

Traitement numérique du signal

2022/2023

COMPOSANTE SIGNAL

Cours-Exercices-Tds-APP

Florence ROSSANT

SOMMAIRE

SOMMAIRE	3
1. INTRODUCTION	5
1.1. <i>Contexte</i>	5
1.2. <i>Objectifs pédagogiques de la composante signal</i>	6
1.3. <i>Organisation et évaluation des compétences</i>	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. <i>Valeur moyenne, énergie, puissance, valeur efficace</i>	10
2.1.1. Signaux continus	10
2.1.2. Signaux échantillonnés	11
2.2. <i>Amplification/atténuation d'un signal, gain</i>	13
2.3. <i>Rapport signal à bruit</i>	13
2.4. <i>Corrélation</i>	14
2.5. <i>EXERCICES</i>	16
2.6. <i>Travail en séances d'APP</i>	18
3. ANALYSE FREQUENTIELLE	21
3.1. <i>Transformée de Fourier d'un signal continu</i>	21
3.2. <i>Transformée de Fourier discrète</i>	22
3.3. <i>Signaux stochastiques et densité spectrale de puissance</i>	23
3.4. <i>Analyse spectrale</i>	24
3.5. <i>Théorème de Parseval</i>	25
3.6. <i>EXERCICES</i>	26
3.7. <i>Travail en séance d'APP</i>	28
4. NUMERISATION D'UN SIGNAL	30
4.1. <i>Echantillonnage</i>	30

4.2.	<i>Interpolation</i>	32
4.3.	<i>Quantification</i>	33
4.4.	<i>En pratique</i>	34
4.5.	<i>EXERCICES</i>	34
4.6.	<i>Travail en séances d'APP</i>	36
5.	FILTRAGE NUMERIQUE	39
5.1.	<i>Fonction de transfert</i>	39
5.2.	<i>Principaux filtres</i>	39
5.3.	<i>Gabarits et synthèse</i>	42
5.4.	<i>EXERCICES</i>	43
5.5.	<i>Travail en séance d'APP</i>	44
	ANNEXE I : IMPLEMENTATION MATERIELLE	48
	ANNEXE II : EXEMPLE DE RAPPORT SCIENTIFIQUE	49

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, afin d'évaluer la qualité de l'environnement dans lequel évolue une personne. Dans la composante signal, on s'intéresse particulièrement à la qualité de l'environnement sonore. Pour cela, on réalise l'acquisition de signaux numériques grâce à un microphone placé dans la rue, un jardin, sur un poste de travail, etc. et on cherche à détecter dans ces signaux les différents bruits qui peuvent être gênants (par leur puissance, leur durée, leur contenu spectral, etc.), à les caractériser, et éventuellement à les catégoriser en bruits pénibles voire dangereux, ou acceptables. En outre, on estime que le rythme cardiaque d'une personne au repos est lié au degré de stress auquel elle est soumise. Ainsi, une autre mission à effectuer est l'estimation du rythme cardiaque à partir de signaux d'électrocardiogramme (ECG).

Durant les cinq semaines que dure la composante signal, on mettra au point sous Matlab ces différentes fonctionnalités, via la résolution d'un ensemble de problèmes. La fonction de détection d'un signal ECG et d'estimation du rythme cardiaque sera implémentée sur le microcontrôleur, en lien avec le travail fait dans la composante électronique. La caractérisation de signaux sonores ou la détection de signaux sonores pénibles pourront aussi être implémentées (optionnel).

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 la 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. **Un exemple de rédaction est donné sous Moodle et en annexe II de ce document.**

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 4 séances par semaines sur 5 semaines. 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. Il est indispensable de lire préalablement le cours correspondant à

chaque séance et de préparer les exercices afin de profiter pleinement de ces séances. Des exemples pratiques sous Matlab sont aussi effectués lors de ces séances.

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'entraînement 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'entraînement 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.

Symétriquement, la non validation d'une compétence technique à l'issue du travail d'APP entraîne aussi l'évaluation « non acquis » de cette compétence, quel que soit le résultat obtenu à l'examen.

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 signal).

1.3.3. Planning général

Le tableau ci-dessous résume le planning général des travaux à effectuer durant les séances d'APP sur les 5 semaines que dure la composante signal. Chaque semaine, chaque groupe a 4 séances d'APP signal programmées, dont deux sont encadrées (1 séance de 4h et une séance de 2h). A ces séances s'ajoutent 5 séances de cours/Tds pour renforcer les aspects théoriques. Il faut aussi ajouter le travail personnel de compréhension du cours et les exercices d'entraînement.

	SEANCES TDS	SEANCES APP	TRAVAIL A EFFECTUER	
SEMAINE 1	Exercices de la section 2.5	S1	Initiation à Matlab. Etude des signaux échantillonnés dans le domaine temporel (signaux déterministes et stochastiques). Détection et analyse d'un signal (puissance, corrélation).	
		S2		
		S3		
		S4		
SEMAINE 2		S1	Résolution du problèmes I	
		S2		
		S3	Implémentation sur le microcontrôleur de la détection d'un signal audio Livrable I : rapport sur la résolution du problème I et l'implémentation sur microcontrôleur, code Matlab, C et démonstration sur la carte.	
		S4		
SEMAINES 3	Exercices de la section 3.6	S1	Etude des signaux dans le domaine fréquentiel. TFD des signaux déterministes. Analyse fréquentielle des signaux stochastiques.	
		S3		
		S3	Résolution du problème II : mesure du rythme cardiaque pour analyse temporelle et fréquentielle.	
		S4		
SEMAINE 4	Exercices de la section 4.5	S1	Etude de l'échantillonnage, théorème de Shannon, sous-échantillonnage. Quantification. Résolution du problème III : conception d'un système numérique d'acquisition de signaux audio. Livrable II : rapport sur la résolution du problème III	
		S2		
		S3		
		S4		
SEMANE 5	Exercices de la section 5.4	S1	Etude du filtrage numérique. Résolution du problème IV : détection de signaux « pénibles ». Livrable III (final) : résolution finale des problèmes II et IV et codes Matlab associés. Code C final de l'implémentation de la détection d'un signal ECG et de mesure du rythme cardiaque. Démonstration.	
		S2		
		S3		
		S4		
			EXAMEN INDIVIDUEL portant sur toutes les notions et méthodes vues en cours, TD et séances d'APP.	

FIG. 1- Planning de la composante signal

2. SIGNAUX ECHANTILLONNES

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

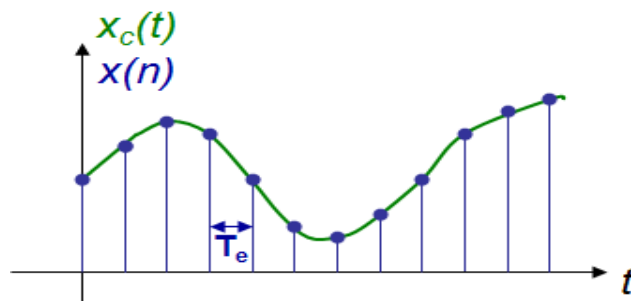


FIG. 2. Signal continu et signal échantillonné

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, une 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 :

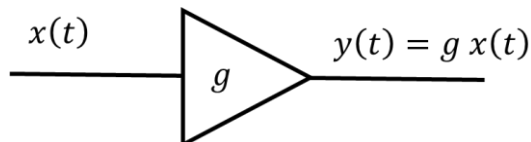


FIG. 3. Amplification

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 :

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

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

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

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

τ : décalage temporel

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

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

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

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

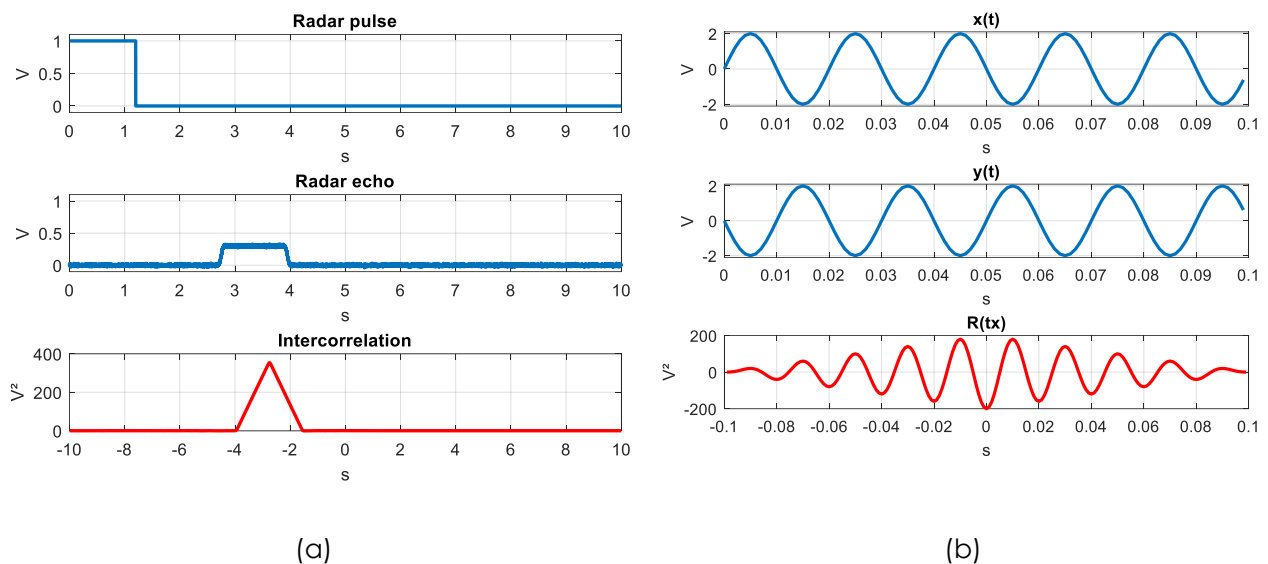


FIG. 4. Fonction d'intercorrélation. (a) le maximum de la fonction d'intercorrélation indique le décalage temporel entre l'impulsion d'un radar et l'écho reçu. (b) Intercorrélation pour deux fonctions sinusoïdales de période $T_0 = 0.02s$ et déphasées de π (demi-période).

