Introduction aux techniques numériques utilisées dans les « *Software Defined Radios* (SDR) ».

1 INTRODUCTION

De nos jours (2024), la grande majorité des *transceivers*[1] commerciaux sont basés sur des techniques numériques et font appel à des DSP (*Digital Signal Processors*) ou à des FPGA[2], qui souvent contiennent un ou plusieurs DSP.

Bien que ces techniques soient généralement hors du domaine de compétences de l'amateur moyen (connaissances mathématiques, outils de développement, connaissances de programmation), il n'en demeure pas moins qu'il faut s'y intéresser au vu de leur large diffusion.

Une autre raison pour acquérir des bases dans ce domaine est que l'OFCOM[3] vient d'introduire, pour l'examen de radioamateur, plusieurs questions plutôt pointues sur des sujets en rapport avec la mise en oeuvre des DSP.

Ce document s'adresse donc aux amateurs désireux de trouver une introduction d'un niveau abordable aux techniques numériques, et aux candidats à l'examen de radioamateur désirant mettre toutes les chances de leur coté.

1.1 Avertissement

L'étude des techniques de traitement numérique du signal fait appel à des notions mathématiques pas nécessairement à la disposition du candidat typique à l'examen de radioamateur. C'est pourquoi cet aspect a été volontairement simplifié ici dans la mesure du possible. Les documents cités en références pourront être consultés pour plus de détails s'il y a besoin.

1.2 Vocabulaire

Une grande partie du - sinon tout le - vocabulaire technique en rapport avec les DSP existe en français, mais n'est pas couramment utilisé, du fait que la grande majorité des publications techniques en rapport est en anglais. Dans ce document, peu d'effort a été fait pour utiliser les termes français, quand cela n'est pas courant. Ainsi nous parlerons de techniques numérique, mais nous auront affaire à des *Digital to Analog Converters (DACs)*, et des *Digital Signal Processors* (DSP) par exemple. Les termes anglais seront cependant toujours écrits en *italique*. Typiquement lors de la première utilisation de termes anglais, leur traduction en français est donnée.

1. Émetteur-récepteur, de l'anglais ***trans****mitter-re***ceiver***.
2. *Field Programmable Gate Array.*
3. Office Fédéral de la Communication, organisme responsable pour les examens de radioamateurisme en Suisse.

2 CHAÎNE DE TRAITEMENT NUMÉRIQUE

Le monde qui nous entoure est quasiment à 100% analogique. Pour pouvoir traiter un signal numériquement, il faut le numériser (en faire un signal *digital*), on parle d'*Analog/Digital Converter* (ADC). Une fois numérisé, le signal peut être traité (transformé, manipulé, par exemple, filtrage, modulation, démodulation, compression ou décodage) dans un DSP, puis retourné sous une forme analogique dans un *Digital/Analog Converter* (DAC). Ceci est représenté sur la figure ci-dessous (nous parlerons des autres blocs sous peu).

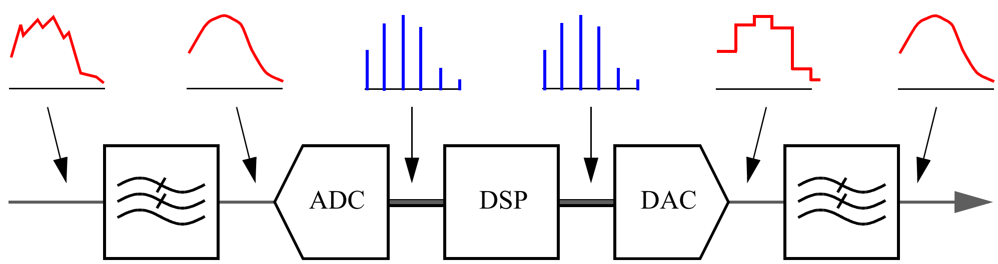

Figure 1 Chaîne de traitement numérique du signal. Le signal analogique passe d'abord à travers un filtre passe-bas pour éliminer toute fréquence au delà de la fréquence de Nyquist, puis le signal est converti en numérique (ADC). Après traitement dans le DSP, le signal numérique est envoyé sur un convertisseur digital/analogique (DAC), puis un filtre passe-bas en sortie reconstitue le signal analogique.

2.1 *Sampling* (échantillonnage), Nyquist et *aliasing* (repli)

Pour convertir un signal analogique en signal numérique, il faut l'échantillonner. Cela signifie qu'à intervalles réguliers et suffisamment rapprochés, on va déterminer sa valeur instantanée sous forme d'un nombre. Sur la Figure 2, les lignes verticales en pointillé représentent les moments d'échantillonnage du signal analogique (trait continu rouge), en bas sont les valeurs numériques équivalentes à chaque point, selon l'échelle de gauche.

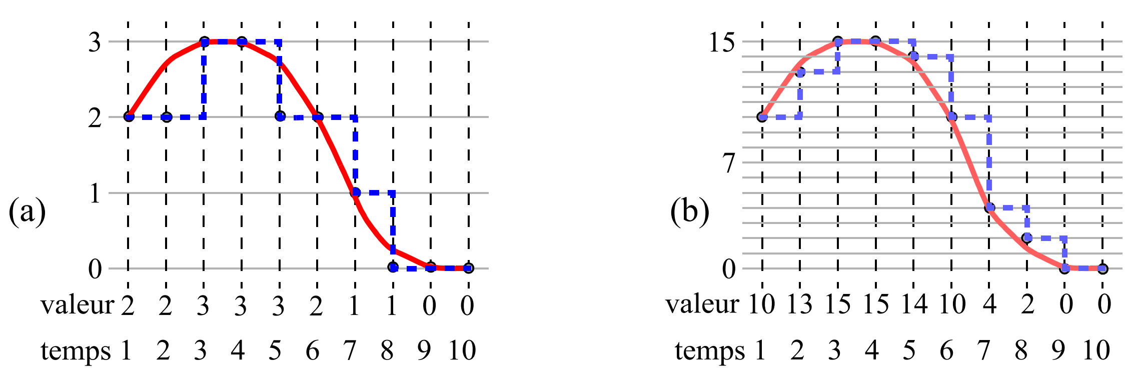

Figure 2 (a) Un signal est échantillonné sur 4 bits, le manque de précision est important. (b) Un plus grand nombre de bits diminue fortement les erreurs.

La Figure 2 représente un signal quelconque (en trait plein rouge) échantillonné pour chaque ligne verticale en pointillé, En (a) il n'y a que 4 niveaux d'échantillonnage, ce qui signifie que les valeurs numériques ne peuvent valoir que 0, 1, 2 ou 3. Au temps 1, la valeur déterminée est 2. Au temps 2, le signal n'a pas atteint le niveau 3, la valeur déterminée est encore 2. Au temps 3, le signal a atteint le niveau 3 et la valeur déterminée est 3. Au temps 4, le signal est encore au niveau 3. Au temps 5, le signal est inférieur à 3 et vaut donc 2, etc.

En pointillé bleu, le signal qui serait restitué dans ce cas (avant filtrage). En (b), il y a un plus grand nombre de niveaux d'échantillonnage. On remarque une amélioration du signal restitué.

En pratique on parle du nombre de bits d'une conversion : 8 bits permettent 256 niveaux, 10 bits 1024 niveaux, etc., il est rare qu'une telle conversion se fasse sur moins de 12 bits, les valeurs les plus communes étant 12, 14 ou 16 bits.

2.2 Conversion A/D

Un **convertisseur A/D** (ou **ADC** – *Analog to Digital Converter* soit convertisseur analogique/numérique) se présente généralement sous la forme d'un circuit intégré. Il y a deux paramètres principaux à considérer. Premièrement la cadence de conversion, qui n'est pas la même pour un signal BF (relativement lent) et un signal HF (qui nécessite clairement une plus grande rapidité), puis deuxièmement, le nombre de bits nécessaires pour une bonne reproduction du signal.

