signal à bruit de quantification ?

EXERCICE 4-5

Soit $x(nT)$ un signal échantillonné à la fréquence $f_e = 1$ MHz. Afin de réduire la quantité de données à stocker, on souhaite que le signal soit échantillonné à $f_e' = 250$ KHz.

Indiquer les traitements qui permettent de passer du signal échantillonné à $f_e = 1$ MHz au signal échantillonné à $f_e' = 250$ KHz.

4.6. Travail en séances d'APP

Echantillonnage : étude sur un cas d'école

Générer un signal sinusoïdal s_1 dont les caractéristiques sont les suivantes :

- Fréquence du sinus $f_0 = 440Hz$, période T_0

Amplitude $A = 1V$

- Fréquence d'échantillonnage $F_{e1} = 8\,KHz$
- Durée : 0.1 seconde.

A. Afficher le signal s_1 dans le domaine temporel, ainsi que le module de la FFT, que l'on calculera sur $M = 2^{11}$ points. Interpréter.

Créer maintenant le même signal sinusoïdal, mais en prenant une fréquence d'échantillonnage $F_{e2} = 0.5\,KHz$. On notera s_2 ce second signal.

B. Afficher le signal s_2 dans le domaine temporel, ainsi que le module de la FFT, toujours calculé sur $M = 2^{11}$ points. Que remarquez-vous ?

C. Superposer les deux signaux sur un même graphique, dans le domaine temporel. Tracer le signal s_1 par un trait continu, mais n'afficher que les échantillons du signal s_2 (`plot(…,…,'mo')`) Comment interprétez-vous ce graphique ? Quelle serait le signal sinusoïdal de fréquence
$f_3 < F_{e2}/2$ qui permettrait de relier les échantillons de s_2 ?

D. Si on déséchantillonne le signal s_2 par un filtre interpolateur de fréquence de coupure $F_{e2}/2$, que récupèrerait-on ? Est-ce satisfaisant ?

E. Conclure sur le théorème de Shannon et sa signification physique.

Sous-échantillonnage (décimation)

Charger le signal de parole *Pi_C_96K.wav* en utilisant la fonction `audioread`.

F. Quelle est la fréquence d'échantillonnage ? Ecoutez-le signal grâce à la fonction *sound.m*.

G. Tracer le spectre de ce signal. Quelle est l'occupation spectrale de ce signal ?

H. Sous-échantillonner le signal (i.e. décimer), en prenant un échantillon sur $k = 6$. Afficher le signal temporel. Afficher également le spectre. Ecouter ce nouveau signal. Que remarque-t-on ?

I. Faire varier le facteur de décimation *k*. Quelle est la plus grande valeur acceptable, pourquoi ?

J. Avant de sous-échantillonner, que faut-il faire pour éviter le repliement de spectre ?

Quantification

K. Charger le signal vocal *BonneJournee.wav*. Ce signal, noté x_e, a une amplitude variant entre -1 et 1 et est codé dans le fichier sur 16bits. Diminuer le nombre de bits de quantification($b \in [1,16]$) et tracer le rapport signal à bruit de quantification, en fonction du nombre de bits $\left(SNR(b)\right)$ et en fonction du débit $\left(SNR(D)\right)$. Tracez également l'erreur de quantification et écoutez le signal quantifié.

Pour quantifier le signal sur *b* bits (dont 1 bit de signe), on appliquera la formule suivante :

$$x_q = \frac{round\left(x_e . 2^{(b-1)}\right)}{2^{(b-1)}}$$

- Justifier la formule de quantification.
- Commenter l'évolution de l'erreur en fonction du nombre de bits
- Commenter et justifier la courbe *SNR(b)* ou *SNR(D)*.
- Quelle est l'influence du nombre de bits de codage sur la qualité d'un signal ?

L. Conclure sur la numérisation des signaux : choix de la fréquence d'échantillonnage et du nombre de bits de quantification.