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 :

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

2.5. EXERCICES

EXERCICE 2-1

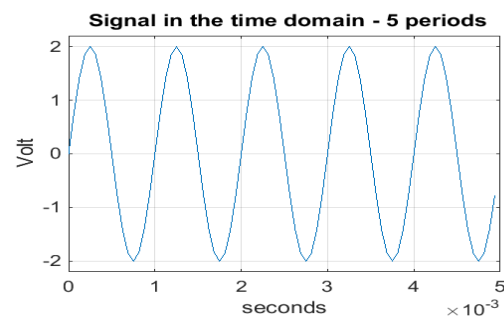
Prise en main de Matlab.

1. Générer un signal sinusoïdal x_1 de fréquence $f_1 = 1\text{KHz}$ de durée $D = 50\text{ms}$ et d'amplitude $A_1 = 1\text{V}$, avec un fréquence d'échantillonnage $F_e = 20\text{KHz}$. 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 $f_2 = 100\text{Hz}$ et d'amplitude $A_2 = 0.3\text{V}$. 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

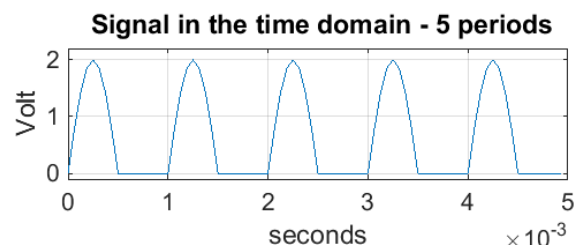
1. Calculer la valeur T_0 de la période, la valeur moyenne, l'énergie, la puissance moyenne et la valeur efficace des signaux analogiques suivants :

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



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

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

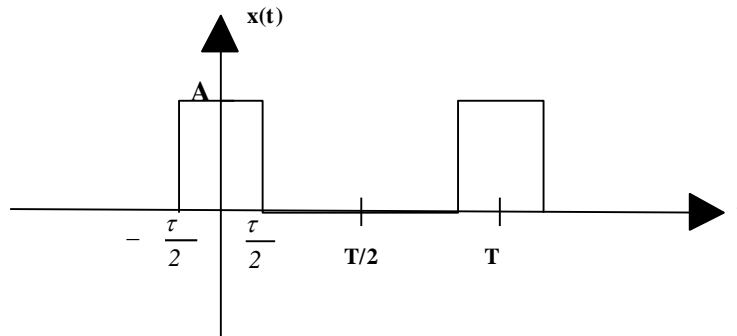


2. On échantillonne ces signaux à la fréquence $f_e = 16\text{KHz}$. Donner l'expression temporelle de chacun d'entre eux. Indiquer la valeur de l'échantillon $n = 347$. A quel instant a-t-il été prélevé ? Combien y a-t-il d'échantillons par période ?

- 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 $A = 2.5V$.



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

EXERCICE 2-4

- On mesure la puissance d'un signal Wifi reçu sur un smartphone : $P_r = -63 \text{ dBm}$. Quelle est la puissance du signal reçu en Watt ?
- On mesure la puissance du signal émis par la borne Wifi : $P_e = 2W$. Quelle est la puissance émise en dBm ?
- 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 échantillonnés 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 $D = \frac{n}{f_1}, n \in \mathbb{N}$.
3. Sous Matlab, **programmer** l'intercorrélation $C_{x_1, x_2}(\tau)$ (avec la normalisation) et retrouver les résultats. On prendra :
 - $f_1 = 500\text{Hz}, f_2 = 550\text{Hz}, A_1 = 0.5\text{V}, A_2 = 1.5\text{V},$
 - fréquence d'échantillonnage $F_e = 10\text{KHz}$.
 - $\tau \in \left[0, \frac{2}{f_1}\right]$
 - $\varphi = 0\text{rad}$
4. Prendre $f_2 = f_1$ et faire varier φ . Interpréter.

2.6. Travail en séances d'APP

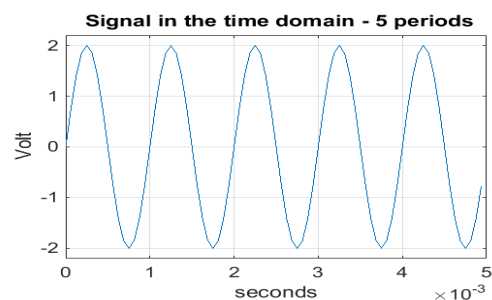
Prise en main de Matlab

Se reporter au tutoriel.

Exercices sous Matlab

- A. Créer un signal sinusoïdal discret $x(nT_e) = A \sin(2\pi f_0 nT_e)$ dont les caractéristiques sont les suivantes :
- Fréquence du sinus $f_0 = 1\text{KHz}$, période T_0
 - Amplitude $A = 2\text{V}$
 - Fréquence d'échantillonnage $F_e = \frac{1}{T_e} = 16\text{KHz}$,
 - Durée : $D = 2\text{s}$.

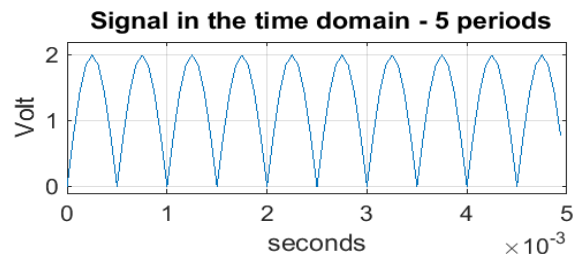
Afficher les 5 premières périodes. Bien indiquer les échelles sur les deux axes. La figure ci-contre montre le graphe que vous devez obtenir.



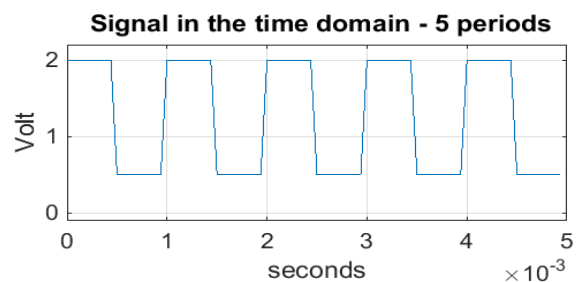
- B. Faire une estimation de la valeur moyenne, de la puissance moyenne (en W et en dBm) et de la valeur efficace de ce signal (échantillonné), en choisissant correctement la durée de l'estimation. Comparer avec les valeurs trouvées dans les exercices. Augmenter la fréquence d'échantillonnage. Conclure.

- C. Refaire le même travail sur les signaux suivants :

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



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



- D. Générer un signal y discret, échantillonné également à $F_e = 16 \text{ KHz}$:
 $y(nT_e) = B \sin(2\pi f_0 nT_e + \varphi)$ tel que
- La puissance du signal y est égale à 32 dBm
 - $\varphi = \frac{\pi}{3} \text{ rad}$
- E. Programmer l'autocorrélation du signal x et l'intercorrrelation des signaux x et y . Afficher les résultats et interpréter.
- F. Estimer la puissance de signaux environnementaux (MarteauPiqueur01.mp3, Jardin01.mp3, Jardin02.mp3, Ville01.mp3) sur des fenêtres glissantes. 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. Ecouter les signaux à l'aide de la fonction `sound`. Etudier l'influence de la durée de la fenêtre d'analyse et justifier le paramétrage retenu.

Résolution de problèmes

Problème I : Proposer un algorithme qui permet de détecter la présence/absence d'un signal audio enregistrant le bruit ambiant. Le micro utilisé a une sensibilité S (dBV) et amplifie le signal électrique avec un gain G . On définit un « bruit pénible », appelé plus simplement « bruit », comme un signal sonore de durée supérieure à Dt (s) et de niveau supérieur ou égal à P_{SPL} (dB SPL¹). A contrario, un bruit acceptable, qu'on appellera par abus

¹ Sound Pressure Level

de langage « silence », correspond à un signal sonore de niveau inférieur à P_{SPL} . Les plages de silence de durée inférieure à dt (s) seront intégrées aux bruits détectés.

Caractériser chaque bruit détecté par :

- sa durée,
- sa puissance moyenne en mW et en dBm,
- sa tension RMS en V
- son coefficient d'autocorrelation $\max_{\tau>0} C_{xx}(\tau)$ et le délai correspondant

$\arg \max_{\tau>0} C_{xx}(\tau)$.

Paramètres $S = -48 \text{ dBV}$, $G = 30 \text{ dB}$, $P_{SPL} = 80 \text{ dB SPL}$, $Dt = 1 \text{ s}$, $dt = 0.5 \text{ s}$.

Les expérimentations seront menées sur les signaux de la question F.

Implémentation I : Implémenter le code de détection d'un signal sur le microcontrôleur, en choisissant une fréquence d'échantillonnage compatible avec les caractéristiques matérielles (micro, filtre passe-bas anti-repliement). Se reporter à l'annexe 1 pour la numérisation des signaux sur la carte Tiva et la gestion du temps réel. L'implémentation de la caractérisation des bruits détectés est optionnelle.

Livrable I

Rédiger le premier rapport (voir le planning en **FIG. 1**). **Vérifier préalablement les consignes données** dans la section 1.2 de l'introduction de ce document (paragraphe compétences générales) et étudier **l'exemple de rédaction fourni en annexe II**. Rendre les codes Matlab et le code C. Faire une démonstration de l'implémentation matérielle à votre tuteur en séance encadrée.

Transmettre à votre tuteur, par mail, le rapport au format pdf ainsi que l'ensemble des programmes demandés, le tout dans un fichier zip. Indiquer dans le titre le numéro complet de groupe (G1a, ...), et mettre tous les membres du groupe en copie du mail.

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 propriété $X_c(f) = X_c^*(-f)$ 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 :

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

où $\text{Re}\{z\}$ et $\text{Im}\{z\}$ 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 $[0, F_e]$ 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

² 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. 5a), 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. 5b, 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.

