



Illustrations d'étapes simples de traitement du signal numérique à l'aide de GNU Radio et d'une SDR HackRF One

Sylvie Jarrix^{1,2*}, Annick Penarier^{1,2}

¹ IES, Université de Montpellier / CNRS, UMR 5214, 860 Rue de St. Priest, Bât. 5, 34090 Montpellier, France

² Département EEA, Faculté des Sciences, Campus Triolet, Université de Montpellier, France

*Auteur de correspondance : sylvie.guenard-jarrix@umontpellier.fr

Résumé : *Des étapes de traitement du signal numérique basique sont illustrées par simulations grâce au logiciel open source GNU Radio. Le même logiciel GNU Radio pilote une radio logicielle, un HackRF One. Ainsi les simulations sont complétées par des manipulations pratiques, permettant une prise en main des appareils usuels de mesure en salle TP. C'est une façon ludique de faire du traitement du signal sans avoir recours à des mathématiques ressenties souvent comme étant "complexes" par les étudiants.*

Mots clés : Radio Définie par Logiciel, SDR, Echantillonnage, Autocorrélation, Traitement du Signal, Shannon

Abstract: **Illustrations of simple digital signal processing steps using GNU Radio and a HackRF One SDR**

Basic digital signal processing steps are illustrated by simulations using the open-source GNU Radio software. The same GNU Radio software drives a software-defined radio, a HackRF One. Thus, the simulations are completed by practical manipulations, allowing a handling of the usual measuring devices found in an electronics lab. It is a fun way to perform signal processing without resorting to mathematics that students often experience as "complex".

Keywords: Software Defined Radio, SDR, Sampling, Autocorrelation, Signal Processing, Shannon.

1. Contexte des Travaux Pratiques de Traitement du Signal

L'enseignement de la thématique "Analyse du Signal" pose souvent des problèmes aux étudiants d'EEA. Pour la plupart d'entre eux, les points abordés semblent abstraits, souvent par manque de bagage mathématique solide. De ce fait l'introduction de Travaux Pratiques est en général bien accueillie et recherchée pour la compréhension des différents traitements qu'un signal peut subir. L'étudiant peut alors concrétiser dans son esprit les principes fondamentaux théoriques.

Dans cette optique de mise en évidence de traitements possibles et simples du signal numérique, nous nous sommes tournés vers le logiciel libre GNU Radio [1] (GNU's Not Unix). Celui-ci fournit directement des blocs préprogrammés de traitement du signal. A partir de tutoriels ou de vidéos disponibles sur internet nous avons rassemblé et complété plusieurs petites manipulations. L'objectif de ces TP est double : d'une part mettre en avant les principaux aspects lors de la numérisation d'un signal comme l'échantillonnage, avec ses limitations et ses besoins de filtrage, d'autre part profiter de GNU Radio et de sa capacité à contrôler une radio logicielle pour utiliser des équipements usuels de l'électronicien comme l'oscilloscope, le générateur basse fréquence (GBF) ou l'analyseur de spectre. L'idée n'est pas nouvelle, en témoigne la nombreuse littérature sur le sujet de la SDR appliquée à l'enseignement [2]. Plusieurs auteurs en effet ont déjà cherché à illustrer les éléments théoriques au moyen de simulations ou manipulations concrètes, souvent en lien avec tout un panel d'applications de transmission sans fil [3][4][5][6]. Néanmoins la plupart traitent très vite d'applications plus complexes combinant plusieurs fonctions comme des systèmes complets de transmission sans fil ou les modulations numériques [7][8][9].

Nous nous démarquons ici en reprenant des simulations ou manipulations beaucoup plus simples, axées vraiment sur les rudiments du traitement du signal. Ces TP se veulent accessibles à un public de niveau universitaire type licence 2^{ème} ou 3^{ème} année, sans avoir recours à de la programmation pour laquelle les étudiants ne sont souvent pas encore formés.

Nous débutons ce papier par la présentation du logiciel GNU Radio et de son interface graphique. Ensuite la radio définie par logiciel (Software Defined Radio, SDR) utilisée, le HackRF One, est détaillée. Nous expliciterons ensuite quelques manipulations de base illustrant les principes de process du signal numérique. Quelques applications concrètes d'utilisation du traitement du signal seront données, avant de conclure.