Résolution de problèmes

Problème III-A : On souhaite effectuer un enregistrement numérique en haute fidélité, avec une bande de fréquences allant de 20Hz à 20 000Hz, et un rapport signal à bruit moyen de 90dB. On suppose que le micro lui-même est idéal (bande infinie, aucune distorsion, aucun bruit). Il enregistre les signaux avec une dynamique de $[-500mV, +500mV]$. La puissance moyenne en sortie du micro est de 30mW.

Donner le schéma fonctionnel de numérisation de ce signal, déterminer tous les paramètres et calculer le débit du signal numérique. Quelle capacité de stockage faut-il pour enregistrer une heure de concert en stéréo ? Les paramètres choisis sont-ils compatibles avec une qualité et un stockage sur CD ? Permettent-ils de respecter le timbre des instruments d'un orchestre symphonique ?

Réfléchir sur l'utilité d'adopter un pas de quantification non pas uniforme mais qui suit une loi logarithmique.

Problème III-B : Synthétiser la gamme chromatique (3^{e} octave, FIG. 5) par la méthode d'échantillonnage à partir de la note donnée dans les fichiers son Fl_XN_96K.wav (flûte traversière), Pi_XN_96K (piano), Vi_XN_96K (Violon), etc. Sur quel intervalle fréquentiel cette méthode vous donne-t-elle des résultats corrects ? Pourquoi ?

Livrable

Continuer la rédaction du second livrable (problèmes III-A et III-B).

5. FILTRAGE NUMERIQUE

Les filtres sont des systèmes linéaires qui permettent de séparer les composantes spectrales d'un signal, en laissant passer certaines fréquences et en en atténuant d'autres. Les applications sont très nombreuses :

- en télécommunications, séparer des signaux émis sur des bandes fréquentielles différentes,
- en audio, égaliseurs pour modifier le timbre des sons,
- En imagerie, pour lisser les images ou au contraire accentuer les contours,
- …

5.1. Fonction de transfert

On raisonne donc dans le domaine fréquentiel pour réaliser des filtres. Un filtre est caractérisé par sa **fonction de transfert** $H(f)$ qui transforme le signal d'entrée $X(f)$ en un signal de sortie $Y(f)$. Le module$|H(f)|$ de la fonction de transfert représente le gain du filtre en fonction de la fréquence. Si ce gain est égal à 1, la fréquence est transmise, si le gain est égal à 0, la fréquence est coupée.

Fonction de transfert d'un filtre :

$$H(f) = \frac{Y(f)}{X(f)}$$

On appelle **bande passante** d'un filtre une plage de fréquences transmises sans atténuation par le filtre. On appelle **bande atténuée ou bande coupée**, une plage de fréquences atténuées par le filtre (FIG. 13).

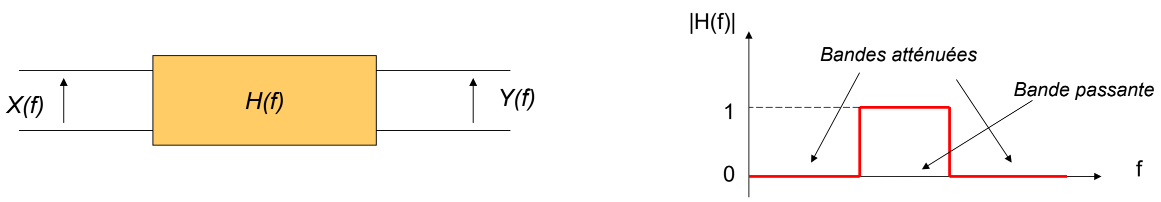


FIG. 13. Filtre de fonction de transfert $H(f)$

5.2. Principaux filtres

On distingue quatre catégories de filtres, schématisés dans la FIG. 14 :