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

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

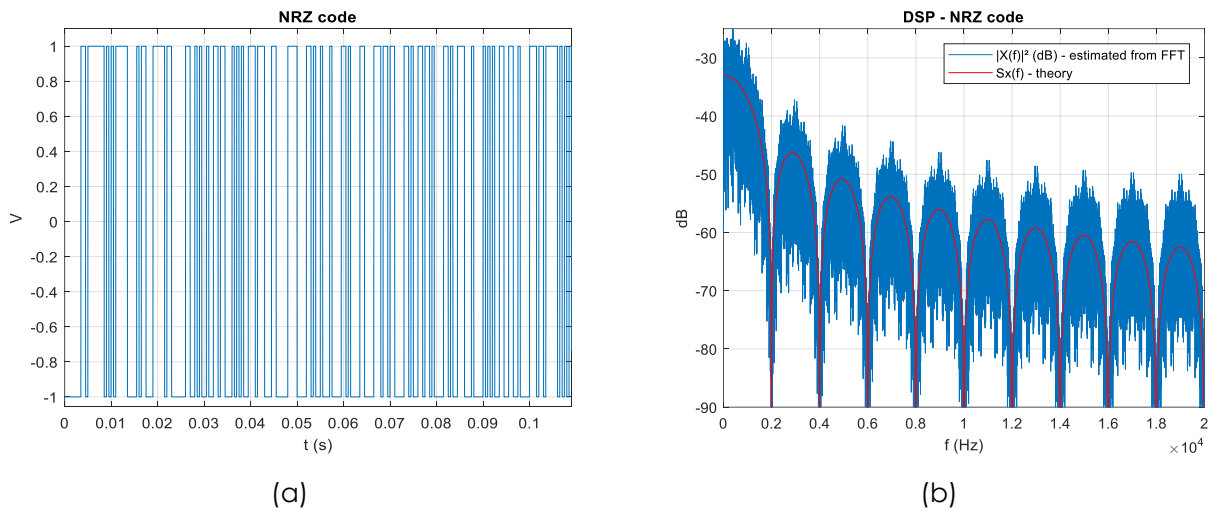


FIG. 5. Estimation de la densité spectrale de puissance (DSP) d'un signal NRZ ; (a) signal dans le domaine temporel : un bit 1 est codé par un créneau de 1V, un bit 0 par un créneau de -1V, avec un débit de 2kbits/s; (b) Densité spectrale de puissance (DSP) : estimation par le module au carré de la TFD du signal (a) et courbe théorique calculée par le théorème de Wiener Khintchine.

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 :

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

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

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.

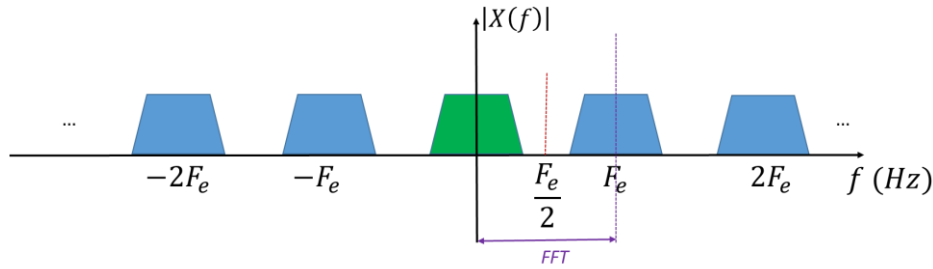


FIG. 6. Interprétation spectrale d'un signal échantillonné à la fréquence F_e . En vert le spectre du signal analogique réel $x(t)$, symétrique par rapport à $f = 0$ Hz. En vert et bleu le spectre du signal échantillonné $x(nT_e)$, qui correspond à la périodisation du spectre du signal $x(t)$. La FFT 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. 5, la DSP du signal a été représentée sur l'intervalle $\left[0, \frac{F_e}{2}\right]$ ($F_e = 40$ KHz).

3.5. Théorème de Parseval

Théorème de Parseval pour les signaux discrets

Soit $x_n, n \in [0, N - 1]$ un signal discret et $X_k, k \in [0, N - 1]$ sa transformée de Fourier discrète. On a

$$\frac{1}{N} \sum_{n=0}^{N-1} x_n^2 = \sum_{k=0}^{N-1} |X_k|^2$$

NB : cette équation est valable pour la normalisation de la TFD par le facteur $\frac{1}{N}$. Elle doit être adaptée dans le cas contraire (utilisation de la fonction fft de Matlab).

3.6. 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.

1. Calculer la transformée de Fourier d'un signal constant d'amplitude A .
2. Calculer la transformée de Fourier du signal $x(t) = Ae^{2\pi jf_0 t}$.
3. Calculer la transformée de Fourier du signal $A\sin(2\pi f_0 t)$.
4. Calculer la transformée de Fourier du signal $A\cos(2\pi f_0 t)$.

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 $x_1(t)$:

- Fréquence $f_1 = 500\text{Hz}$
 - Amplitude $A_1 = 2\text{V}$
 - Fréquence d'échantillonnage $F_e = 8\text{KHz}$
 - Durée : $D = 50\text{ms}$.
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 $f_2 = 600\text{Hz}$ et d'amplitude $A_2 = 1\text{V}$. Commenter le résultat obtenu.

EXERCICE 3-3

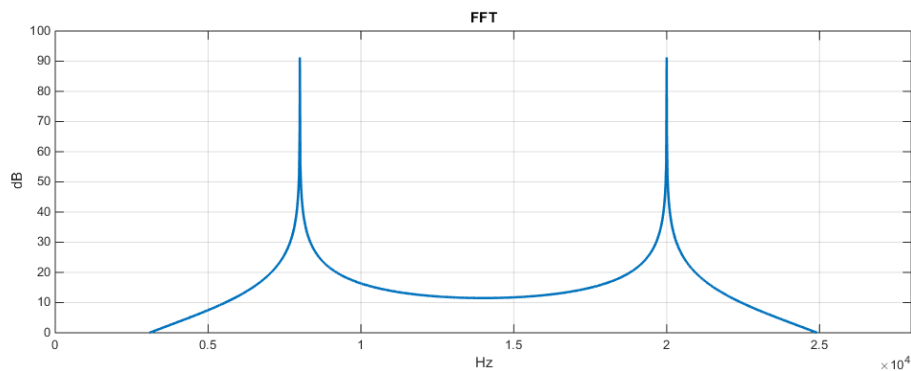
Calculer la transformée de Fourier d'une porte d'amplitude A et de durée τ :

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

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 $F_e = 28 \text{ KHz}$. 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.

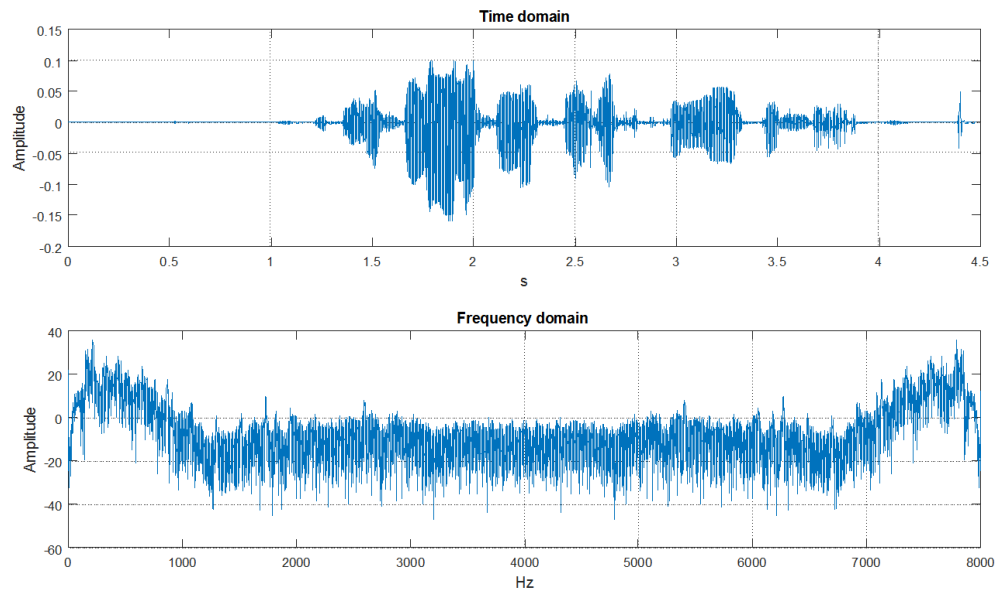
1. Calculer la transformée de Fourier discrète de ce signal. Le résultat est-il réel ?
2. Montrer que la TFD prend des valeurs réelles si on translate temporellement le signal de $-2T, T$ étant la période d'échantillonnage.

EXERCICE 3-6

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

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 ?



3.7. Travail en séance d'APP

Exercices sous Matlab

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 signal 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 environnementaux correspondant à des alarmes. On utilisera le programme de détection/extraction des bruits du Problème I pour faire l'analyse des sons d'alarmes présents dans le fichier alarmes.mp3.

C. Pour chacun des signaux détectés :

- Afficher la densité spectrale de puissance, avec un pas fréquentiel inférieur à 10 Hz.
- Vérifier le théorème de Parseval en estimant la puissance à partir des échantillons sonores et à partir des coefficients de Fourier.
- Détecter la fréquence correspondant à la dsp maximale.
- Estimer la bande spectrale du signal à -30dB et indiquer le pourcentage d'énergie contenue dans cette bande.
- Afficher le spectrogramme (fonction `spectrogram`) en prenant une fenêtre de 0.25s sans recouvrement, avec un pas fréquentiel inférieur à 10 Hz. Qu'apporte le spectrogramme par rapport à la DFT de l'intégralité du son ?

Relever les caractéristiques demandées pour chaque alarmes du fichier et présentez vos programmes et résultats à votre tuteur.

Résolution de problèmes

Problème II : Proposer **deux algorithmes** permettant de détecter un signal ECG et d'estimer le rythme cardiaque correspondant. On travaillera sur les signaux ECG stockés dans les fichiers 100.wav, 101.wav, ..., et 109.wav. Le résultat de ces algorithmes doit être une fréquence cardiaque mesurée au cours du temps, lorsqu'un signal ECG est présent, 0 sinon.