2. Le logiciel GNU Radio et le matériel associé : le HackRF One

2.1. Le Logiciel GNU Radio et son interface graphique

Le logiciel libre GNU Radio est au départ un système d'exploitation lancé en 1984 par Richard Stallman [10]. Il est maintenu par le Projet GNU [11], rassemblement de volontaires et de sociétés. GNU Radio comporte une interface graphique (Graphical User Interface, GUI) nommée GNU Radio Compagnon (GRC). Sur cette interface il suffit de positionner des "blocs" de fonctions dans lesquels sont déjà programmés les principaux traitements pour l'analyse du signal. Ces blocs Open Source sont basés sur une application QT ou WX, cette dernière n'étant plus maintenue. Ils sont écrits en langage C++ et sont tout à fait modifiables par l'utilisateur. Il est de ce fait également possible de créer de nouveaux blocs, adaptés à une fonction ou besoin particulier. Le graphe lui-même, constitué de l'ensemble de ces "blocs", est nommé "Flowgraph" et génère un fichier en langage Python interprété au fur et à mesure de son exécution. La facilité d'utilisation est compensée par le fait de devoir disposer d'un ordinateur

un peu rapide car l'interprétation d'une source prend plus de puissance machine qu'une simple exécution.

Comme dans la théorie des graphes, les "flowgraphs" s'apparentent aux graphes de flux par lesquels les données circulent. Chaque bloc est dédié à une tâche, de cette façon GNU Radio reste modulaire et flexible. Les blocs sont connectés entre eux par des ports. Un exemple de flowgraph est donné Figure 1. Dans cet exemple le premier bloc intitulé "Signal Source" génère un signal. Un tel bloc, comportant seulement des ports de sortie, est appelé une "source". Les derniers blocs du flowgraph à l'inverse ne contiennent pas de port de sortie, mais uniquement des ports d'entrée, ce sont des "sink". Dans cet exemple, nous avons un bloc de génération du signal, et deux blocs de visualisation, un dans l'espace temporel, l'autre dans l'espace fréquentiel. Dans cet exemple, les données circulent de gauche à droite. Le logiciel GNU Radio n'est pas conçu pour limiter son traitement en temps réel. Il fonctionne aussi vite que le processeur peut exécuter le code dans les différents blocs. L'exécution est ainsi opportuniste : si un bloc a suffisamment de données dans ses tampons d'entrée et assez d'espace pour écrire dans ses tampons de sortie, il sera programmé et s'exécutera dès que le processeur aura le temps de l'exécuter. Pour éviter la saturation du CPU de l'ordinateur, en l'absence de matériel physique associé comme une radio définie par logiciel (SDR), le bloc "Throttle" se chargera de fluidifier le flux d'échantillons. Enfin un bloc toujours présent par défaut dans les flowgraph et indispensable est le bloc "Variable" d'identifiant "samp-rate" qui permet de régler la valeur de la fréquence d'échantillonnage du signal.

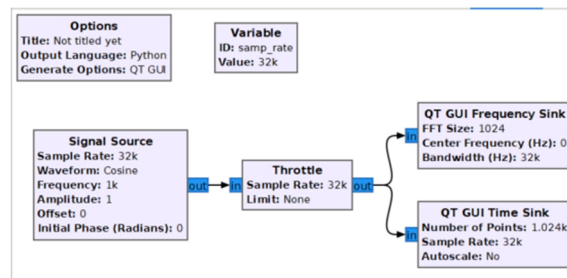


Figure 1. Exemple de flowgraph permettant de visualiser un signal issu du bloc "source" dans le domaine temporel et dans le domaine fréquentiel.

Tous les paramètres des différents blocs sont modifiables facilement en ouvrant ceux-ci avec un double click de la souris, comme le montrent les exemples de la Figure 2.

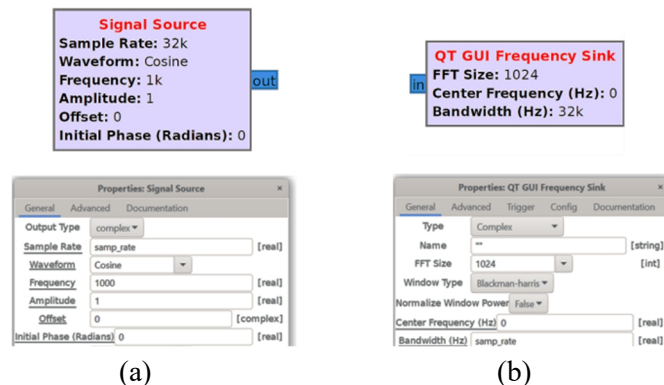


Figure 2. Exemple de paramètres modifiables sans programmation dans des blocs de fonction de l'interface graphique GNU Radio Compagnon. (a) bloc source, (b) bloc de visualisation

Les blocs les plus usuels pour un flowgraph simple sont donc de trois sortes. Le premier est le bloc "signal source" qui génère classiquement un signal sinusoïdal, carré ou triangulaire, puis les blocs de visualisation comme les blocs "QT GUI Time Sink", équivalent de l'oscilloscope,

ou "QT GUI Frequency Sink", qui affiche la densité spectrale du signal. Le troisième type de bloc sont les blocs de traitement mathématique comme une addition ou une multiplication.