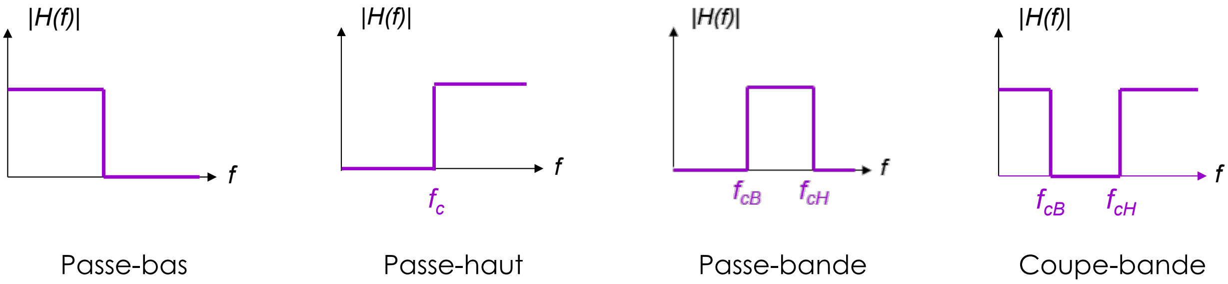


FIG. 14. Principaux filtres

Les filtres numériques sont simples à réaliser car ils ne reposent que sur 3 opérations : l'addition, la multiplication et la fonction mémoire.

Dans le domaine temporel, les filtres **RIF** (dits à **réponse impulsionnelle finie**) sont définis par une équation de la forme :

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i)$$

où les coefficients $h(i), i \in [0, N-1]$ sont les N coefficients (nombres réels) du filtre numérique d'ordre N. Cette équation est appelée **équation aux différences**.

Dans le domaine fréquentiel, la **fonction de transfert** du filtre est donnée par :

$$H(f) = \frac{Y(f)}{X(f)} = \sum_{i=0}^{N-1} h(i)\, e^{-2\pi j i f T_e}$$

avec $T_e = 1/F_e$ la période d'échantillonnage.

La synthèse d'un filtre consiste à se donner un gabarit, spécifiant les bandes passantes et atténuées, et à en déduire les coefficients $h(i), i \in [0, N-1]$ du filtre. Nous n'abordons pas cet aspect dans le cadre de cette introduction, nous nous focalisons sur la spécification du gabarit d'un filtre, en fonction des contraintes du problème posé. Nous utiliserons les outils de synthèse de Matlab.

5.3. Gabarits et synthèse

Les filtres idéaux présentés dans la FIG. 14 ne sont pas synthétisables en pratique. C'est pourquoi, on introduit la notion de gabarit, qui spécifie une « plage » dans laquelle le module de la fonction de transfert doit passer.

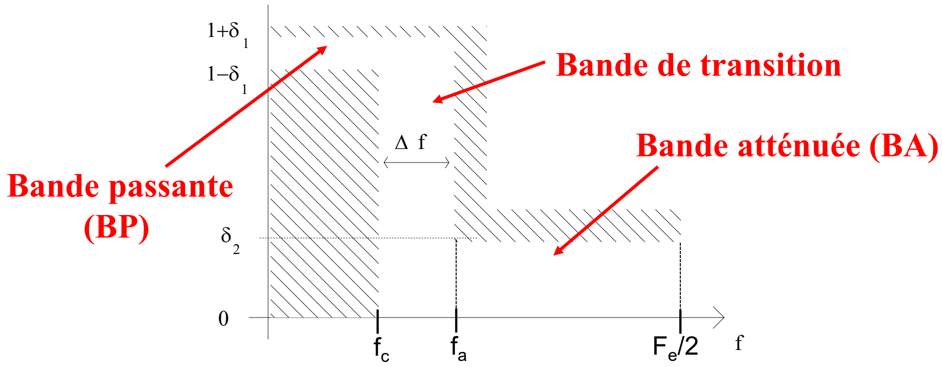


FIG. 15. Gabarit d'un filtre numérique passe-bas

Les paramètres suivants définissent le **gabarit** du filtre passe-bas :

f_c : **fréquence de coupure** (la **bande passante** du filtre est l'intervalle $[0, f_c]$)

f_a : fréquence définissant le début de la **bande atténuée** ($\left[f_a, \frac{F_e}{2}\right]$)

δ_1 : erreur par rapport au gain idéal 1 dans la bande passante

δ_2 : erreur par rapport au gain idéal 0 dans la bande atténuée

$\Delta f = f_a - f_c$: largeur de la **bande de transition**.