Appliquer les algorithmes proposés sur tous 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.

Remarque : 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 !

Implémentation II : Commencer au plus tôt l'implémentation sur microcontrôleur de la détection d'un signal ECG et de mesure du rythme cardiaque (voir le planning en **FIG. 1**). Choisir une fréquence d'échantillonnage compatible avec les caractéristiques matérielles (micro, filtre passe-bas anti-repliement). Adapter le code de l'implémentation I pour la détection préalable du signal ECG : on ne doit pas mesurer de rythme quand il n'y a pas de signal utile ! Choisir l'une des deux approches proposées pour résoudre le Problème II, en motivant ce choix.

Le programme devra être rendu dans le livrable final.

4. NUMERISATION D'UN SIGNAL

Nous avons directement travaillé sur des signaux échantillonnés. Or la plupart des capteurs (par exemple un microphone) 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. 7) :

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

On montre alors que, dans le domaine fréquentiel, cela correspond à une périodisation du spectre (FIG. 7) :

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

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. 8) ?

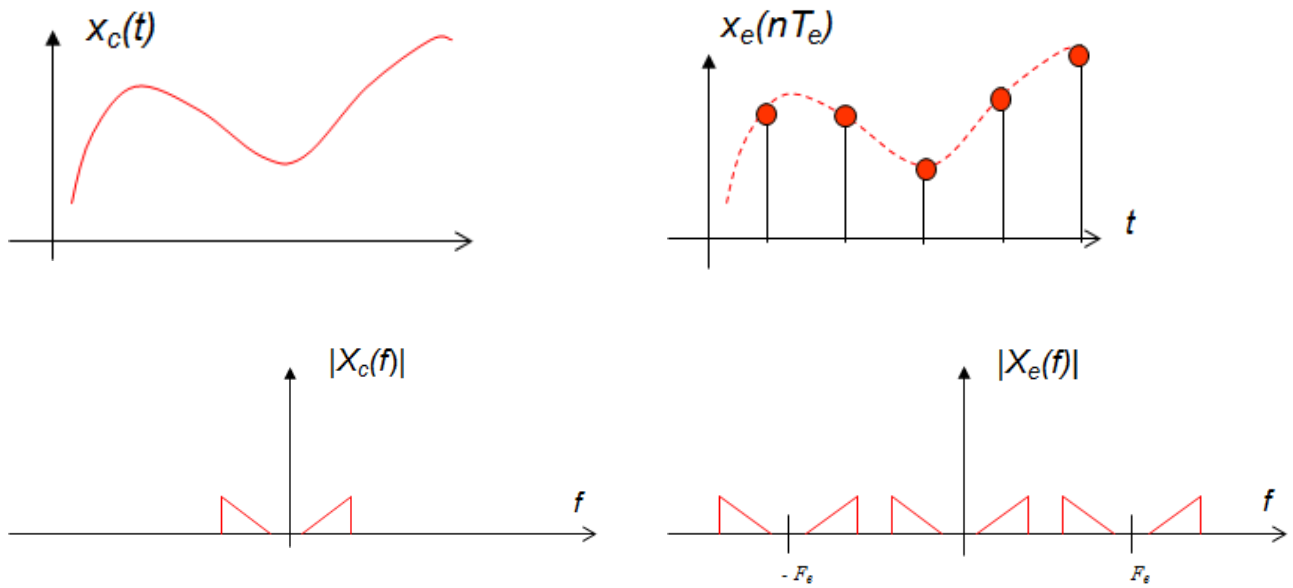


FIG. 7. Principe de l'échantillonnage

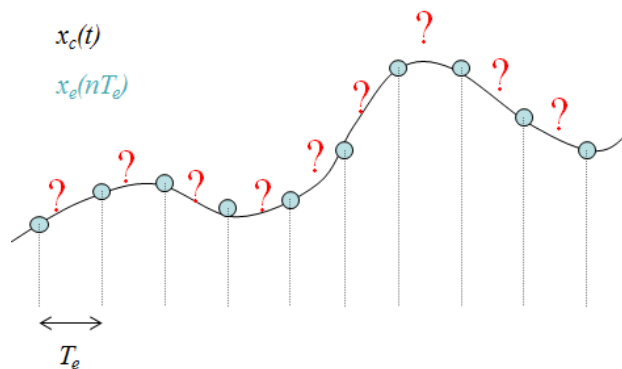
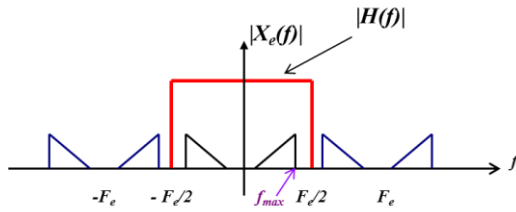


FIG. 8. Problématique de reconstruction du signal continu $x_c(t)$ à partir des échantillons $x_n = x_e(nT_e)$

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 $F_e \geq 2f_{max}$.

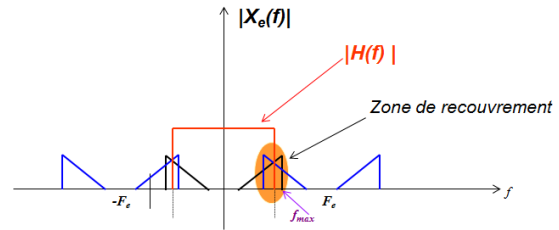


Le théorème de Shannon est respecté puisque $F_e \geq 2f_{max}$

Il n'y a pas de superposition des copies du spectre $X_c(f)$ (partie en noir 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 copies du spectre (en bleu) du spectre initial (en noir).

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

FIG. 9. Illustration de l'aliasing et de la reconstruction du signal continu $x_c(t)$ à partir des échantillons $x_e(nT_e)$

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 copies centrées sur les multiples de la fréquence d'échantillonnage. Si on filtre le spectre du signal échantillonné avec un filtre passe-bas $H(f)$ ³ qui supprime toutes les copies 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. 9 illustre cela.

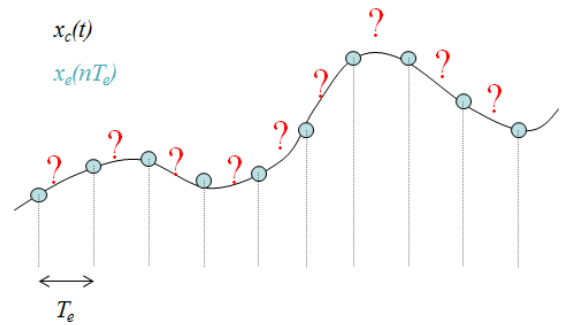
4.2. Interpolation

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

³ La notion de filtrage est abordée dans la section 0. Un filtre est un système linéaire qui supprime certaines fréquences dans le signal mis à son entrée.

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

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 approximatés par des valeurs discrètes multiples du **pas de quantification** q . Chaque valeur discrète ($k \cdot q$, 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. 10 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 :

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

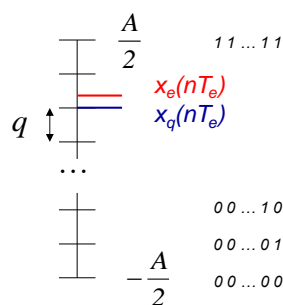


FIG. 10. Quantification d'un signal échantillonné

La quantification des échantillons conduit à une erreur puisque les échantillons sont approximatés sur une grille de valeurs discrètes (FIG. 11). On crée donc un bruit appelé **bruit de quantification**. Le bruit de quantification est la différence $e(nT_e) = x_e(nT_e) - x_q(nT_e)$.

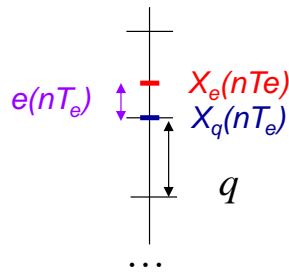


FIG. 11. Bruit de quantification

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 $f_0 = 1000\text{Hz}$, $A_0 = 1$, $A_1 = 2$, $A_2 = 0.5$.

1. Calculer la transformée de Fourier de ce signal.

On échantillonne le signal $x(t)$ à la fréquence $F_e = 5000\text{ Hz}$.

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 $x_0 = A \cos(2\pi f_0 t)$, 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 $f_0 = 1KHz$, échantillonné 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 :

$$x_q = \frac{\text{round}(x_e \cdot 2^{(b-1)})}{2^{(b-1)}}$$

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

Exercices sous Matlab

Echantillonnage : étude sur un cas d'école

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

- Fréquence du sinus $f_0 = 440\text{Hz}$, période T_0

Amplitude $A = 1V$

- Fréquence d'échantillonnage $F_{e1} = 8\text{ KHz}$
- 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 $F_{e2} = 0.5\text{ KHz}$. 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 récupérerait-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 *BonneJournée.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 ($SNR(b)$) et en fonction du débit ($SNR(D)$). 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 :

$$x_q = \frac{\text{round}(x_e \cdot 2^{(b-1)})}{2^{(b-1)}}$$

- 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: On souhaite effectuer un enregistrement numérique en utilisant un micro de sensibilité $S = -30\text{dBV}$, de bande passante 200 Hz-20 kHz. On suppose que ce micro n'introduit aucune distorsion sur la bande considérée et aucun bruit. Le signal analogique est ensuite amplifié avec un gain $G = 20\text{dB}$. La dynamique après amplification est $[0\text{V}, +5\text{V}]$. Ce signal est ensuite numérisé. On choisit des paramètres de numérisation afin de garantir un rapport signal à bruit d'au moins 60dB pour un

signal acoustique de $P_{dB\ SPL} = 6\text{ dB SPL}$ (2 fois le minimum de pression audible).

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 d'audio 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.

Livrable II

Rapport sur la résolution du problème III. Respecter les consignes sur la rédaction d'un rapport scientifique. Transmettre le rapport au format pdf à votre tuteur, par mail.

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. 12).