Les données traitées pour une simulation d'application peuvent être au format "réel" ou au format "complexe" via une transformée de Hilbert [12] transparente pour l'utilisateur.

Ce logiciel permet de simuler des chaînes d'émission ou de réception radiofréquence (RF), mais aussi comme on le verra par la suite, des effets d'échantillonnage ou de Transformée de Fourier rapide sur un signal. Dans l'idée de compléter les simulations par des observations pratiques nous pouvons adjoindre au logiciel une radio définie par logiciel.

2.2. La Radio définie par logiciel

Les systèmes d'émission-réception radiofréquence sont révolutionnés aujourd'hui par les systèmes de radio définis par logiciel ou Software Defined Radio, (SDR) [13]. Ceci a été permis avec l'évolution de la technologie, notamment l'apparition d'échantillonneurs à haute fréquence permettant d'effectuer des traitements numériques autrefois réservés à la bande de base. La SDR est un système de radiocommunication qui utilise des composants logiciels reconfigurables pour le traitement et la conversion des signaux numériques dans une chaîne de radio transmission. Un exemple de comparaison de l'architecture entre une radio analogique et une radio SDR est donnée Figure 3.

Contrairement aux systèmes traditionnels, les SDR sont très flexibles et polyvalentes, permettant de changer de gamme de fréquence sans avoir besoin de changer des parties matérielles comme des filtres. Une SDR effectue ainsi des quantités importantes de traitement du signal à l'aide de paramètres pouvant être facilement reconfigurables par logiciel.

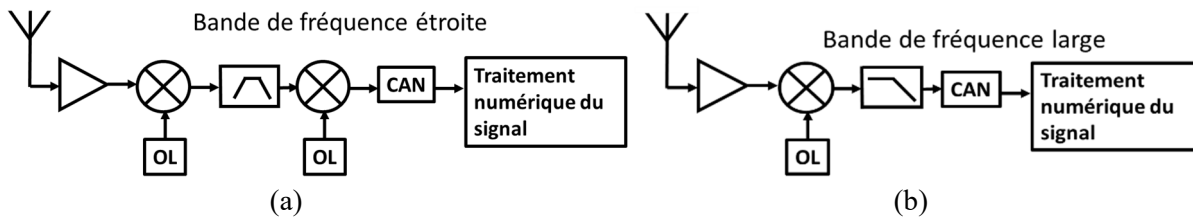


Figure 3. Schématisation simple d'une architecture d'un récepteur (a) hétérodyne conventionnel analogique (b) d'une radio logicielle.

L'architecture d'une SDR se compose ainsi généralement d'un front-end radiofréquence (RF), de convertisseurs analogiques à numérique (CAN), et de convertisseurs numérique à analogique (CNA) pour convertir la fréquence intermédiaire numérisée (IF) des signaux de -et vers - la forme analogique, respectivement. Le reste de traitement se fait alors en bande de base de façon numérique. La numérisation du signal intervient beaucoup plus tôt dans la chaîne qu'avec une radio classique, permettant par logiciel l'ajustement des différents filtres et mélangeurs en fonction de la bande de fréquence et du type de modulation souhaitée.

La SDR choisie ici est le module émetteur -récepteur HackRF One [14] de Great Scott Gadgets. Cette radio fonctionne avec des ondes porteuses sur une bande de fréquence allant de 1 MHz à 6 GHz avec 20 MHz de bande passante. Elle propose une communication en half-duplex avec un taux d'échantillonnage maximum de 20 Méga-échantillons par seconde et une résolution de 8 bits sur les voies en phase (I, In-phase) et en quadrature (Q, Quadrature). La puissance d'émission varie selon la bande de fréquence utilisée, comme indiqué dans le Tableau 1.

Tableau 1. Puissance d'émission maximale du module émetteur – récepteur HackRF One en fonction de la bande de fréquence utilisée.

Bande de fréquence de la porteuse (MHz)	Puissance (dBm)	Commentaire
1 – 10	5 à 15	Augmente avec la fréquence
10 – 2170	5 à 15	Décroît avec l'augmentation de la fréquence
2170 – 2740	13 à 15	Non répertoriée
2740 - 4000	0 à 5	Décroît avec l'augmentation de la fréquence
4000 – 6000	0 à -10	Décroît avec l'augmentation de la fréquence

D'autres SDR pourraient être pareillement utilisées à des fins d'illustration dans nos TP. L'USRP de Ettus Research [15] et LimeSDR de Lime Microsystems [16] présentent de meilleures performances en résolution et bande passante mais au prix d'un coût allant de deux à huit fois plus cher par rapport au HackRF One. Un équivalent en termes de prix est l'Adalm Pluto d'Analog Devices [17], qui monte moins haut en fréquence (3 GHz) mais propose du full duplex. En revanche il est peut-être moins simple d'utilisation. Les avantages de ces systèmes HackRF One ou Adalm Pluto est de permettre aussi bien l'émission que la réception de signal sur une gamme de fréquence étendue, mais l'avantage premier est d'être faible coût, environ 300 euros.