$A_{max} = 20\log_{10}(1 + \delta_1)$: **ondulation en bande passante**, exprimée en dB

$A_{min} = 20\log_{10}(\delta_2)$: **réjection** garantie en bande atténuée, exprimée en dB

Ainsi, avec ce gabarit, on garantit un gain de 1 en bande passante, à une erreur δ_1 près, une atténuation de A_{min} (dB) en bande atténuée. Plus l'ordre du filtre (i.e. *N*, le nombre de coefficients) est élevé, plus la bande de transition est étroite et les erreurs δ_1 et δ_2 par rapport aux gains idéaux (respectivement 1 et 0) sont faibles. On se rapproche alors du filtre passe-bas idéal.

5.4. EXERCICES

<u>EXERCICE 5-1</u>

Considérant un filtre RIF d'équation

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i),$$

démontrer que sa fonction de transfert est

$$H(f) = \frac{Y(f)}{X(f)} = \sum_{i=0}^{N-1} h(i)\, e^{-2\pi j i f T_e}.$$

EXERCICE 5-2

Soit le système défini par l'équation aux différences :

$$y(n) = 0.25x(n) + 0.5x(n-1) + 0.25x(n-2).$$

1. Quelle est la réponse impulsionnelle de ce système ?
2. Quelle est la réponse au signal d'entrée : $\begin{cases} x_0 = 1, x_1 = 2.5, x_2 = -1, x_3 = 0, x_4 = 1.5, x_5 = -0.2 \\ x_n = 0 \text{ pour tout } n \notin [0,5] \end{cases}$
3. Quelle est la réponse en fréquence du système. De quel type de filtre s'agit-il ?
4. Représenter le système par un schéma. Quelles sont les 3 opérations de base impliquées ?

EXERCICE 5-3

On souhaite réaliser la détection de la note de musique « SI » de fréquence f_{SI}=*494Hz*. Les notes de musique dont les fréquences sont les plus proches sont le « LA# » de fréquence $f_{LA\#}$=*466 Hz* et le « DO » de fréquence f_{DO}=*523Hz*. Le musicien n'est pas nécessairement bien accordé, et la fréquence des notes jouées peut varier de 2Hz par rapport à leur valeur théorique.

On envisage pour cela de concevoir un filtre et de comparer les puissances en entrée et en sortie du filtre, pour en déduire la présence ou l'absence de la note « SI » dans le signal audio.

1. Quel type de filtre faut-il concevoir ?
2. Proposer un gabarit tenant compte des informations numériques fournies. Spécifier <u>toutes</u> les valeurs définissant ce gabarit.
3. Faire la conception de ce filtre sous Matlab. Voir l'influence des paramètres sur l'ordre du filtre obtenu.

5.5. Travail en séance d'APP

A. Synthèse d'un filtre. Charger le fichier *notesFlute.wav* et réaliser un filtre FIR qui permet de sélectionner la première harmonique d'un SOL médium (392 Hz). Tester aussi avec le fichier notesPiano.wav. Quel type de filtre concevez-vous, pourquoi ? Quel est l'ordre du filtre obtenu ? Optimiser les paramètres du gabarit pour que le son restitué soit un SOL pur sans résidu d'autres notes. Utiliser la fonction *sound* pour écouter le fichier source et le signal filtré. La procédure de design d'un filtre est décrite ci-dessous.

Sous la fenêtre de commande de Matlab, taper *sptool*. Cliquer ensuite sur « New » dans la colonne « Filters ». Vous obtenez alors la fenêtre de design de filtres (FIG. 16).