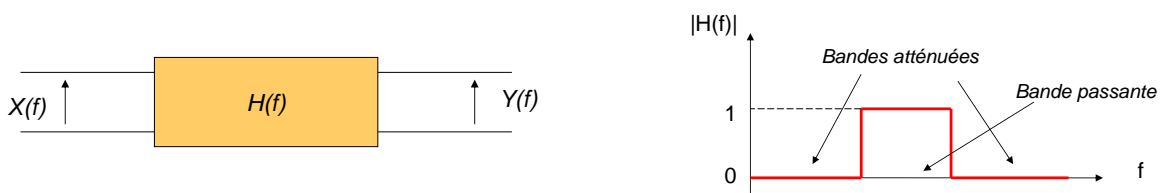


FIG. 12. Filtre de fonction de transfert $H(f)$

5.2. Principaux filtres

On distingue quatre catégories de filtres, schématisés dans la FIG. 13 :

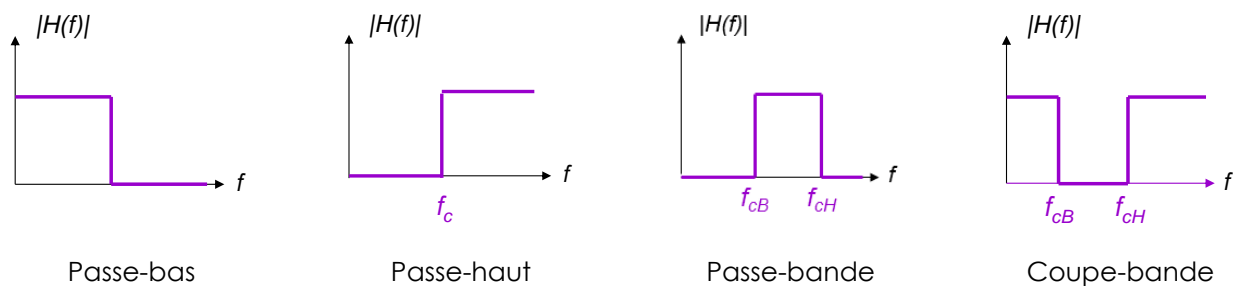


FIG. 13. Principaux filtres

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.

Dans ce qui suit, on va relier ces équations fondamentales des filtres numériques RIF aux relations plus générales du filtrage. On se place dans le cas continu.

On définit le **produit de convolution** de deux signaux analogiques $x(t)$ et $y(t)$ par :

$$x(t) * y(t) = \int_{-\infty}^{+\infty} x(\tau) y(t - \tau) d\tau$$

La transformée de Fourier d'un produit de convolution est le produit des transformées de Fourier :

$$TF\{x(t) * y(t)\} = TF\{x(t)\} TF\{y(t)\} = X(f) Y(f)$$

On définit la **réponse impulsionnelle du filtre** comme la réponse de ce filtre à une impulsion de Dirac $\delta(t)$. Soit $h(t)$ la réponse d'impulsionnelle d'un filtre, soient $x(t)$ le signal mis en entrée du filtre et $y(t)$ le signal de sortie. On a les relations suivantes :

$$y(t) = h(t) * x(t) = \int_{-\infty}^{+\infty} h(\tau) x(t - \tau) d\tau$$

$$Y(f) = H(f)X(f)$$

Ainsi, on peut envisager le **filtrage dans le domaine fréquentiel** par une multiplication de la transformée de Fourier du signal d'entrée par la réponse en fréquence du filtre, et **dans le domaine temporel** par le produit de convolution du signal d'entrée par la réponse impulsionnelle du filtre.

Revenons au cas discret et au cas particulier des filtres à réponse impulsionnelle finie. Pour un filtre RIF, il est facile de démontrer que la réponse impulsionnelle est égale aux coefficients du filtre : $y(n) = h(n)$ quand $x(n) = \delta(n)$. Ainsi on voit que :

- L'équation aux différences correspond en fait au produit de convolution discret du signal d'entrée avec la réponse impulsionnelle du filtre.
- La réponse en fréquence du filtre est la transformée de Fourier de la réponse impulsionnelle du filtre.

Tout ceci est donc conforme aux concepts généraux du filtrage. La figure ci-dessous illustre le principe du filtrage numérique avec des filtres RIF :

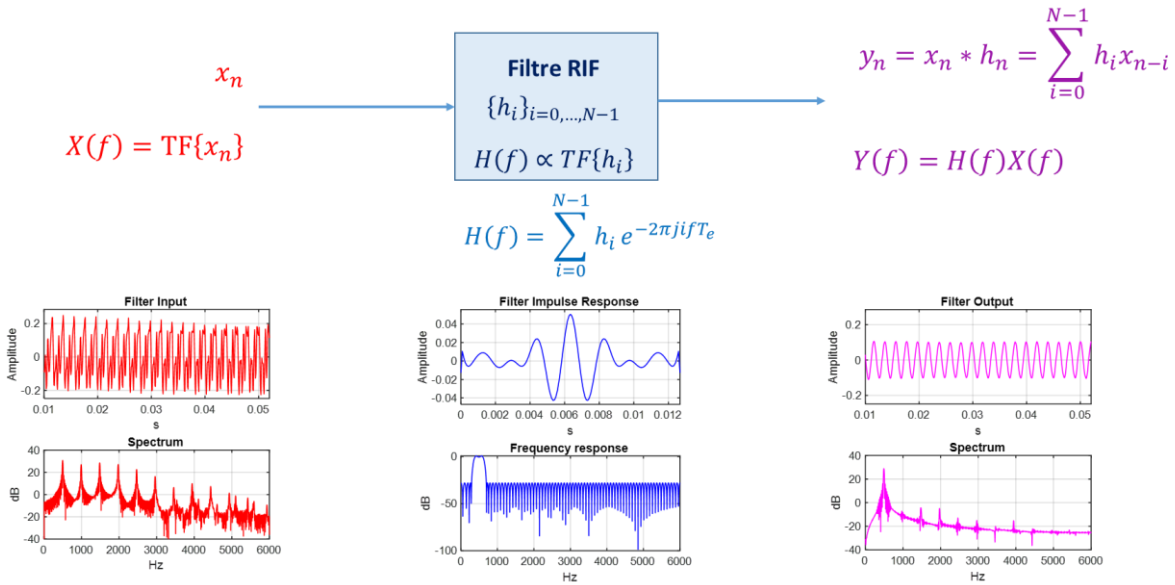


FIG. 14. Filtrage RIF : relations dans le domaine temporel et dans le domaine fréquentiel

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'aborderons 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. 13 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.

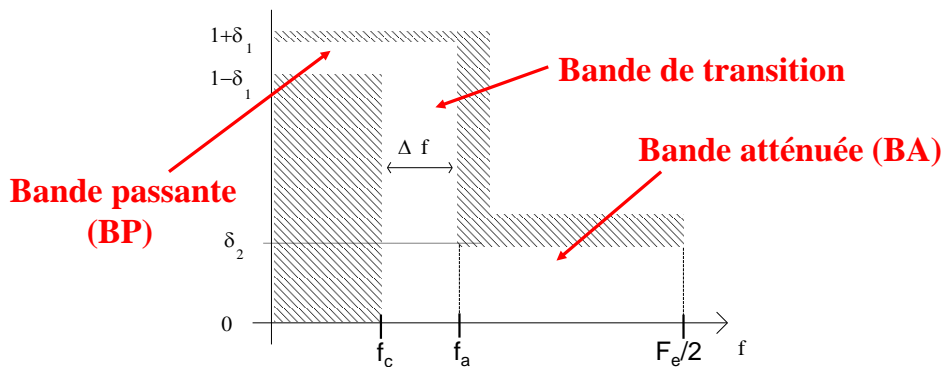


FIG. 15. Gabarit d'un filtre numérique passe-bas

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** ($[f_a, \frac{F_e}{2}]$)

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

EXERCICE 5-1

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 $f_{SI}=494\text{Hz}$. Les notes de musique dont les fréquences sont les plus proches sont le « LA# » de fréquence $f_{LA\#}=466\text{ Hz}$ et le « DO » de fréquence $f_{DO}=523\text{Hz}$. 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 toutes 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

Exercices sous Matlab

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 \sharp	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 \sharp ou ré \flat	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 \sharp ou mi \flat	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 \flat	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 \sharp	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 \sharp ou sol \flat	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 \sharp ou la \flat	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 \sharp ou si \flat	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 \flat	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. 16. Tableau des fréquences de le tempérément égal. Le passage d'une note à la suivante, distance d'un demi ton, se fait par multiplication de la fréquence en Hz par $2^{\frac{1}{12}}$.

- 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, **FIG. 16**). 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.

Lancer l'application *Filter Designer*, qui permet de concevoir des filtres via une interface graphique. Cliquer ensuite sur « New » dans la colonne « Filters ». Vous obtenez alors la fenêtre de design de filtres (FIG. 17).

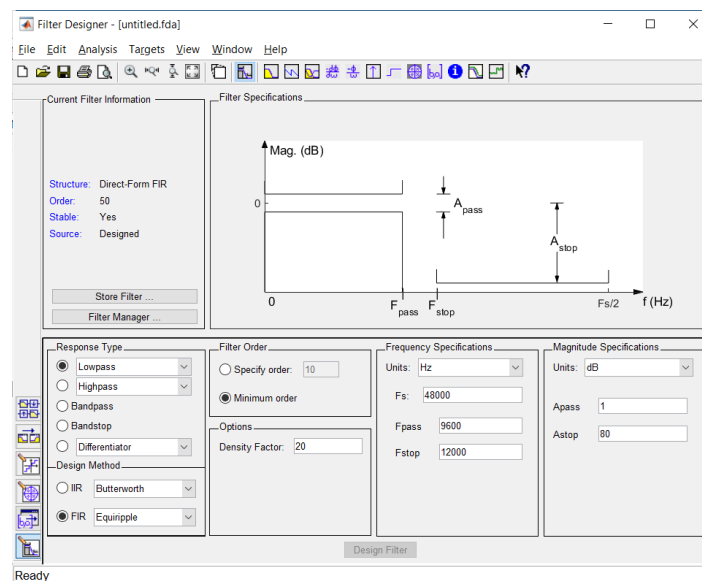


FIG. 17. Conception d'un filtre sous Filter Designer

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. 18) et cliquer sur « export ». Donner ensuite un nom à votre fichier, par exemple *filtre.mat*.