Pour le HackRF One la puissance maximale reçue est limitée à -5 dBm, ce qui le rend un peu fragile. Il suffit cependant de placer un atténuateur sur l'entrée pour éviter une dégradation due à une puissance détectée trop forte.

Le HackRF One est pilotable grâce au logiciel GNU Radio. Dédié aux communications RF, ce système reçoit et émet systématiquement un signal modulé sur porteuse. Ainsi le premier traitement pour une réception, ou le dernier pour une émission, est nécessairement une conversion de fréquence [18]. De ce fait il faut injecter par exemple en réception un signal utile déjà modulé sur porteuse HF pour étudier le signal utile, mais cela sera transparent pour les études en TP.

Dans l'interface GNU Radio Compagnon les blocs permettant d'utiliser le HACKRF One en émission ou en réception relèvent respectivement des blocs "Osmocom" source ou sink.

3. Manipulations de base du traitement numérique des signaux

Les Travaux Pratiques proposés s'intègrent dans un module d'analyse du signal numérique basique, enseigné en Licence EEA. Le programme couvert comprend l'échantillonnage, le produit d'intercorrélation, le produit de convolution, et la Transformée de Fourier Discrète amenant à la Transformée de Fourier Rapide (Fast Fourier Transform, FFT).

Une prise en main du logiciel est d'abord proposée au travers de la conception d'un flowgraph très simple comportant une source de signal sinusoïdal et la visualisation de sa FFT. Puis d'autres manipulations plus abouties sont prévues, elles sont décrites dans les sections suivantes.

3.1. Etude temps-fréquence dans le cas d'une FFT

Dans un premier temps l'aspect temporel et fréquentiel d'un signal est mis en avant. Les notations du cours théorique de licence associé à ce module TP sont les suivantes. Un signal temporel sera nécessairement périodique de période notée T , comportant N échantillons, N étant obligatoirement un multiple de 2 pour la mise en place de la FFT. La durée d'observation du signal, notée T_0 , doit comprendre normalement un nombre entier m de périodes T . La fréquence d'échantillonnage f_e du signal temporel correspond bien évidemment à la bande passante du signal fréquentiel issu du calcul de FFT. Cette valeur de f_e correspond à la variable "sample-rate" du bloc "variable" d'un flowgraph. Le flowgraph initial est simple, du type de celui de la Figure 1, avec un bloc signal source réglé sur un sinus de fréquence $f_0 = 10$ kHz, et une fréquence d'échantillonnage $f_e = 512$ kHz. La visualisation se fait en fonction de la fréquence (bloc "QT GUI Frequency Sink"), ou en fonction du temps (bloc QT GUI Time Sink). La simulation est lancée en activant le flowgraph. Les notions mises en évidence lors de la simulation sont les suivantes :

- Travail sur les unités logarithmiques,
- Lien entre la bande passante d'affichage d'une période de la FFT et la valeur de fréquence d'échantillonnage,
- Lien entre le nombre d'échantillons N , la fréquence d'échantillonnage f_e et la durée d'observation du signal temporel T_0 ,
- Lien entre un nombre entier de périodes T du signal temporel et la durée d'observation T_0 ,
- Nombre d'échantillons par période T du signal sinusoïdal en fonction du nombre N global d'échantillons.

3.2. Etude des paramètres d'une FFT

Dans un second temps, les paramètres propres à l'échantillonnage d'un signal lors de son utilisation lors d'un calcul de FFT sont étudiés. Les phénomènes suivants sont mis en exergue :

- Effet de la fréquence d'échantillonnage réduite à la valeur de Nyquist, à la fois sur le signal temporel et fréquentiel (cf. l'exemple Figure 4).
- Effet de l'abaissement du nombre N d'échantillons.
- Effets d'un nombre non entier de périodes T du signal temporel dans la durée d'observation T_0 . Ceci entraîne des instabilités au niveau temporel et des fréquences de pics erronées dans le domaine fréquentiel. Ces observations sont faites avec un signal généré en format complexe ou réel.

Des exemples de l'effet du choix de la fréquence d'échantillonnage f_e d'un signal sinusoïdal sont donnés Figure 4.