2.3 Vitesse d'échantillonnage

On peut aisément comprendre que si l'on échantillonne un signal très rapidement, on aura une reproduction sous forme numérique plus proche de la réalité. D'un autre coté, il y a des barrières technologiques à cette rapidité. Le problème est alors de déterminer à quelle vitesse minimale échantillonner un signal afin de pouvoir le reproduire après traitement numérique.

Considérons, à la Figure 3, trois signaux sinusoïdaux que nous échantillonnons. Chaque flèche correspond à un échantillon, et la valeur de chaque échantillon est indiquée par un cercle sur la sinusoïde. Ces échantillons ont été reproduits en bas de la figure, et une ligne dessinée, qui passe au centre de chaque cercle. Cette ligne représente le signal tel qui serait restitué après reconversion en analogique. Pour la première sinusoïde (a), on constate que la courbe du bas est identique à celle du haut, ce qui est le résultat désiré. En (b) une sinusoïde de fréquence supérieure à la première donne toujours un résultat adéquat après reconstitution. Cependant en (c), la courbe qui passe pourtant par les points échantillonnés n'est pas identique à la courbe originale.

En fait la fréquence du signal échantillonné est trop élevée en regard de la rapidité de l'échantillonnage (ou ce qui revient au même, la rapidité de l'échantillonnage est insuffisante). Il est de ce fait impossible de reconstituer le signal original, pire on reconstitue un signal différent, ce qui n'est pas acceptable. Ce phénomène s'appelle le **repliement** ou l'***aliasing***.

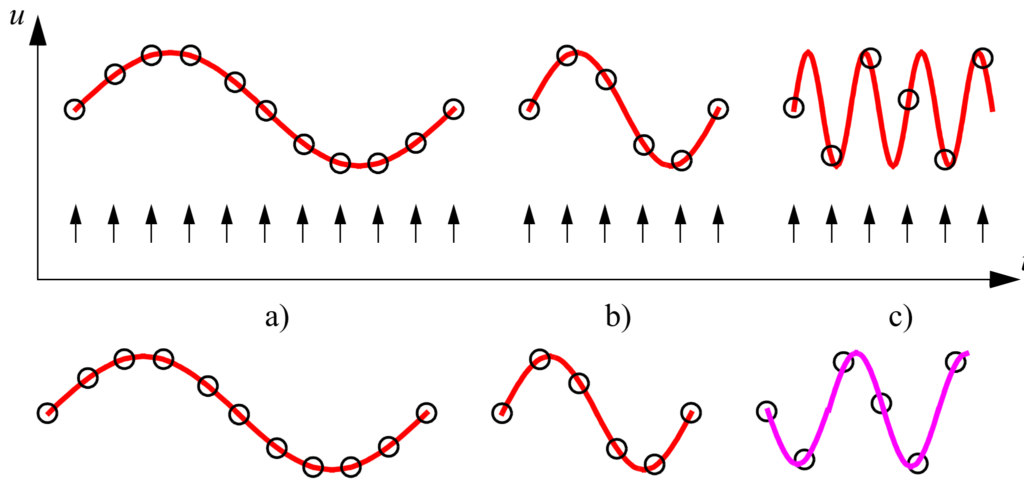

Figure 3 Trois signaux de fréquences croissantes sont échantillonnés à chaque petite flèche. Les signaux reconstitués à partir de ces échantillons sont reproduits dessous. On constate que le signal de la fréquence la plus élevée n'est pas reproduit correctement, ceci est le phénomène de repliement.

Il est impératif d'empêcher ceci, et la méthode usuelle est d'utiliser un filtre passe-bas, avant l'échantillonnage, pour éliminer les fréquences élevées qui ne pourront pas être reconstituées convenablement. Ce filtre porte naturellement le nom de **filtre antirepliement** (***anti-aliasing filter***).

En théorie, il faut échantillonner au minimum à une fréquence double de la fréquence la plus haute à convertir (théorème de Nyquist-Shannon), ce qui fait que la fréquence maximale que l'on peut échantillonner est de la moitié de la fréquence d'échantillonnage. En pratique, on préfère rester quelque peu en dessous de cette fréquence limite dite **fréquence de Nyquist**.

2.4 Conversion D/A

Après traitement numérique, il faut généralement retourner le signal sous sa forme analogique, par exemple pour amplification et reproduction par haut-parleur. Cette opération inverse de la précédente est appelée naturellement conversion D/A (*Digital* vers Analogique) et s'effectue au moyen d'un Convertisseur D/A (**DAC** – *Digital to Analog Converter* soit convertisseur numérique/analogique).

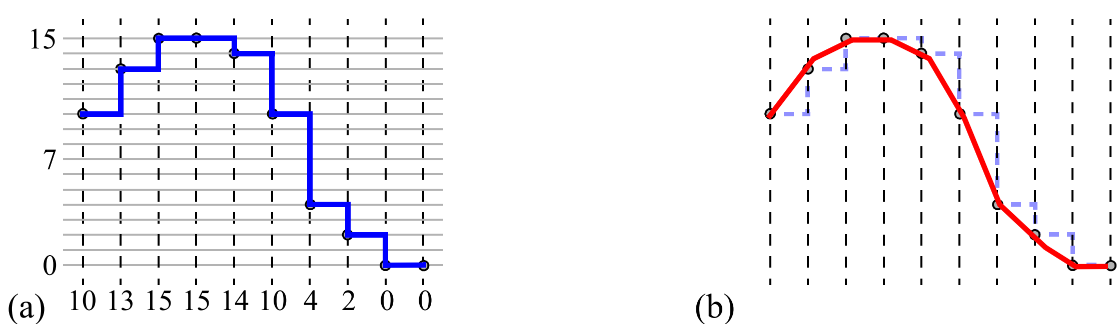

Figure 4 En (a), le signal tel qu'on le retrouve à la sortie du convertisseur D/A. En (b) le même signal après filtrage.

Ici les choses sont plus simples. Il suffit de convertir chaque valeur numérique en sortie du processeur, puis de faire suivre cette conversion d'un filtre passe-bas destiné à éliminer les effets de la numérisation.

Considérons le même signal que celui de la Figure 2 ci-dessus. Après conversion, il prend la forme de la Figure 4a. Cependant nous savons que ces petits escaliers sont une indication que le signal contient des fréquences élevées, que l'on peut aisément supprimer au moyen d'un filtre passe-bas (b).

3 ADC (*ANALOG TO DIGITAL CONVERTERS*)

Voyons quelques aspects plus avancés des ADC.

Il existent plusieurs techniques pour mettre en oeuvre une conversion A/D. Une des plus simple, surtout à comprendre est le convertisseur *flash* décrit ci-dessous.