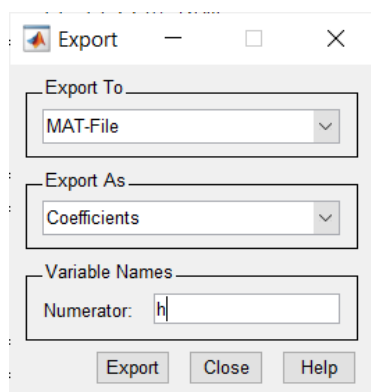


FIG. 18. Sauvegarde des coefficients d'un filtre sous Filter Designer

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 problèmes

Problème IV : On souhaite détecter les signaux environnementaux qui comprennent des composantes spectrales très aiguës et pénibles pour l'oreille humaine. Ces signaux sont définis ainsi : 20% de leur puissance est comprise dans les fréquences supérieures à 2KHz, avec une puissance sonore totale supérieure à 74 dB SLP.

Concevoir un système de détection de ces signaux pénibles fondé sur filtrage numérique. On prendra un micro de sensibilité égale à -55 dBV et de gain égal à 30 dB. Discuter les critères énoncés.

Implémentation III : Finaliser l'implémentation de la détection d'un signal ECG et de mesure du rythme cardiaque (fonction indispensable). Optionnel : implémenter la solution du Problème IV.

Livrable III (final)

Transmettre à votre tuteur le livrable final conformément au planning (**FIG. 1**) : résolution des problèmes II et IV, codes Matlab associés, codes C, démonstration.

Transmettre le rapport (pdf) et les codes par mail à votre tuteur, via un fichier zip.

ANNEXE I : IMPLEMENTATION MATERIELLE

On utilise l'environnement de développement **Energia**. Vous devez donc transcrire en langage C les algorithmes que vous avez mis au point sous Matlab, en prenant en compte le fait qu'il faut maintenant effectuer les opérations en temps réel. Dans un premier temps, Le signal sera envoyé sur la carte via la prise jack, puis sera issu des capteurs de la carte électronique. Utiliser le connecteur jack permet de s'affranchir de problèmes liés à l'acquisition des signaux afin de mettre au point plus facilement les programmes de traitement du signal. Durant la phase finale d'intégration, cette connexion sera remplacée par la connexion micro/carte ou la connexion capteur ECG/carte.

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).

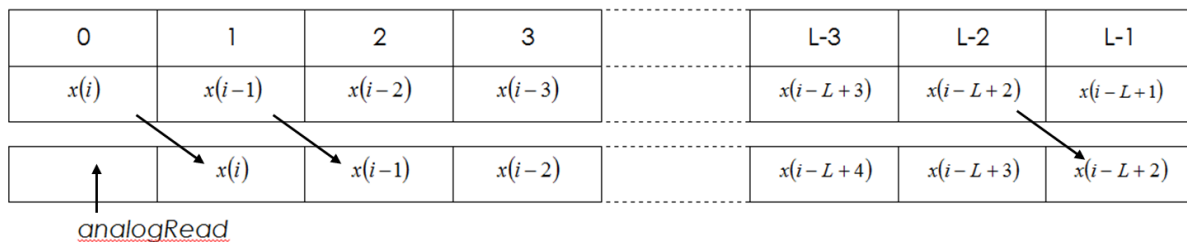


FIG. 19. Principe d'un buffer x de L éléments pour le stockage des L derniers échantillons

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.

ANNEXE II : EXEMPLE DE RAPPORT SCIENTIFIQUE

Problème : Une télécommande infrarouge (IR) envoie des trames composées de 8 bits d'entête (header) et de 32 bits de données utiles (data) à un débit $D = 500 \text{ bits/s}$. Les trames sont répétées tant que l'utilisateur appuie sur la télécommande. Ce signal est transmis au récepteur, qui le transforme en signal NRZ (tension +V pour un bit 1, -V pour un bit 0) échantillonné à une fréquence $F_e = 10 \text{ KHz}$. Le signal est jugé décodable si le rapport signal à bruit est supérieur à $SNR_{min} = 7 \text{ dB}$ et si au moins $NbF_{min} = 3$ trames sont détectées. L'en-tête est défini par la séquence binaire 1 0 0 1 1 0 1 0.

Proposer une méthode de décodage des signaux de cette télécommande :

- détection du signal utile dans le signal reçu,
- synchronisation par détection du header.
- décodage des données

et évaluer quantitativement la méthode.

Les signaux à traiter sont stockés dans les fichiers IR1.mat, ..., IR5.mat. Ces fichiers contiennent les signaux reçus et échantillonnés (signal) et les bits de données émises (data), utiles pour évaluer la qualité du décodage. On suppose que les 60 premières ms de ces signaux de contiennent que du bruit.

Voir le rapport avec mise en page complète (titre, table des matières, etc.) sous Moodle

SYSTEME DE DECODAGE D'UN SIGNAL DE TELECOMMANDE IR

1. Introduction

On propose une méthode de décodage de signaux de télécommande IR, dont les caractéristiques sont les suivantes :

- Trames constituées de 8 bits d'entête et de 32 bits de données utiles.
- Signal reçu codé en NRZ, échantillonné à la fréquence $F_e = 10 \text{ KHz}$.

Le signal est jugé décodable si le rapport signal à bruit est supérieur à $SNR_{min} = 7 \text{ dB}$ et si au moins $NbF_{min} = 3$ trames sont détectées. Notre approche est composée de trois étapes : la détection d'un signal de télécommande, la synchronisation du récepteur par la détection du header, le décodage des bits de données utiles. Nous évaluerons notre méthode sur les 5 signaux IRn.mat proposés.

2. Méthode

La Figure 1 montre un exemple signal reçu (IR1.mat).

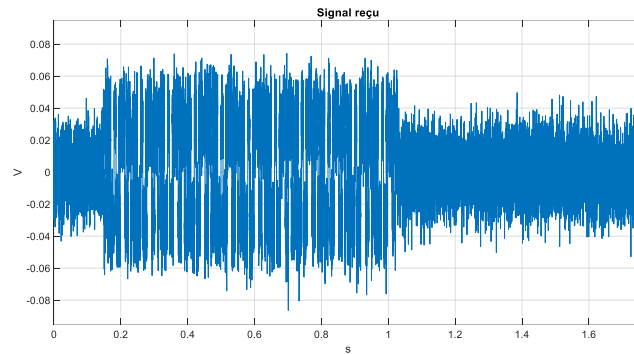


Figure 1. Exemple de signal reçu par le récepteur

Dans un premier temps, nous proposons de détecter le signal utile par analyse de la puissance moyenne calculée sur des fenêtres glissantes. Le signal sera décodé uniquement si le signal utile détecté a une durée supérieure à la durée de 3 trames. Le cas échéant, nous calculerons l'intercorrélacion de ce signal extrait avec le header et nous détecterons les maxima locaux qui correspondent à la position temporelle du début de chaque header. Une fois cette synchronisation effectuée, nous extrairons le signal correspondant à une trame et nous décodons les données par sous-échantillonnage et seuillage. La Figure 1 montre le schéma fonctionnel du système.

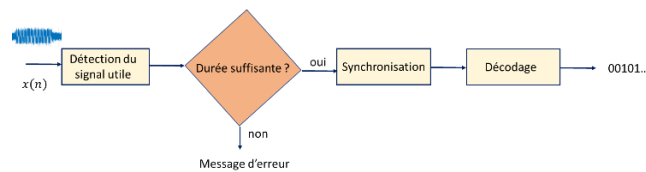


Figure 2. Schéma fonctionnel

On note :

- $F_e = 10\text{KHz}$ la fréquence d'échantillonnage, $T_e = \frac{1}{F_e}$ la période d'échantillonnage,
- $x(n) = x(nT_e)$ le signal échantillonné reçu,
- $D = 500\text{bits/s}$ le débit binaire,
- $T_b = \frac{1}{D}$ la durée d'un bit
- $N_h = 8$ le nombre de bits dans le header, $N_d = 32$, le nombre de bits de données utiles dans chaque trame
- $d_{noise} = 60\text{ms}$ la durée minimale de signal ne contenant que du bruit
- $SNR_{min} = 7\text{dB}$ le rapport signal à bruit minimal.

2.1. Détection du signal utile.

On estime la puissance en chaque instant nT_e sur une fenêtre temporelle de durée d :

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

avec $K = \left\lfloor \frac{1}{2} \left(\frac{d}{T_e} - 1 \right) \right\rfloor$, où $\lfloor y \rfloor$ désigne la partie entière de y . On a choisi $d = N_h T_b / 2$, correspondant à la moitié de la durée de l'entête, ce qui expérimentalement donne une courbe lisse mais avec des fronts raides en début et fin de signal, comme illustré sur la Figure 3.

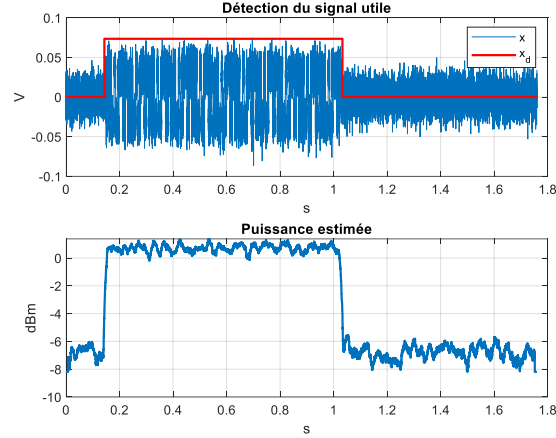


Figure 3. Estimation de la puissance en dBm et détection du signal utile

La puissance du bruit est estimée sur les premières $d_{noise} = 60ms$, en excluant les K premiers échantillons de P qui ne sont pas calculés (effet de bord), par la valeur médiane.

$$P_b = \text{median}\{P(n)\}_{n \in [K+1, \lfloor d_{noise}/T_e \rfloor]}$$

Dans l'exemple de la Figure 3, on a trouvé $P_b = -6.7dBm$.