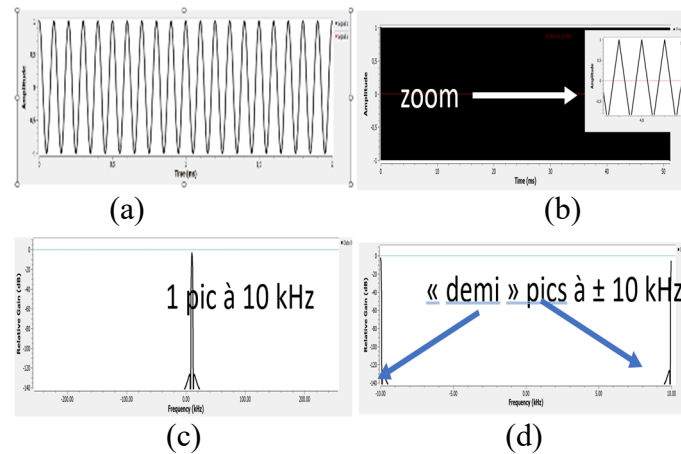


Figure 4. Exemple d'impact de la fréquence d'échantillonnage f_e sur un signal temporel et son spectre. (a) : fréquence d'échantillonnage correcte, signal temporel, (b) : fréquence d'échantillonnage égale à la fréquence de Nyquist, signal temporel. Encart : Zoom sur le signal temporel, (c) : fréquence d'échantillonnage correcte, spectre, (d) fréquence d'échantillonnage égale à la fréquence de Nyquist, spectre.

Enfin il paraît intéressant d'associer ces analyses à des manipulations en utilisant le HackRF One. Il faut alors remplacer le bloc "QT GUI Signal Source" de génération du signal sur le flowgraph par un bloc "Osmocom Source" réglé sur une porteuse f_p haute fréquence, par exemple $f_p = 20$ MHz. Un GBF envoie un signal sinusoïdal de fréquence $f = 20.01$ MHz à l'entrée de la SDR HackRF One. La SDR fera un abaissement de fréquence de 20 MHz, ainsi il ne restera plus qu'à observer le signal utile de 10 kHz comme précédemment dans la simulation, et de faire les mêmes observations sur l'interface graphique. Il est aussi possible à partir de la mesure sur l'interface de l'amplitude du signal temporel, et de l'amplitude connue du signal du GBF, d'évaluer le gain total de la chaîne d'émission de la SDR.

A l'inverse le même type d'étude peut être mené en utilisant la SDR comme récepteur. Dans ce cas le bloc de génération du signal sur l'interface graphique "QT GUI Signal Source" est remis en action pour délivrer un signal sinusoïdal en bande de base, et au travers du bloc "Osmocom Sink" la SDR délivre un signal sur l'oscilloscope et l'analyseur de spectre. Le spectre affiche un signal transposé en fréquence haute puisque la SDR génère automatiquement une porteuse, modulée par le signal en bande de base

3.3. Notion de densité spectrale

Le but ici est de faire travailler les étudiants sur la notion de densité spectrale à partir d'un signal généré par l'ensemble GBF et SDR.

Les différentes étapes de calcul implémentées sont les suivantes. D'abord le calcul du spectre du signal est effectué par le bloc "FFT" (cf. Figure 5). Un bloc multiplicateur par la constante $1/N$ permet la normalisation de la FFT. Puis le complexe conjugué est calculé grâce au bloc "complex conjugate". Ce dernier résultat est multiplié avec le spectre précédent. Le résultat de cette multiplication est ensuite traduit en module grâce au bloc "Complex to Mag". L'affichage se fait en dB à l'aide du bloc "Log10" qui effectue le calcul suivant : $n \cdot \log_{10}(\text{input}) + k$, avec input le signal en entrée de bloc. La densité spectrale résultante est affichée grâce au bloc "Vector Sink" en fonction du nombre d'échantillons.

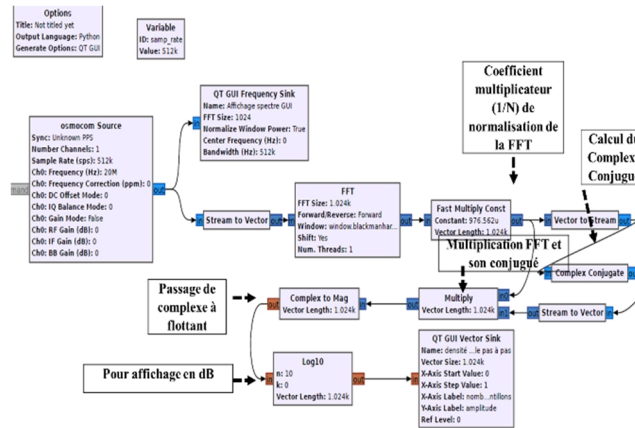


Figure 5. Flowgraph montrant les étapes de calcul d'une densité spectrale d'un signal numérique et comparé à une visualisation directe grâce au bloc "QT GUI Frequency" prêt à l'emploi.