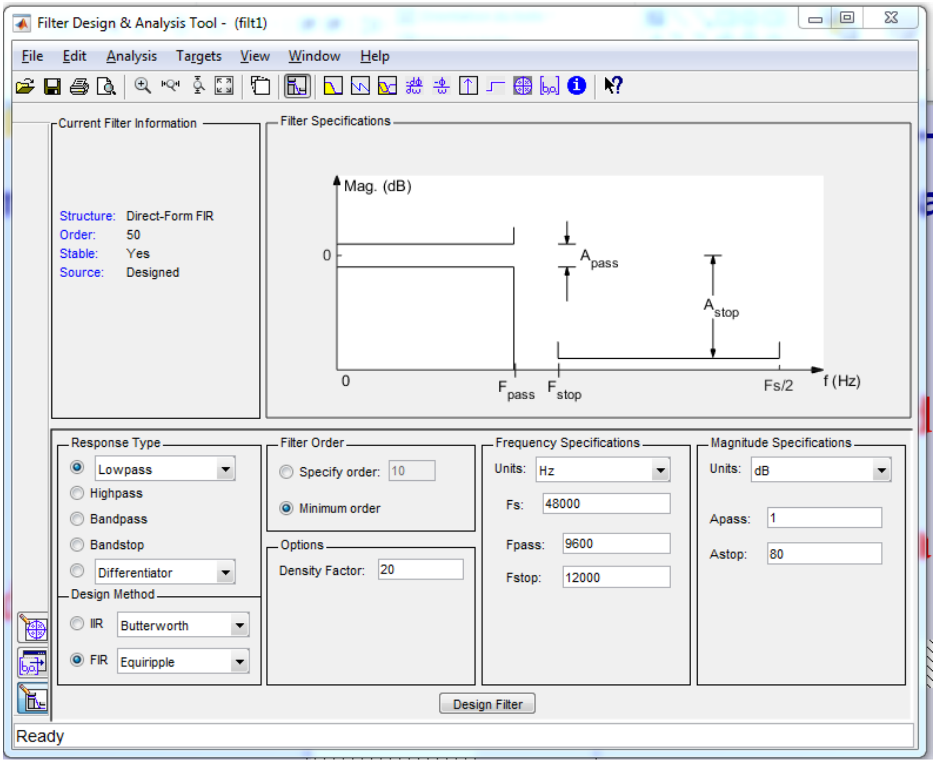


FIG. 16. Conception d'un filtre sous sptool

Cette fenêtre vous permet de spécifier le gabarit, suivant la méthodologie suivante :

1. Indiquer le type de filtre (Response Type)

2. Garder la méthode de design par défaut (FIR (Finite Impulse Response), Equiripple)

3. Spécifier l'ordre du filtre : garder « Minimum order » afin de minimiser le nombre d'opérations à réaliser lors de l'application du filtre.

4. Indiquer la fréquence d'échantillonnage « Fs » : c'est celle du signal à filtrer.

5. Indiquer ensuite les paramètres du gabarit, « Fpass », « Fstop », « Apass », « Astop ».

En pratique, on choisit une valeur de « Astop » de l'ordre du rapport signal à bruit, puis une valeur de « Apass » correspondante (i.e. $\delta_1 = \delta_2$).

Cliquer ensuite sur « Design Filter ». Lorsque le résultat vous convient, exporter le filtre dans un fichier .mat, chargeable à partir d'un fichier de commande : cliquer sur File / Export. Indiquer le nom du vecteur des coefficients du filtre (*h* par exemple, FIG. 17) et cliquer sur « export ». Donner ensuite un nom à votre fichier, par exemple *filtre.mat*.

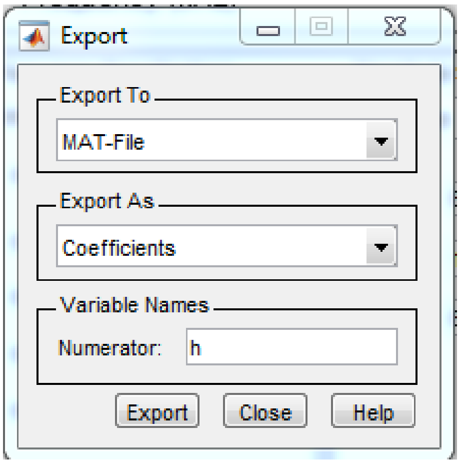


FIG. 17. Sauvegarde des coefficients d'un filtre sous sptool

Dans votre fichier de commande, vous pouvez alors charger le vecteur *h* des coefficients du filtre par la commande *load 'filtre.mat'*. La fonction de filtrage d'un signal est la fonction *filter.m*. Utiliser l'aide en ligne de Matlab pour connaître la syntaxe précise.