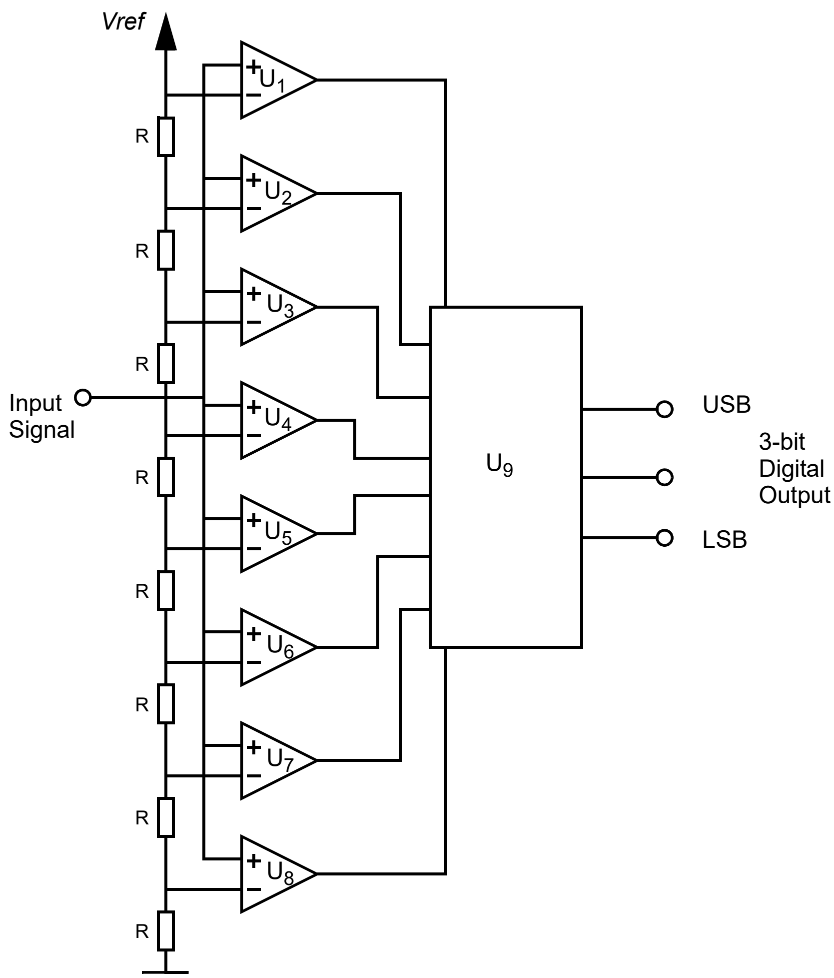

Figure 5 Exemple d'un convertisseur A/D *flash* à 8 bits. Une série de résistances agit comme un diviseur de tension entre la masse et la tension de référence *Vref*. Ces tensions issues du diviseur servent de référence à une série de comparateurs, qui fournissent une valeur binaire sur 8 bits, équivalente à la tension d'entrée. Finalement ce mot binaire est encodé en binaire sur 3 bits ($2^3 = 8$) en sortie.

3.1 Référence

Les ADC ont besoin d'une tension de référence stable. Cette tension est divisée intérieurement par le nombre de pas du convertisseur. Ainsi un convertisseur à 10 bits (1024 pas), utilisera 1/1024 de sa tension de référence par pas et sera capable de numériser un signal entre zéro volt et sa tension de référence. Dans l'exemple ci-dessous, nous avons 8 pas et si la référence *Vref* est de 3,2 V, chaque pas vaut 0,4 V. Une tension de 1,5 V donnera à l'entrée de U_9 (*8-bit binary encoder*) « 00000111 » puisqu'elle enclenchera les comparateurs U_8, U_7 et U_6. Après encodage, U_9 fournira en binaire « 011 » ce qui correspond à « 3 » en décimal (ou hexadécimal).

De même, une tension de 3,2 V enclenchera tous les comparateurs, résultant en « 11111111 » à l'entrée de l'encodeur et « 7 » à sa sortie, le maximum que le circuit peut gérer. Une tension supérieure à 3,2 V ne changera rien à la sortie, le convertisseur est saturé. Une autre façon de décrire ceci est que la dynamique du convertisseur est de 3,2 V au maximum. Cette tension maximale qui limite la dynamique du convertisseur vers le haut, immuable pour un convertisseur donné est nommée 0 dBFS (dB *full scale* - soit dB à pleine échelle). Cela implique que l'ADC ne pourra pas traiter des signaux supérieurs à 0 dBFS.

Une plus grande tension de référence augmenterait la dynamique, mais au détriment de la résolution du convertisseur, puisque maintenant les pas seraient plus importants. Or la résolution du convertisseur est une spécification importante, sinon des faibles variations du signal ou simplement des signaux faibles ne seront pas numérisés et par conséquent perdus.

Pour résumer, on voit que :

- La dynamique dépend de la tension de référence, mais cette dernière ne peut pas être augmentée indéfiniment, en tous cas pas au delà de ce que la technologie des circuits utilisés le permet.
- La résolution dépend du nombre bits, mais ce dernier ne peut pas être augmenté au delà d'un certain nombre, limité essentiellement pas le coût.
- On voit donc que le signal minimal qui peut être numérisé dépend du nombre de bits du convertisseur, et de la tension de référence.
- La vitesse d'échantillonnage doit être de plus du double de la fréquence la plus élevée numériser, mais là aussi on se heurte à des facteurs liés au coût pour les très hautes vitesses. De surcroît, plus la vitesse d'échantillonage est élevée, moins il y a de temps disponible pour que le DSP puisse faire les calculs requis, comme nous le verrons ci-dessous.

3.2 Dithering

Examinons la situation où un signal de faible amplitude doit être numérisé. Les 5 niveaux représentés ici (Figure 6), sont les 5 du bas (donc d'un niveau très faible) pour un total de 4096 (12 bits) ou 65536 (16 bits), donc les amplitudes ont été exagérées ici pour les besoins de la cause.

En a) sur la figure ci-dessous, le signal analogue rouge est numérisé. Comme ses variations d'amplitudes sont faibles, le signal digital (en bleu) est essentiellement plat. Un tel signal plat est à mettre en comparaison avec un signal carré qui comporte des zones plates. Or nous savons qu'un signal carré comporte nombre d'harmoniques impaires, entres autres 3 et 5 avec une amplitude non négligeable.

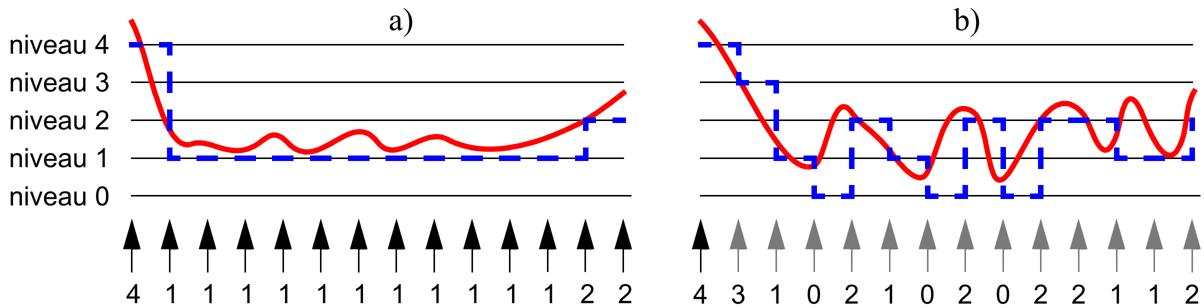

Figure 6 En a), un signal de faible amplitude (en rouge) avec de faibles variations est numérisé, mais comme ses variations sont trop faibles pour s'étendre sur plusieurs niveau de quantification, le signal numérisé (en bleu) est essentiellement plat. En b), le même signal, avec la même amplitude moyenne, mais avec une petite quantité de bruit superposée. Maintenant, le signal s'étend sur 3 niveaux. Il n'y a plus de plat sur le signal numérisé, donc plus d'harmoniques de rang impaires. De surcroît, on a maintenant gagné en précision puisque le niveau moyen n'est plus « 1 », mais plus près de la réalité à « 1,308 », au détriment d'une petite dégradation du rapport signal/bruit.

Cette distortion est appréciable et pour y remédier on introduit dans le signal analogue une petite quantité de bruit. Comme on le voit sur la Figure 6b, ceci a pour effet de faire sauter le signal continuellement entre 2 ou 3 niveaux. La valeur moyenne reste la même, mais il n'y a plus de plat et donc plus de génération d'harmoniques indésirables. Bien entendu, l'introduction de bruit réduit un peu le rapport signal/bruit, mais c'est un petit prix à payer pour l'élimination de la distortion harmonique.

L'ajout de bruit dans le signal à numériser est appelé *dithering* d'un mot anglais signifiant hésitation, souvent en rapport avec une décision à prendre. Ici, la décision est : quel niveau donner à un signal entre 2 niveaux de quantification (2 niveau digitaux adjacents).