Comme le rapport signal à bruit est supérieur ou égal à $SNR_{min} = 7dB$, on choisit un seuil de détection S qui s'ajuste dynamiquement par rapport au niveau de bruit estimé :

$$S = P_b + \frac{1}{2} SNR_{min}$$

Ainsi, la détection du signal utile est donnée par

$$x_d(n) = \begin{cases} 1 & \text{si } P(n) > S \\ 0 & \text{sinon} \end{cases}$$

Ce signal de détection est représenté en rouge sur la Figure 3. Il couvre bien la durée durant laquelle des données sont reçues. Les indices n_1 et n_2 , correspondant respectivement au front montant et au front descendant de $x_d(n)$, mémorisent les indices de début et de fin du signal.

On teste alors la validité du signal reçu, qui doit contenir au moins $NbF_{min} = 3$ trames, en vérifiant que $(n_2 - n_1) > NbF_{min}(N_h + N_d)$. Le programme stoppe sur un message d'erreur si cette condition n'est pas vérifiée. Sinon, le programme enchaîne sur la détection des en-têtes.

2.2. Synchronisation

Cette étape permet de synchroniser le récepteur par la détection d'un instant correspondant au début d'un en-tête. Soit $x_h(n)$ la séquence binaire de l'entête (8 bits) codée en NRZ.

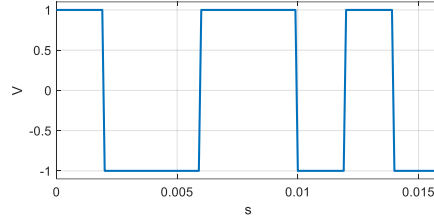


Figure 4. Header codé en NRZ, de durée $N_h T_b = 0.016s$ et contenant $N = N_h T_b F_e = 160$ échantillons

Soit $N = N_h T_b F_e = 160$ le nombre d'échantillons de $x_h(n)$ ($n \in [0, N]$). On calcule l'intercorrrelation du signal reçu utile avec x_h .

$$C(m) = \frac{1}{N} \sum_{k=0}^{N-1} x_h(k)x(m+k), n_1 \leq m \leq n_2 - N$$

L'intercorrrelation est donc calculée sur des décalages temporel mT_e couvrant le signal utile (Figure 5).

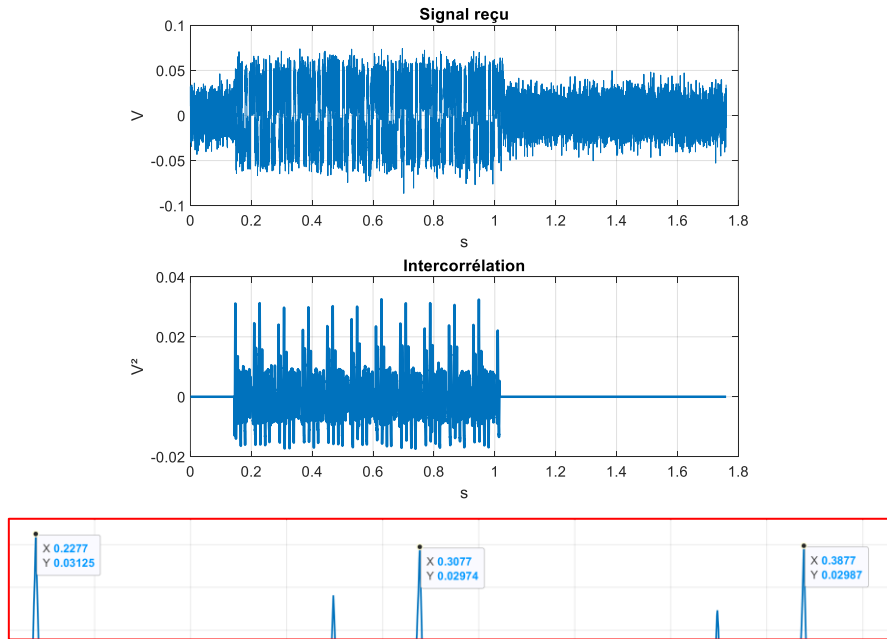


Figure 5. Intercorrrelation du signal utile avec le signal d'entête. Dans le cadre rouge, un zoom sur le signal d'intercorrrelation. On observe des maxima locaux régulièrement espacés de la durée d'une trame (80ms)

On observe bien sur la Figure 5 des maxima régulièrement espacés de la durée d'une trame $(N_h + N_d)T_b = 80ms$. Pour déterminer de manière robuste quel maximum relatif peut être pris comme marqueur d'un début de trame, on propose de sommer les valeurs de $C(m)$ décimées avec un pas $M = (N_h + N_d)T_b F_e$, correspondant au nombre d'échantillons dans une trame complète. Ainsi, les maxima correspondant bien à un début de trame vont s'additionner, réduisant le risque de sélectionner un optimum qui ne correspondrait en fait pas à un début de trame, à cause du bruit et de l'intercorrélation avec les données. Soit $C_e^{(m)}$ le vecteur des coefficients d'intercorrélation sous-échantillonnés à partir de l'instant mT_e :

$$C_e^{(m)} = \{C(m + kM), m + kM < n_2\}, m \in [n_1, n_1 + 2M]$$

On calcule la moyenne de chaque vecteur

$$CS(m) = \frac{1}{|C_e^{(m)}|} \sum_k C_e^{(m)}(k)$$

Le coefficient $CS(m)$ maximal est trouvé à l'indice m_0 correspondant au début d'un header. La Figure 6 montre en rouge le signal $CS(m)$ superposé à $C(m)$, avec la flèche qui indique la position m_0 du maximum obtenu.

$$m_0 = \arg \max_m CS(m)$$

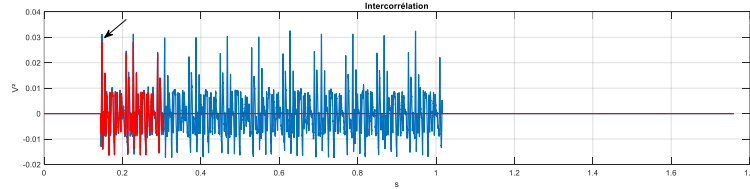


Figure 6. Sélection de l'indice de début de trame.

2.2. Décodage

Une fois le début de trame déterminé, on extrait le signal utile correspondant à une trame :

$$y(n) = x(m_0 + n), n \in [0, M[$$

Soit $n_e = T_b F_e$ le nombre d'échantillons par bit. On sous échantillonne le signal $y(n)$ au milieu de chaque bit et on seuille par rapport à 0 pour retrouver le train binaire de la trame :

$$b(k) = \begin{cases} 1 & \text{si } y\left(\frac{n_e}{2} + kn_e\right) \geq 0 \\ 0 & \text{sinon} \end{cases}, k = [0, (N_h + N_d)[$$

La Figure 7 montre la trame extraite, le signal estimé par seuillage, avec en rouge les instants d'échantillonnage pour estimer les bits reçus.

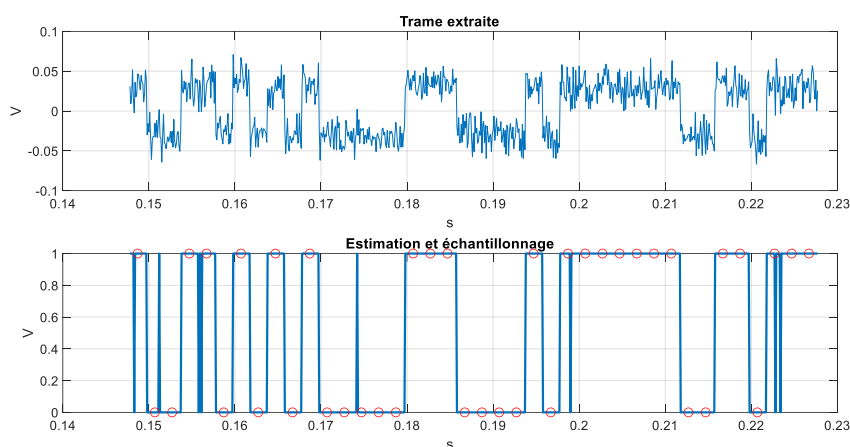


Figure 7. Décodage

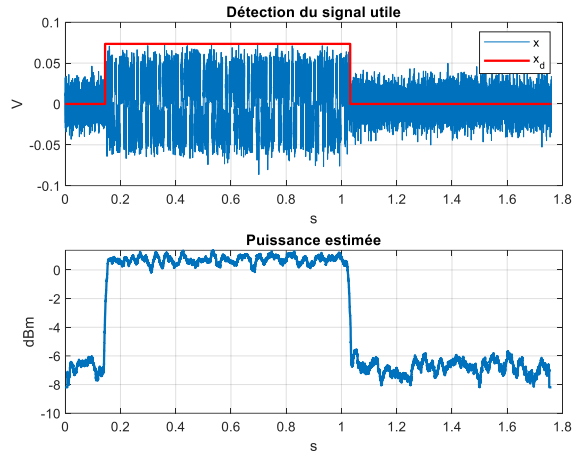
Dans cet exemple, on a décodé :

[1 0 0 1 1 0 1 0] [1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1]

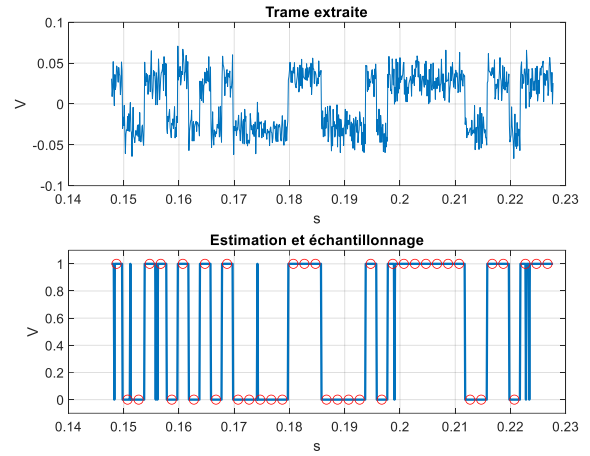
On a vérifié que les 40 bits de la trame ont été décodés sans erreur. On reconnaît notamment le header en grisé (8 premiers bits). On remarque cependant sur la Figure 7 que le bruit peut faire passer le signal de l'autre côté du seuil, conduisant à des erreurs de décodage si cela se produit au milieu d'un symbole, là où on rééchantillonne le signal.