B. Est-ce qu'un SOL est présent dans l'accord du fichier *AccordsPiano.wav* ?

C. Combien d'opérations doit-on réaliser pour traiter un échantillon ?

D. Comment baisser l'ordre du filtre sans altérer la qualité de détection ?

Résolution de problème

Problème IV : Proposer un algorithme, qui permet de réaliser la fonctionnalité de mesure de la qualité de reconnaissance d'une tonalité (FT1-1-3-2, **FIG. 18**). Le programme de simulation devra réaliser les fonctions suivantes :

- générer un signal sonore sinusoïdal puis carré correspondant à une note de musique de la gamme à tempérament égal, dans les octaves 2 à 6 (FIG. 5). Il s'agit du signal de référence dont la tonalité est à reproduire.
- simuler la réception d'un signal sonore sinusoïdal bruité et d'amplitude variable, dont la durée est d'au moins 2 secondes et dont le début est aussi aléatoire (pas nécessairement au temps $t = 0s$). Il s'agit du signal reproduit par l'utilisateur.
- réaliser la détection de la partie utile du signal reçu et vérifier que sa fréquence est bien dans la bande spécifiée.
- Analyser l'écart fréquentiel entre la note de référence et la note reproduite, et renvoyer un score correspondant à la qualité de reproduction de la note jouée. Réfléchir à différentes approches compte-tenues des contraintes

d'implémentation matérielle, de complexité algorithmique, de robustesse et de précision des résultats.

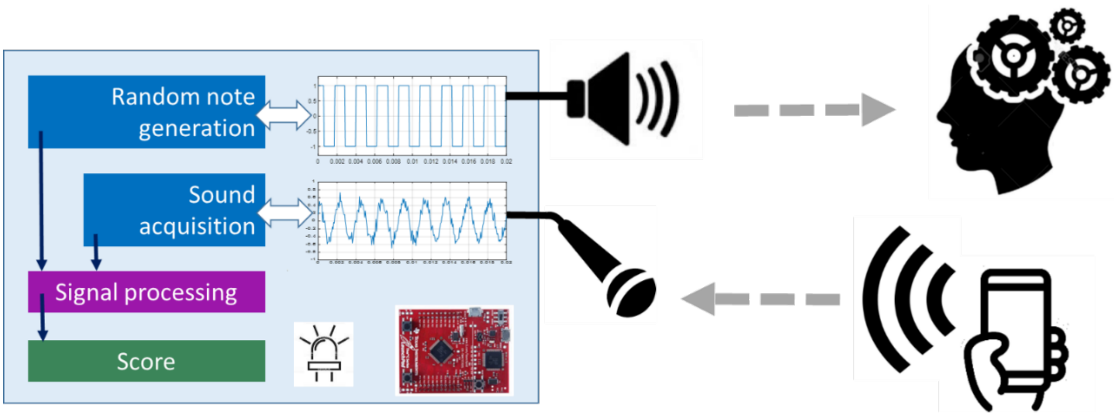


FIG. 18. Schéma synoptique du système de mesure de la qualité de reconnaissance d'une tonalité (FT1-1-3-2). *Le système génère une note sous la forme d'un signal carré, dont la fréquence est tirée aléatoirement dans l'ensemble des fréquences de la gamme à tempérament égal. L'utilisateur écoute ce signal et tente de générer une note pure (i.e. signal sinusoïdal) de même fréquence avec son smartphone. Ce signal est acquis par le micro et le système de traitement du signal effectue la comparaison entre la fréquence qu'il a générée et la fréquence du signal reçu. Il renvoie une réponse binaire (1 si la fréquence est bien reproduite à une marge près) et un score qui reflète plus finement la qualité de la reproduction de la fréquence.*

Instructions complémentaires pour la rédaction du rapport sur le problème IV

En plus des instructions générales habituelles, le rapport devra contenir :

- un schéma fonctionnel du système proposé,
- une discussion argumentée sur les différentes approches possibles et les raisons des choix effectués.
- La justification de tous les paramètres du programme (fréquences d'échantillonnage, seuils, gabarits de filtres, etc.)
- des résultats de simulation avec une estimation de la précision des mesures,
- une estimation de la complexité de l'algorithme en termes d'opérations +/* à réaliser.