Un autre bénéfice lié au *dithering* est une amélioration statistique de la précision. En effet, subjectivement, le signal en a) vaut ~1,3 mais est quantifié à 1. Si l'on fait la moyenne des échantillons (du second au dernier - gris) en b), on trouve 1,308) !

En résumé : Le *dithering* consiste à ajouter une petite quantité de bruit dans le signal à numériser afin de réduire la distortion harmonique introduite par les erreurs de quantification pour des signaux d'une faible amplitude.

4 PROCESSEUR NUMÉRIQUE DE SIGNAUX (DSP)

Le traitement du signal sous forme numérique s'effectue généralement au moyen d'un microprocesseur spécialisé appelé DSP – *Digital Signal Processor* (processeur numérique de signaux). La différence principale entre un microprocesseur et un DSP est que ce dernier est optimisé pour effectuer rapidement des calculs (multiplications, additions) en temps réel sur un flux continu de données, alors qu'un microprocesseur est plutôt orienté vers les mouvements de données et des calculs plus généraux.

L'opération la plus commune pour un DSP est une multiplication suivie d'une addition. Cette opération appelée MAC (*Multiply/Accumulate*) est répétée des dizaines voir des centaines de fois ou plus pour chaque échantillon lors d'une opération de filtrage. C'est pourquoi la vitesse d'un DSP est souvent donnée par le temps nécessaire pour cette opération. Un DSP moderne (en 2024, mais ça change très vite) peut effectuer une tel opération en 3 ns, soit 333 millions de MACs/s[1].

4.1 Décimation, interpolation

S'il est important d'échantillonner un signal à haute vitesse, cela peut générer plus d'échantillons que nécessaires pour certaines fonctions. La **décimation** consiste à éliminer (jeter) un certain nombre d'échantillons quand ils ne sont pas ou plus utiles. De plus ceci permet d'avoir plus de temps entre les échantillons conservés pour effectuer les calculs requis. L'opération inverse, l'ajout d'échantillons, quelquefois nécessaire, s'appelle l'interpolation, en effet, les échantillons « manquants » sont calculés en fonction des échantillons présents par interpolation.

4.2 Convolution

La convolution est la multiplication de 2 fonctions. C'est la base des opérations de filtrage dans les DSP, en fait c'est l'opération la plus importante en traitement du signal dans les DSP. Les 2 fonctions dont il est question sont d'une part le flux de données à traiter et d'autres part une série de coefficients représentant par exemple une fonction de filtrage à effectuer par le DSP. Nous entrerons dans plus de détails quand nous parlerons des filtres numériques. Examinons un cas simple de convoluteur à 6 coefficients.

Sur ce schéma (Figure 7), l'entrée du flux de données se fait en $x(t)$ (valeur x au temps t). La notation z^{-1} signifie simplement « échantillon précédent », ceci est dénoté par $x(t-1)$, $x(t-2)$, etc. Ici $x(t-5)$ est le premier échantillon entré dans le filtre et il est suivi de 5 autres. Ce flux continuel dont nous considérons la valeur à chaque instant d'échantillonnage et la première des 2 fonctions de notre convolution.

b_0 à b_5 sont les coefficients du filtre, en termes de convolution, c'est notre seconde fonction.

1. Source Wikipedia.

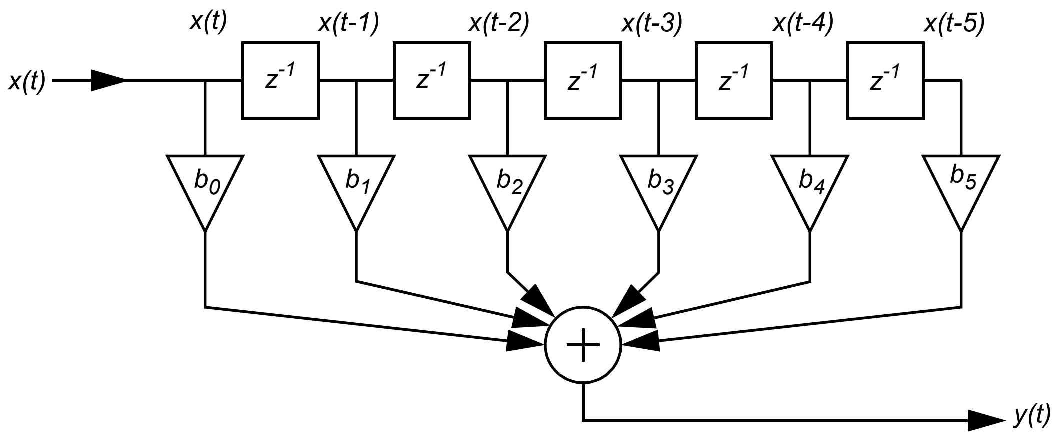

Figure 7 Convoluteur à 6 coefficients. Le flux avance à chaque pas dans les registres z^{-1} (z^{-1} représente une unité de temps). Ce flux d'entrée en $x(t)$ est multiplié à chaque pas par les coefficients b_0 ... b_5, puis pour chaque pas la somme de ces multiplications fournit le flux de sortie $y(t)$.

Pour chaque nouvel échantillon qui entre dans le filtre, la sortie $y(t)$ est la somme des 6 échantillons dans le filtre multipliés par les coefficients b_0 à b_5 :

$$y(t) = x(t) \cdot b_0 + x(t-1) \cdot b_1 + x(t-2) \cdot b_2 + x(t-3) \cdot b_3 + x(t-4) \cdot b_4 + x(t-5) \cdot b_5$$

4.3 Quelques cas particuliers de convolution

Considérons que les coefficients b_0 à b_5 sont : 1; 0; 0; 0; 0; 0.

Seul l'échantillon $x(t)$ se retrouvera en sortie, puisque les autres coefficients sont tous à zéro. Dans ce cas la sortie $y(t)$ suit l'entrée $x(t)$.

Modifions les coefficients b_0 à b_5 comme suit : 0; 1; 0; 0; 0; 0.

Seul l'échantillon $x(t-1)$ se retrouvera en sortie, puisque les autres coefficients sont tous à zéro. Dans ce cas la sortie $y(t)$ suit l'entrée $x(t)$ avec un décalage de 1 échantillon (1 pas, 1 élément de temps).

Encore quelques cas intéressant valent la peine d'être examinés. Utilisons les coefficients : 1; 0; 0; 0; 0; 0,5.

Le signal d'entrée se retrouve intégralement en sortie, puis après un décalage (retard) de 5 coefficients le signal se répète en sortie mais avec une amplitude moindre : nous avons créé un écho !

Avec les coefficients : 0,5; 0; 0; 0; 0; 0. nous avons un atténuateur puisque le signal en sortie est la moitié du signal en entrée.

Bien entendu avec les coefficients : 1,5; 0; 0; 0; 0; 0. nous avons un amplificateur.

4.4 Réponse impulsionnelle (*impulse response*)

Comme son nom l'indique, on s'intéresse ici au résultat de la convolution d'un signal d'entrée et d'une impulsion unique.

Maintenant modifions les coefficients comme suit : 0,10; 0,15; 0,25; 0,25; 0,15; 0,10; et utilisons comme flux d'entrée la séquence 0; 0; ... 0; 1; 0; 0; ... 0. ceci est illustré sur la figure ci-dessous.

Tant que le flux d'entrée est 0, la sortie est aussi à zéro. Dès que le 1 atteint $x(t)$, la sortie est à 0,10. Au temps suivant, le 1 s'est déplacé en $x(t-1)$ et la sortie vaut 0,15. Quand le 1 arrive en $x(t-2)$, la sortie vaut 0,25; et ainsi de suite jusqu'à ce que le 1 disparaisse après $x(t-5)$.