3. Résultats expérimentaux

La méthode proposée a été implémentée sous Matlab. On l'a appliquée aux 5 signaux proposés (IR1.mat, IR2.mat, ..., IR5.mat). La Figure 8 montre les résultats obtenus. On remarque que la détection du signal utile fonctionne dans tous les cas, ainsi que la synchronisation. Le signal IR4 n'est pas décodé car il contient moins de 3 trames utiles. On a une erreur de décodage sur IR2, à cause du bruit à l'instant d'échantillonnage.



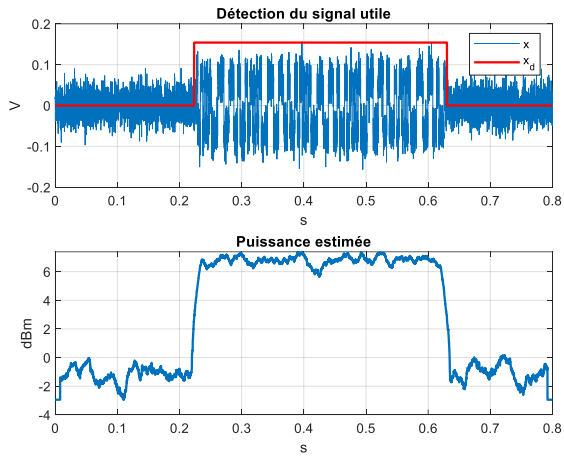
(a) IR1.mat



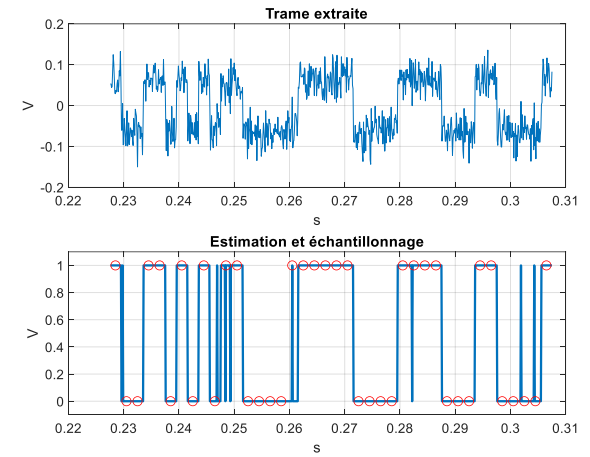
```

=== IR1.mat
Puissance du bruit ==-6.73 dBm
Signal utile entre t1 =0.14s (i1= 1445) et t2 = 1.03s (i2= 10317)
DECODAGE
Header décodé   = [1 0 0 1 1 0 1 0]
Header émis     = [1 0 0 1 1 0 1 0]
Données décodées = [1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1]
Données émises  = [1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1]
Erreurs commises = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
=> 0 bits erronés

```



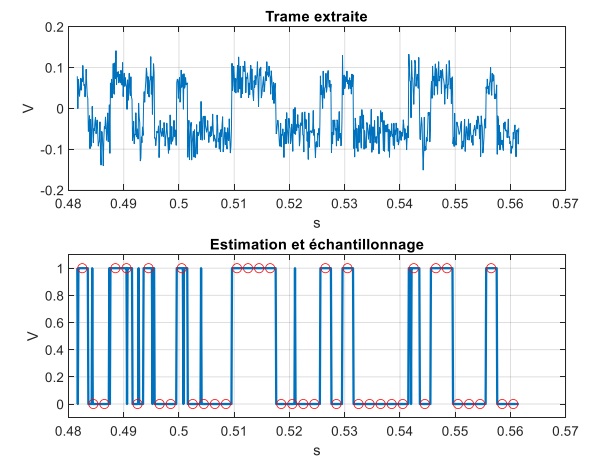
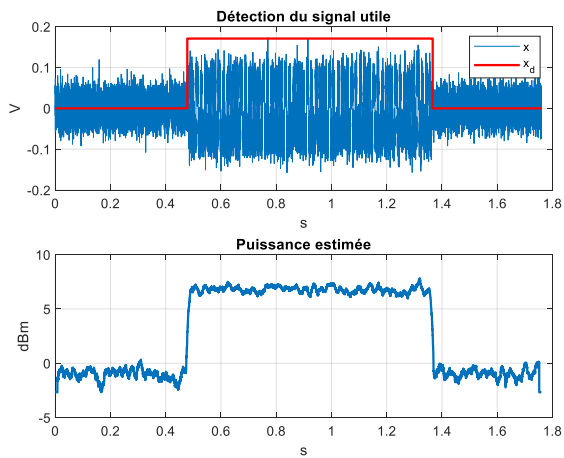
(b) IR2.mat



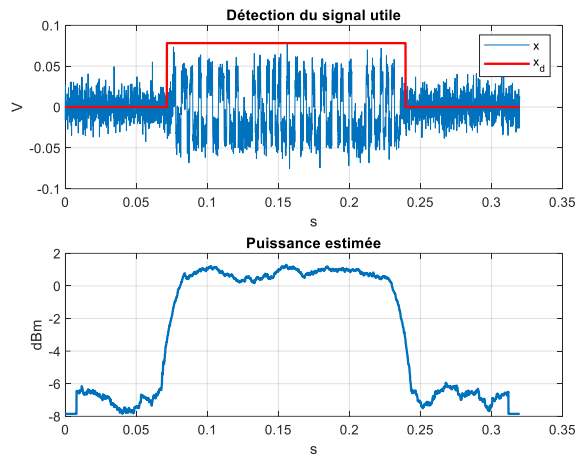
```

=== IR2.mat
Puissance du bruit ==-0.81 dBm
Signal utile entre t1 =0.22s (i1= 2236) et t2 = 0.63s (i2= 6303)
DECODAGE
Header décodé   = [1 0 0 1 1 0 1 0]
Header émis     = [1 0 0 1 1 0 1 0]
Données décodées = [1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 0 1]
Données émises  = [1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 0 1]
Erreurs commises = [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
=> 1 bits erronés

```



(c) IR3.mat

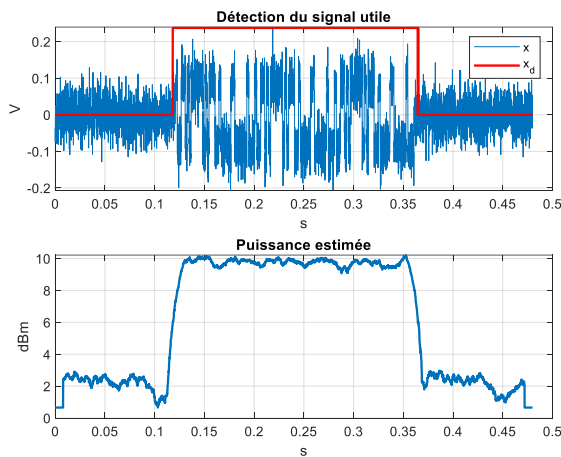


```

=== IR3.mat
Puissance du bruit = -0.91 dBm
Signal utile entre t1 = 0.48s (i1= 4783) et t2 = 1.37s (i2= 13665)
DECODAGE
Header décodé    = [1 0 0 1 1 0 1 0]
Header émis      = [1 0 0 1 1 0 1 0]
Données décodées = [0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0]
Données émises  = [0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0]
Erreurs commises = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
=> 0 bits erronés

```

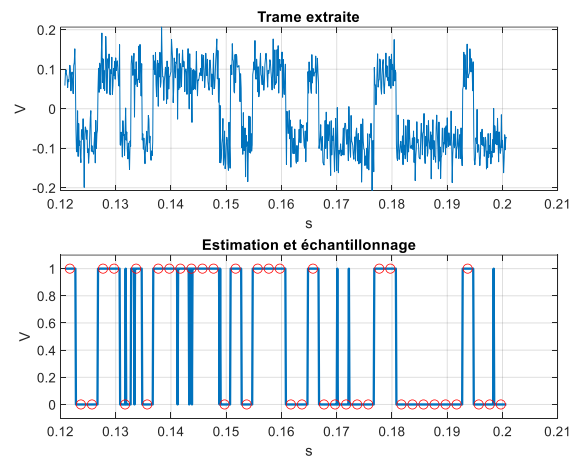
(d) IR4.mat



```

=== IR4.mat
Puissance du bruit = -6.89 dBm
Signal utile entre t1 = 0.07s (i1= 717) et t2 = 0.24s (i2= 2396)
ERREUR : signal non décodable184_end

```



(e) IR5.mat

```

=== IR5.mat
Puissance du bruit = 2.46 dBm
Signal utile entre t1 = 0.12s (i1= 1183) et t2 = 0.37s (i2= 3652)
DECODAGE
Header décodé    = [1 0 0 1 1 0 1 0]
Header émis      = [1 0 0 1 1 0 1 0]
Données décodées = [1 1 1 1 1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0]
Données émises  = [1 1 1 1 1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0]
Erreurs commises = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
=> 0 bits erronés

```

Figure 8. Résultats expérimentaux

4. Conclusion

On a réalisé un système de décodage d'un signal de télécommande IR, fondé sur la détection d'un signal utile par l'analyse de la puissance, la synchronisation sur l'entête grâce à l'analyse de l'intercorrélation avec le signal de header de référence, et enfin le décodage d'une trame par échantillonnage au centre des symboles NRZ et seuillage. Les résultats expérimentaux valident l'approche proposée. Cependant, l'étape de décodage peut être rendue plus robuste au bruit en appliquant un filtre adapté à la forme d'onde NRZ.