On peut montrer l'équivalence de résultat de calcul de la densité spectrale entre le bloc utilisable tel quel "QT GUI Frequency Sink" et la mise en place pas à pas de calcul grâce au bloc "QT GUI Vector Sink". Un exemple pour un signal cosinus à la fréquence $f_0 = 10$ kHz et échantillonné à la fréquence $f_e = 512$ kHz est donné Figure 6. A noter que le dernier bloc de la chaîne de mise en en place de calcul de spectre, le bloc "QT GUI Vector sink", affiche le résultat en fonction du nombre d'échantillons N . On peut retrouver néanmoins la fréquence du pic du spectre en connaissant N et la fréquence d'échantillonnage f_e .

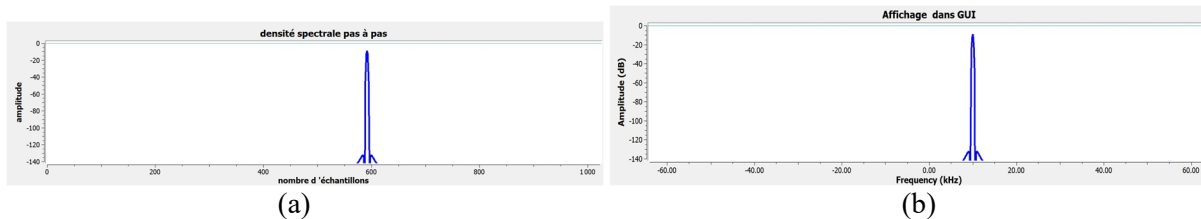


Figure 6. Densité spectrale d'un cosinus : a) implémentée pas à pas à l'aide de divers blocs de GRC et affichée en fonction du nombre d'échantillons, b) : calculée grâce au bloc "QT Gui Frequency Sink", affichée en fonction de la fréquence.

3.4. Réponse impulsionnelle

On cherche à montrer ici l'équivalence entre une réponse impulsionnelle traduite dans l'espace fréquentiel et une réponse harmonique. Une page wiki de GNU Radio indique comment tester la réponse impulsionnelle d'un filtre [19]. Un vecteur comportant un seul échantillon assimilable à une impulsion de Dirac est injecté en entrée d'un filtre passe-bas, et la fonction de transfert du filtre est visualisée directement grâce au bloc "QT GUI Frequency Sink".

Le résultat est ensuite comparé au diagramme de Bode du même filtre en injectant un signal sinusoïdal de fréquence variable en entrée de filtre. Dans le flowgraph un bloc "slider" permet de paramétrer les valeurs minimale et maximale de fréquence ainsi que le pas de variation. Une fois le programme lancé il suffit de choisir, comme sur un analyseur de spectre, une touche "max hold" et de changer la fréquence à l'aide du slider pour voir apparaître le gabarit du filtre sur toute la bande passante souhaitée. Un exemple de la comparaison est donné Figure 7

Erreur ! Source du renvoi introuvable..

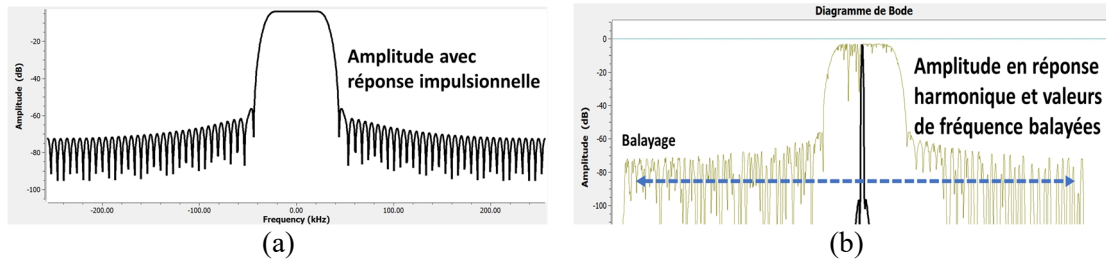


Figure 7. Gabarit d'un filtre passe-bas, (a) : Simulation de la réponse impulsionnelle d'un filtre passe-bas (b) : Simulation d'un diagramme de Bode sur le même filtre passe-bas.

3.5. Application : détermination du retard d'une fonction par le produit d'autocorrélation

Le produit d'autocorrélation est présenté dans le TP comme une application d'un problème de propagation en espace libre. On imagine un signal empruntant simultanément plusieurs chemins, ce même signal arrive au récepteur avec des délais temporels différents qu'il faut évaluer. La SDR est utilisée comme source (cf. Figure 8 **Erreur ! Source du renvoi introuvable.**). Le signal est séparé en deux chemins dont l'un passe par un bloc "délai" de valeur fixe, l'autre par un bloc "délai" de valeur variable. Ici encore un bloc "slider" permet de paramétrer les valeurs et le pas de changement du délai variable. Le bloc "délai" est défini en nombre d'échantillons. Le produit d'autocorrélation équivaut dans l'espace fréquentiel à la multiplication des spectres de chacun des signaux. On implémente ainsi après multiplication des spectres une transformée de Fourier inverse pour revenir dans le domaine temporel. Il suffit alors de faire glisser le slider et de repérer la valeur de délai amenant à un maximum du produit d'autocorrélation.