Le résultat obtenu, qui représente en sortie les coefficients du filtre est la réponse impulsionnelle du filtre (réponse du filtre à une impulsion).

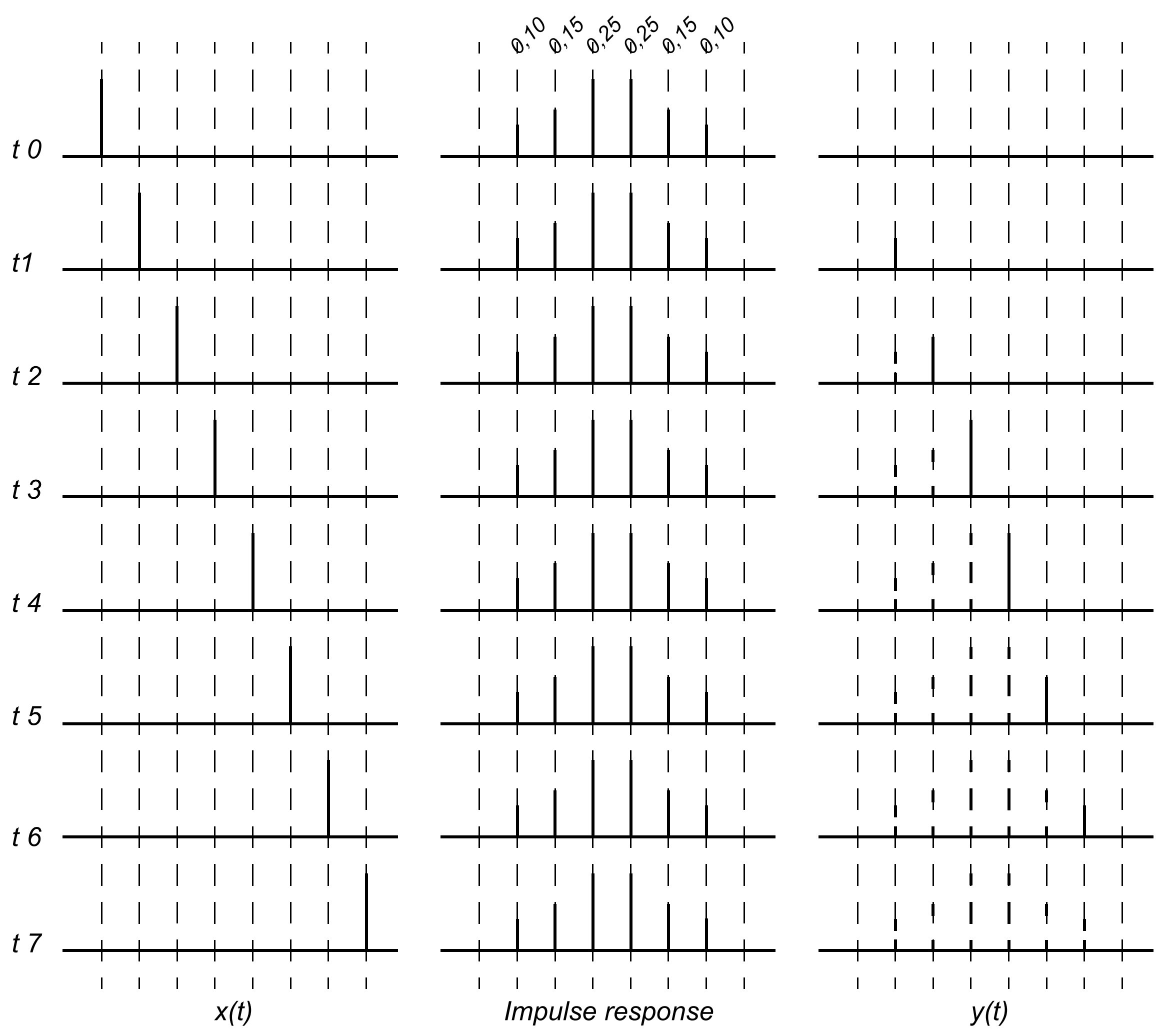

Figure 8 Mise en évidence de la réponse impulsionnelle d'un convoluteur. Une impulsion avançant dans un convoluteur met en évidence sa réponse impulsionnelle.

5 FILTRES NUMÉRIQUES

Les filtres sont utilisés essentiellement pour séparer des composants spectraux (séparer des signaux de fréquences différentes ayant été mélangés) et pour la restauration de signaux ayant été distordus (éliminations de signaux indésirables).

La raison principale du succès des DSP est leur capacité à réaliser des filtres avec des performances sans comparaisons avec leurs équivalents analogiques. Il est ainsi possible de réaliser des filtres avec une bande de transition extrêmement étroite entre la bande passante et la bande rejetée (par exemple passe-bas), pouvant s'étaler sur moins de 1 Hz !

5.1 Filtres FIR

Il existe 2 catégories principales de filtres en technique DSP, les filtres FIR (*Finite Impulse Response filters* - Filtres à Réponse Impulsionnelle finie - RIF en français) et les filtres IIR que nous verrons dans la section suivante.

L'architecture d'un filtre FIR est exactement le convoluteur décrit ci-dessus. Typiquement un filtre FIR comporte plus que 6 pas. Le terme technique anglais pour le nombre de pas d'un filtre est *tap* (que l'on pourrait traduire par prise). On peut aussi parler du nombre d'étages d'un filtre ou de son nombre de coefficients.

Typiquement les filtres FIR peuvent avoir de quelques dizaines de coefficients à quelques centaines. En règle générale, plus il y a de coefficients, plus on peut réaliser un filtrage efficace, mais ceci au détriment du temps de calcul.

Ces filtres sont dits à réponse impulsionelle finie car si l'on injecte une impulsion à l'entrée du filtre, une fois qu'elle a atteint le dernier *tap*, elle disparaît et n'a plus d'effet sur la sortie (voir par exemple Figure 7 ci-dessus). Ce n'est pas le cas des filtres IIR.

La topologie FIR n'a pas d'équivalent dans le monde analogique.

Avantages :

- Les filtres FIR sont inconditionnellement stables,
- Ils ont une réponse en phase linéaire, c'est à dire qu'ils retardent tous les composants spectraux de la même quantité.
- Du fait de leur topologie, on peut plus facilement leur donner une réponse en fréquence arbitraire.

Inconvénients :

- Ils sont plus gourmands en mémoire que les filtres IIR.
- Avec un plus grand nombre de coefficient, leur temps de latence[1] augmente, ce qui peut être prohibitif pour certaines applications.

1. temps de calcul.

5.2 Filtres IIR

Les filtres IIR (*Infinite Impulse Response* - filtres à Réponse Impulsionnelle Infinie RII en français) représente l'autre grande catégorie de filtres numériques. Ils utilisent la récursion plutôt que la convolution puisque une partie du signal de sortie est réutilisée pour les opérations de filtrage.

Il existe plusieurs topologies de filtre IIR, cependant toutes ont en commun le fait qu'en théorie, leur réponse impulsionnelle est infinie.

La topologie sur la Figure 9 ci-dessus est appelée *bi-quad*. Il est possible de cascader plusieurs *bi-quads* pour une réponse en fréquence avec une bande de transition plus rapide.