Le programme sera implémenté sur le microcontrôleur une fois validé sous Matlab.

Livrables

Transmettre à votre tuteur le second livrable conformément au planning (**FIG. 1**). Le rapport doit prendre en considération les instructions générales données en introduction de ce document, les instructions complémentaires ci-dessous, ainsi que les retours donnés par votre tuteur sur le premier rapport. Tous les problèmes et exclusivement ceux-ci doivent être traités, un problème par section.

Le rapport est à transmettre au tuteur par mail au format pdf, avec le code Matlab correspondant au problème IV.

6. IMPLEMENTATION MATERIELLE

L'objectif est maintenant de réaliser l'implémentation matérielle des fonctionnalités étudiées sur le microcontrôleur. Au minimum, les deux fonctionnalités obligatoires doivent être implémentées : la mesure du rythme cardiaque (FT1-1-2-1) et la mesure de la qualité de reconnaissance d'une tonalité (FT1-1-3-2). Vous utiliserez l'environnement de développement Energia. Vous devez donc transcrire en langage C l'algorithme que vous avez simulé sous Matlab, en prenant en compte le fait qu'il faut maintenant effectuer les opérations en temps réel. Le signal audio sera envoyé sur la carte via la prise jack. Durant la phase finale d'intégration, cette connexion sera remplacée par la connexion micro/carte.

6.1. EXERCICES

Réfléchir à la traduction des algorithmes conçus sous Matlab en vue d'une implémentation temps réel en langage C sous Matlab : acquisition et stockage des échantillons dans un buffer, implémentation des fonctions de base comme le filtrage, la corrélation, l'estimation de puissance.

6.2. Travail en séance d'APP

A. Prise en main : acquisition des échantillons sonores

Ouvrir le programme 'power.ino'. Dans ce programme, le tableau *buf* de *BUF_SIZE* nombres flottants permet de stocker les échantillons au fur et à mesure de leur acquisition par la fonction *analogRead* : l'échantillon courant est stocké dans la case d'indice 0 tandis que les échantillons précédents sont décalés dans le registre. La variable *Ts* et la fonctions *micros()* permettent de gérer la fréquence de l'échantillonnage (choisir celle définie en simulation).

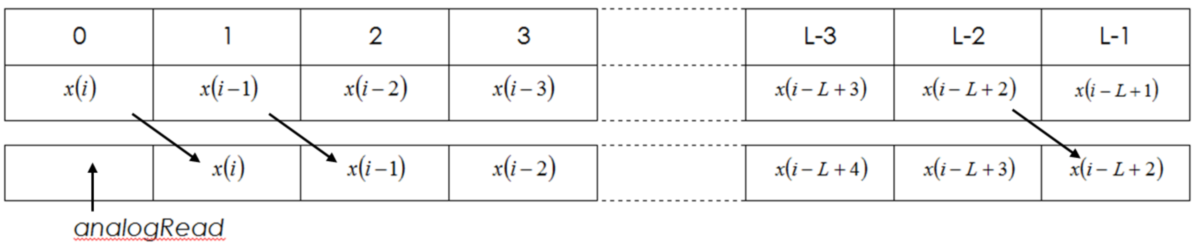


FIG. 19. Principe d'un buffer x de L éléments pour le stockage des L derniers échantillons

Compléter ce programme. Afficher des valeurs d'échantillons sur le moniteur (fonction *Serial.println,* le moniteur est ouvert en cliquant dans le menu *Outils/Moniteur série*).

Attention, l'affichage avec Serial.println est lent et perturbe l'échantillonnage. Il devra donc être supprimé lorsqu'on appliquera des traitements plus évolués comme le filtrage.

B. Implémentation de la fonctionnalité (FT1-1-2-1)

C. Implémentation de la fonctionnalité (FT1-1-3-2)