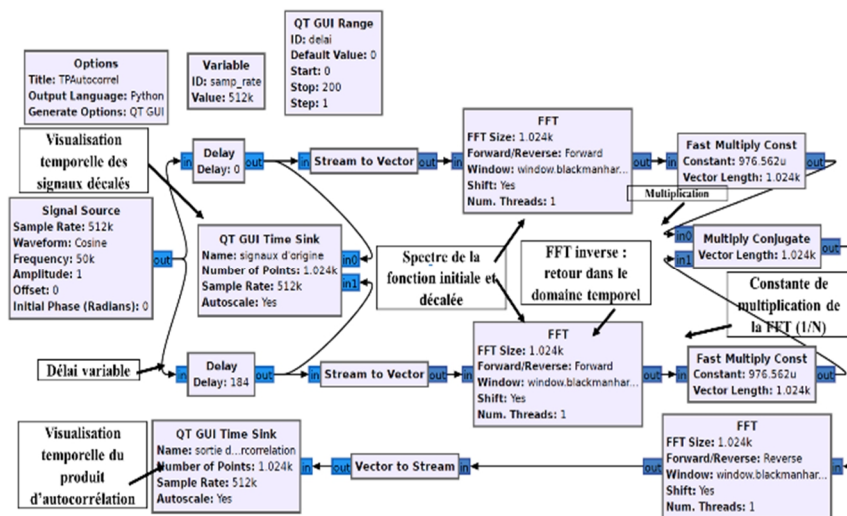
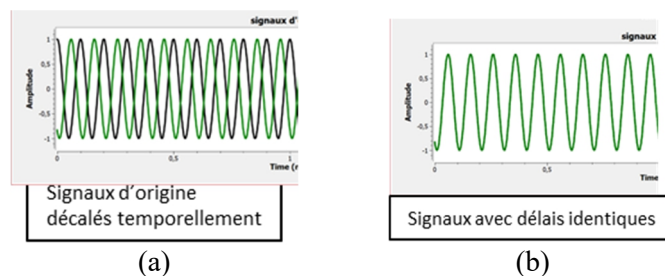


Figure 8. Flowgraph pour détermination du maximum d'un produit d'autocorrélation



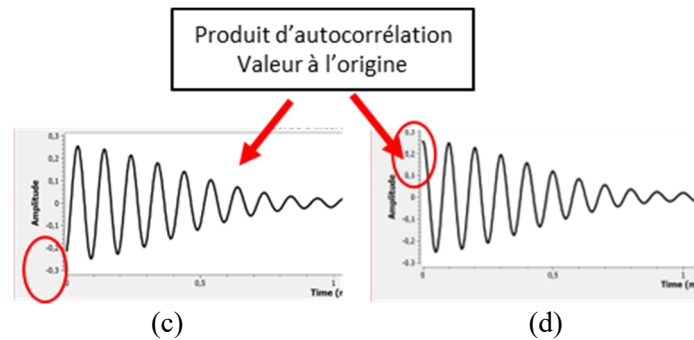


Figure 9. (a) Signaux décalés temporellement, (b) signaux superposés, (c) résultat du produit d'autocorrélation de deux signaux décalés temporellement (d) résultat du produit d'autocorrélation de deux signaux non décalés temporellement

Comme le montre la Figure 9 le maximum du produit d'autocorrélation a bien lieu au temps $t = 0$ et ce pour deux signaux non décalés temporellement. Le décalage temporel nul est confirmé par l'observation des deux signaux d'origine, la valeur du délai d'une des fonctions étant maintenant égale à la valeur de délai constante de l'autre signal.

4. Mise en place des TP et retour des étudiants

L'ensemble des simulations et manipulations n'ayant été conçus que cette année il n'a pas été implémenté dans son intégralité pour cette année universitaire. Une seule séance de 3h a été expérimentée au niveau Licence 3, touchant une cinquantaine d'étudiants. La séance portait sur l'observation des effets de changement de paramètres de la FFT. Le TP a été bien reçu par le public étudiant avec des retours positifs, au point que l'ensemble des étudiants a demandé à installer sur leurs ordinateurs personnels le logiciel GNU Radio. Les apprenants ont trouvé l'illustration des notions de cours intéressante de par la visualisation des phénomènes abordés en théorie, et complémentaires justement à ce cours. L'aboutissement allant jusqu'à l'achat d'une SDR par ces mêmes étudiants n'est pas envisageable, nécessitant en plus de la SDR des appareils de mesure ou des sources de signal qui ne sont pas à la portée d'un public étudiant. Le fait néanmoins d'utiliser ces appareils en complément en TP a montré une bonne implication des étudiants. Il est à noter que ces manipulations sont très peu onéreuses du fait de la gratuité du logiciel GNU Radio qui permet de mettre en évidence par simulation un panel assez large des effets de traitement du signal.