Pour chaque nouvel échantillon $x(t)$ qui entre dans ce filtre, la sortie $y(t)$ est donnée par :

$$y(t) = x(t) \cdot b_0 + x(t-1) \cdot b_1 + x(t-2) \cdot b_2 - y(t-1) \cdot a_0 - y(t-2) \cdot a_1$$

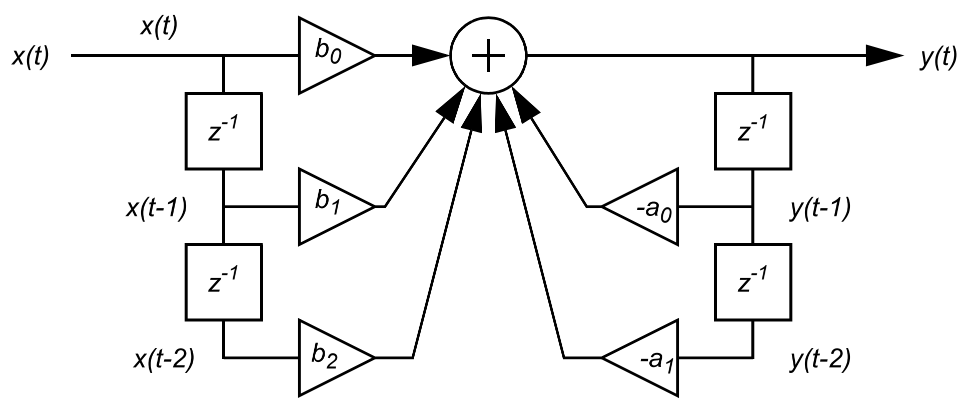

Figure 9 Filtre IIR. Cette topologie est appelée *bi-quad*. D'autres topologies existent. Dans les topologies IIR, comme on le voit ici, les valeurs de sortie, $y(t)$, sont utilisée dans un boucle de rétroaction (récursion).

On remarque que contrairement aux filtres FIR, Les valeurs en sortie $y(t)$ sont réutilisées pour le calcul des valeurs de sorties suivantes, et donc qu'en théorie, une impulsion en entrée continue à avoir de l'effet sur la sortie pendant un temps infini. En pratique ce n'est bien entendu pas le cas, car pour chaque itération, sont effet est moindre, sinon le filtre ne serait pas stable.

Avantages :

- Ils se satisfont de moins de mémoire et de puissance de calcul que les filtres FIR
- Beaucoup moins de latence (temps de calcul) que les filtres FIR.
- Ils ont des équivalents dans le monde analogique, ce qui simplifie leur conception.

Inconvénients :

- Réponse en phase non linéaire
- Peuvent ne pas être stable en raison des boucles de rétroaction.

6 TRANSFORMÉE DE FOURIER, DFT, FFT

Commençons par un peu de vocabulaire.

- Transformée de Fourier. Il s'agit d'une opération mathématique permettant de calculer le contenu spectral (fréquentiel) d'un signal temporel non périodique.
- DFT *Discrete Fourier Transform* - Transformée de Fourier Discrète, c'est-à-dire utilisable sur un signal échantillonné (par exemple dans un DSP). Il s'agit donc de la version de la transformée de Fourier applicable à des signaux échantillonnés.
- FFT *Fast Fourier Transform* - Transformée Rapide de Fourier. C'est un (en fait il y en a plusieurs) algorithme qui permet de calculer la transformée de Fourier sur un DSP plus rapidement que la DFT de base, surtout pour traiter un grand nombre d'échantillons.

La FFT est donc un algorithme très rapide qui permet de décomposer un signal discret en ses composants spectraux, c'est-à-dire passer du domaine temporel au domaine fréquentiel. Bien entendu il existe une opération inverse (IFFT) pour revenir dans le domaine temporel.

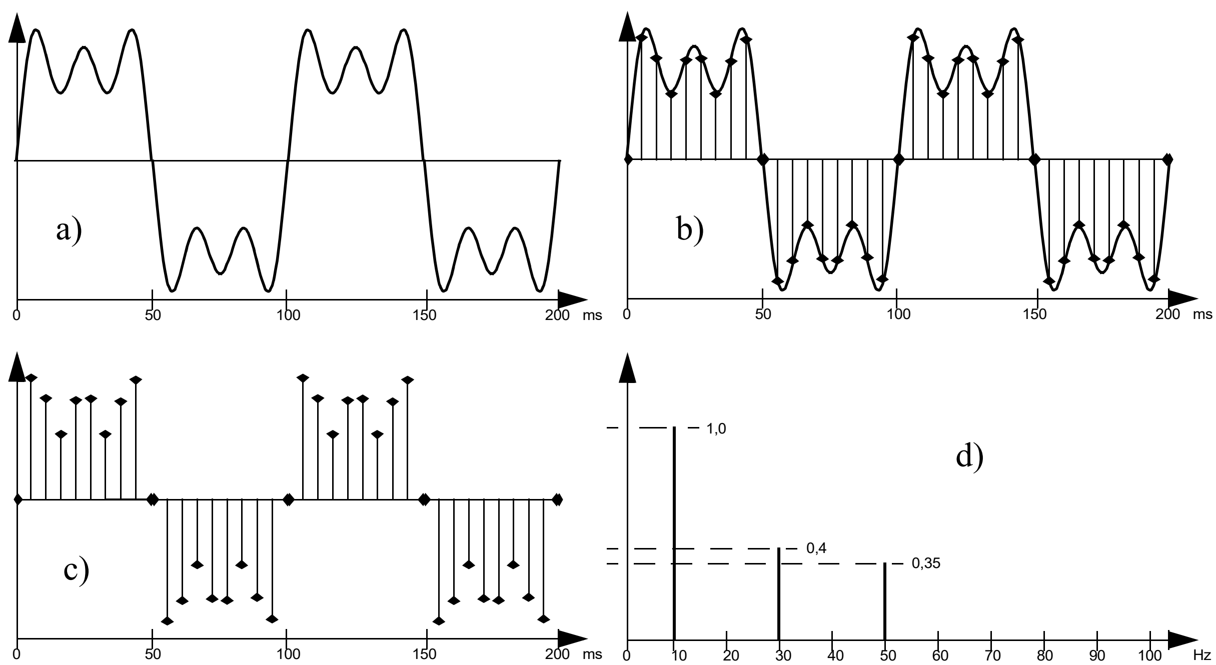

Figure 10 En a) un signal dont un observateur averti verra qu'il contient une fondamentale et des harmoniques 3 et 5. Ce signal est numérisé selon b) pour obtenir la suite d'échantillons représentés en c). En d) le résultat d'une FFT met ceci en évidence ainsi que les amplitudes de ces harmoniques.

En a) un signal qui contient une fondamentale et ses harmoniques 3 et 5. C'est un signal analogique. En b) le même signal tel qu'il va être échantillonné (ADC). En c) les signaux discrets représentant le même signal. Finalement en d), Le résultat d'une FFT sur ce signal discret.

Une IFFT sur le signal en d) redonnerait le signal en c).

6.1 Convolution par FFT (*FFT convolution*)

À ce stade, on peut se poser la question de la possibilité d'effectuer le filtrage dans le domaine fréquentiel. En effet puisqu'une opération de filtrage consiste à supprimer certaines fréquences, ne serait-il pas plus simple d'effectuer ceci après une FFT, puis de revenir dans le domaine temporel ? La réponse est oui, mais. En effet, si cette opération est parfaitement réalisable, les 2 conversions FFT, puis IFFT prennent du temps.

Une autre opération qui prend du temps est un filtrage FIR, en particulier avec un nombre élevé de coefficients (*taps*).

En pratique le recours à la convolution par FFT s'avère bénéficiaire pour un FIR de plus de 40 à 80 *taps* selon le DSP.

Cette opération se fait par une FFT unique de la réponse impulsionnelle du filtre et une FFT sur des paquets du flux de données. Une convolution dans le domaine temporel correspond à une multiplication dans le domaine fréquentiel, Puis ceci effectué, une IFFT retourne le signal filtré dans le domaine temporel.

6.2 Transformée de Hilbert

La transformée (ou transformation) de Hilbert est une opération mathématique qui introduit un déphasage de 90° sur tous les composants spectraux d'un signal.

L'implémentation dans un DSP d'une transformée de Hilbert prend la forme d'un filtre, typiquement de configuration FIR. C'est un filtre passe-tout, c'est-à-dire qu'il a une réponse plate en fréquence. Il n'a d'effet que sur la phase des signaux.

Un rapport de phase de 90° est d'une grande utilité dans les modulateurs et les démodulateurs comme nous le verrons ci-dessous.

6.3 Filtres et plus

Pour en finir avec les filtres mentionnons les fonctions que l'on trouve le plus souvent sur les SDR et implémentées au moyen d'un DSP. Typiquement ces filtres sont placés dans la moyenne fréquence (*low IF*) ou à défaut en bande de base et bien entendu offrent de meilleures performances que leurs équivalents analogiques :

- Filtres passe-bas et passe-haut, généralement avec une fréquence de coupure réglable (modification des coefficients). Ces 2 filtres utilisés ensembles forment un filtre passe-bande réglable en largeur de bande et en fréquence centrale.
- Filtre(s) *notch* pour éliminer un ou plusieurs sifflement (d'hétérodynage par exemple). Typiquement avec un DSP ces filtres cherchent et suivent automatiquement le ou les signaux à éliminer.
- Filtre de bruit (*noise filter*), réglables pour éliminer le bruit sans (trop) affecter le son. Deux sortes de bruits peuvent être éliminés, soit le bruit à large bande, soit les bruits répétitifs (algorithme adaptatif).
- *Noise blanker* pour éliminer les bruits impulsionnels.

- Compresseur de modulation. Cette fonction quasiment indispensable dans un *transceiver* SSB est parfaitement adaptée au traitement pas DSP.

Les autres fonctions assignées à un DSP sont la modulation et la démodulation, dont nous allons parler maintenant.

7 MODULATION DÉMODULATION :

Il n'est malheureusement pas possible d'aborder ce sujet méthodiquement sans une certaine quantité de mathématiques, pas nécessairement compliquées, mais cependant d'un niveau que tout le monde ne possède pas. Ici nous nous contenterons de présenter les solutions les plus utilisées dans les SDR pour la modulation et la démodulation, suivies d'un minimum de maths qui peuvent être sautées par ceux à qui ça ne parle pas.

7.1 Quadrature I & Q

Quasiment toutes les opérations de modulation/démodulation dans un SDR utilisent des signaux en quadrature. Les 2 signaux en quadrature sont appelés *I* pour (*In phase)*, c'est le signal issu de l'ADC, et une copie de ce signal déphasée de 90°, c'est le signal en quadrature *Q*. De même, l'oscillateur local dans le schéma ci-dessous a des sorties en quadrature (cos et sin).

7.2 Modulateur SSB

La figure ci-dessous représente l'implémentation typique d'un modulateur SSB.

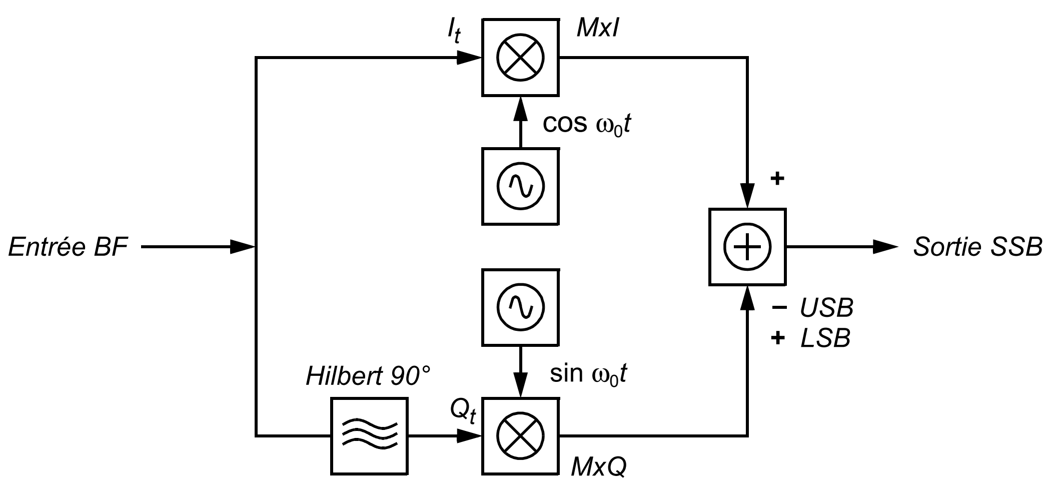

Figure 11 Modulateur SSB. Le signal BF est transformé en un signal complexe *I* et *Q* avant mixage avec la porteuse. Puis la bande latérale désirée est sélectionnée mathématiquement, ce qui élimine la bande latérale indésirable en raison des rapports de phase.

Le signal d'entrée est séparé en deux chemins. Le premier I_t est appliqué au *mixer MxI* alors que le second est d'abord déphasé de 90° avant application au *mixer MxQ*.

Les deux *mixers* sont alimentés par la porteuse, ce qui provoque un changement de fréquence du signal de modulation vers la fréquence de sortie.

À noter que les 2 *mixers* sont alimentés en quadrature à la fréquence $\omega_0 t$ ($\omega = 2\pi f$).

Chacun de ces *mixers* fournit un signal en DSBSC[1].

Finalement, selon que l'on additionne ou soustrait ces 2 signaux, on obtient en sortie un signal en LSB ou en USB, l'autre bande latérale s'annulant en raison des rapport de phase de ces deux signaux.

Version maths :

Soit le signal BF d'entrée $\cos(\omega_s t)$ et la porteuse $\cos(\omega_0 t)$ et $\sin(\omega_0 t)$

I_t vaut donc $\cos(\omega_s t)$ alors que Q_t vaut $\sin(\omega_s t)$

Puis I_t et Q_t sont mixés avec la porteuse $\cos(\omega_0 t)$ et $\sin(\omega_0 t)$

À la sortie du mixer *MxI*, on trouve la somme et la différence des signaux sur ses entrées :

$$\frac{1}{2}[\cos(\omega_0 t - \omega_s t) + \cos(\omega_0 t + \omega_s t)]$$

et à la sortie de *MxQ* :

$$\frac{1}{2}[\cos(\omega_0 t - \omega_s t) - \cos(\omega_0 t + \omega_s t)]$$

où $\cos(\omega_0 t - \omega_s t)$ représente la bande latérale inférieure et $\cos(\omega_0 t + \omega_s t)$ la bande latérale supérieure.

Ajouter ces deux expressions ne laisse que la LSB en sortie :

LSB : $\cos(\omega_0 t - \omega_s t)$

et les soustraire ne laisse que l'USB :

USB : $\cos(\omega_0 t + \omega_s t)$

7.3 Démodulateur SSB

Un démodulateur SSB ressemble beaucoup à un modulateur SSB, mais à l'envers.

Le signal à démoduler est appliqué sur 2 mixers alimentés par un oscillateur local en quadrature à la fréquence $\omega_0 t$ ($\omega = 2\pi f$). En sortie des mixers on trouve la somme et la différence de leurs signaux d'entrées. Les filtres passe-bas (LPF) ne conservent que la différence (contenu de basse fréquence). Le signal *Qt* est obtenu par un déphasage du signal issu de *Mx2* (par exemple transformée de Hilbert). Finalement, selon que l'on

1. *Double Side Band Suppressed Carrier* - Double bande latérale porteuse supprimée.

additionne *It* et *Qt* ou que l'on les soustrait obtient en sortie un signal en LSB démodulé ou en USB démodulé.