5. Conclusion

Le logiciel open source GNU Radio est utilisé pour mettre en évidence des étapes très simples et à moindre coût de d'analyse signal afin d'illustrer des cours de licence EEA. L'avantage de ce logiciel est de permettre de mettre en évidence rapidement les effets de traitement sur un signal numérique, et ce sans programmation, souvent source de problèmes chez les étudiants. Le passage temps-fréquence est montré, les paramètres d'un algorithme FFT et de leurs effets sont étudiés. Des applications sont aussi proposées comme la détermination du retard d'une fonction par rapport à une autre au travers d'un produit d'autocorrélation. Les simulations sont complétées par des manipulations à l'aide d'outils usuels de l'électronicien comme l'oscilloscope ou l'analyseur de spectre, grâce à l'utilisation d'une radio logicielle, elle-même pilotée par le logiciel open source GNU Radio.

6. Références

- [1] GNU Radio. *GNU Radio*, <http://www.gnuradio.org/>
- [2] C. Moy, *La radio logicielle : un sujet d'enseignement et un moyen d'enseigner les communications sans fil à l'université*, J3eA, vol. 21, p. 1008, 2022.
- [3] E. Monier et J-P. Barbot, *TD Gnu Radio*, Université de Bordeaux.
- [4] IUT de Toulon, Texte de TP *Émetteur et récepteur AM et FM*.
- [5] A. Morcos et A.-S. Grimault, *Compte-rendu du TP 02 : Modulations et démodulations numériques*, Master 2, Université de Paris-Saclay, (2019).
- [6] G. Goavec-Merou et J.-M. Friedt, *La réception radiofréquence définie par logiciel (Software Defined Radio – SDR)*, Magazine GNU/Linux. n°153, 2012.
- [7] A. Achouche & F Hachour, *Conception et Réalisation d'une chaîne de communication basée sur SDR à l'aide de l'USRP et GNU Radio*, Mémoire de Master, Faculté de Blida, Algérie, 2022.
- [8] G. Matten, J.-M. Friedt, E. Carry, *Prototypage de traitements numériques de signaux au moyen d'un outil libre : GNU Radio*, J3eA vol. 14, 2015.
- [9] Y. Ben Doudouh, S. Rouet, R. Ibrahim, J.-M. Feybesse, *Radio Logicielle*, Rapport de DU Réseaux et Télécoms, 2^{ème} année, Université de Paris 13, 2016.
- [10] R. M. Stallman, S. Williams, and Christophe Masutti, *Richard Stallman et la révolution du logiciel libre*. Editions Eyrolles, 2013.
- [11] *À propos du projet GNU - Projet GNU - Free Software Foundation*,” *Gnu.org*, 2017. <http://www.gnu.org/gnu/thegnuproject.fr.html>
- [12] J.-M. Friedt, *Quelques éléments de traitement de signaux échantillonnés en temps discret avec GNU Radio*, GNU/Linux. Magazine France n°221, 2018.

Biographie des auteurs

Sylvie Jarrix : Maître de conférences à l'Université de Montpellier au sein du département EEA, elle enseigne depuis la Licence 2^{ème} année jusqu'au Master 2 spécialité PHyS (Photonique Hyperfréquences et Systèmes de Télécommunications). Ses enseignements comportent des Cours, TD et TP en Traitement du Signal analogique et numérique, et en Cours et TP spécifiques sur la thématique des Radiofréquences. Elle est par ailleurs responsable de la salle de Travaux Pratiques dédiée aux Hyperfréquences et est à l'origine de la plupart des Travaux Pratiques sur cette thématique. Ses recherches à l'Institut d'Electronique et des Systèmes se concentrent pour partie sur la CEM et pour partie sur la caractérisation de matériaux par méthodes Hyperfréquences et Compatibilité Electromagnétique dans les systèmes.

Annick Penarier : Maître de conférences à la Faculté des Sciences et à l'Institut d'Électronique et des Systèmes de l'Université de Montpellier. Elle enseigne principalement les bases du traitement du signal ainsi que les micro-ondes, avec un accent particulier sur la simulation haute fréquence et les dispositifs et systèmes actifs en hautes fréquences. Elle est également responsable d'une plateforme technologique dédiée au développement de projets industriels, axés sur la conception et la réalisation de systèmes micro-ondes tels que les antennes et les radars, depuis l'étude initiale jusqu'au prototype intégrable.