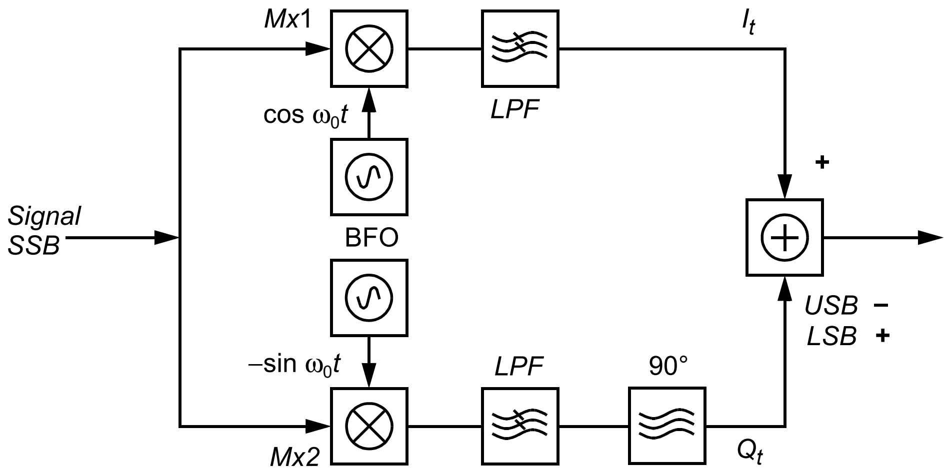

Figure 12 Le signal d'entrée modulé soit en LSB, soit en USB est appliqué sur 2 mixers alimentés en quadrature. À la sortie des *mixers*, après filtrage passe-bas pour éliminer les composantes de haute fréquence, l'un des signaux de sortie et encore déphasé de 90°, puis selon que l'on additionne ces 2 signaux ou que l'on les soustrait, on obtient la démodulation d'un signal USB ou LSB.

Version maths :

Soit le signal HF (ou MF) d'entrée modulé soit en LSB $\cos(\omega_0 t - \omega_s t)$ soit en USB $\cos(\omega_0 t + \omega_s t)$ avec la porteuse réinsérée (BFO) en quadrature $\cos \omega_0 t$ et $-\sin \omega_0 t$

On trouve en sortie de *Mx1* et de *Mx2* la somme et la différence des signaux d'entrée :

Mx1, cas de l'USB :

$$\frac{1}{2}[\cos(\omega_0 t + \omega_s t + \omega_0 t) + \cos(\omega_0 t + \omega_s t - \omega_0 t)] = \frac{1}{2}\cos(\omega_s t) \text{ après filtrage}$$

Mx2, cas de l'USB :

$$-\frac{1}{2}[\sin(\omega_0 t + \omega_s t + \omega_0 t) + \sin(\omega_0 t + \omega_s t - \omega_0 t)] = -\frac{1}{2}\sin(\omega_s t) \text{ après filtrage}$$

Mx1, cas de la LSB :

$$\frac{1}{2}[\cos(\omega_0 t - \omega_s t + \omega_0 t) + \cos(\omega_0 t - \omega_s t - \omega_0 t)] = \frac{1}{2}\cos(\omega_s t) \text{ après filtrage}$$

Mx2, cas de la LSB :

$$-\frac{1}{2}[\sin(\omega_0 t - \omega_s t + \omega_0 t) + \sin(\omega_0 t - \omega_s t - \omega_0 t)] = -\frac{1}{2}\sin(\omega_s t) \text{ après filtrage}$$

Les termes $\omega_0 t$ dans les membres de droite s'annulent dans chaque équation, et de plus, en raison du filtrage, tous les termes contenant $\omega_0 t$ (membres de gauche) disparaissent.

Cas de l'USB : I_t est (1/2)cos($\omega_s t$) et après le déphasage de 90° Q_t est −(1/2)cos($\omega_s t$)

si l'on additionne ces 2 signaux en inversant le signe de Q_t, on obtient cos($\omega_s t$) ce qui est bien le signal appliqué précédemment à notre modulateur SSB.

Cas de la LSB : I_t est (1/2)cos($\omega_s t$) et après le déphasage de 90° Q_t est (1/2)cos($\omega_s t$)

si l'on additionne ces 2 signaux, on obtient cos($\omega_s t$) ce qui est bien le signal appliqué précédemment à notre modulateur SSB.

7.4 Démodulation AM

Ici on veut simplement appréhender l'enveloppe du signal. Il existe plusieurs méthodes utilisables pour ceci en technique DSP. Une technique particulièrement élégante met en jeu l'identité trigonométrique : $\sin(\omega t)^2 + \cos(\omega t)^2 = 1$

Considérons le schéma ci-dessous.

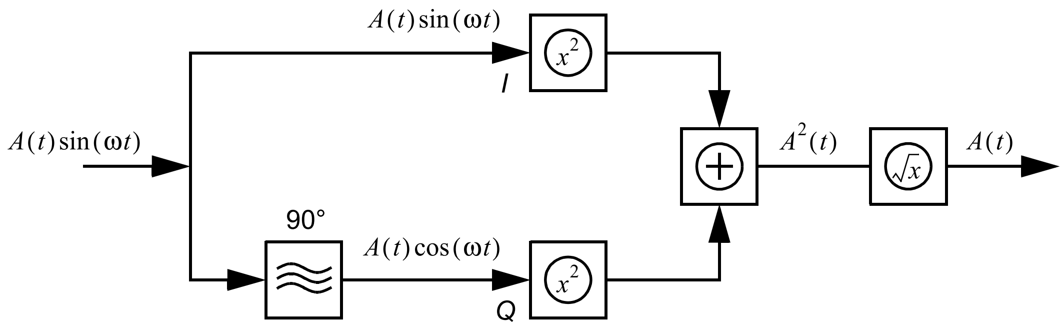

Figure 13 Les signaux I et Q sont élevés au carré puis additionnés. Selon l'identité trigonométrique $\sin(\omega t)^2 + \cos(\omega t)^2 = 1$, il ne reste plus que le carré du signal de modulation que l'on peut récupérer en prenant la racine de ce résultat.

Le signal à démoduler est composé de $A(t)$, l'enveloppe du signal et $\sin(\omega t)$, la porteuse, c'est notre signal I. Le déphasage de 90° (par exemple transformateur de Hilbert) fournit le signal en quadrature Q. Ces 2 signaux sont élevés au carré puis additionnés. Toute la partie HF est éliminée ($\sin(\omega t)^2 + \cos(\omega t)^2 = 1$) et il ne reste que $A^2(t)$. Finalement après la racine, il ne reste plus que $A(t)$, qui est bien notre signal de modulation.

7.5 Autres démodulateurs

Il est bien entendu possible de réaliser des démodulateurs pour toute autre forme de modulation, FM, PM, FSK, PSK, QAM, etc. Cependant le niveau de complexité (surtout en ce qui concerne les maths) les place hors du niveau de cette brève introduction. Tant que l'OFCOM n'en fait pas des sujets d'examen, laissons-les de coté.

Appendice 1 : Vocabulaire

ADC	*Analog to Digital converter* - Convertisseur Analogue/Numérique
Aliasing	Repliement
DAC	*Digital to Analog Converter* - Convertisseur Numérique.Analogue
Digital	Numérique
DSBSC	*Double Side Band Suppressed Carrier* - Double bande latérale, porteuse supprimée
DSP	*Digital Signal Processor* - Processeur numérique de signaux
FPGA	*Field Programmable Gate Array* - Circuit logique programmable
LSB	*Least Significant Bit* - bit de poids moindre
MSB	*Most Significant Bit* - bit de poids le plus important
OFCOM	Office Fédéral de la Communication (Suisse)
Sampling	Échantillonnage
SDR	*Software defined Radio* - Radio Logicielle
Signal discret	Signal numérisé
Transceiver	Émetteur-récepteur

Appendice 2 : Références et lectures recommandées (liens).

The Scientist & Engineer's Guide to Digital Signal Processing, 1999

Biquad Cascade IIR Filters Using Direct Form I Structure

Finite Impulse Response (FIR) Filters

Digital Envelope Detection: The Good, the Bad, and the Ugly

Signals, Samples and Stuff: A DSP Tutorial (Part1)

Dithering in Analog-to-digital Conversion

OP 2024-01-02 ; 2024-03-05.