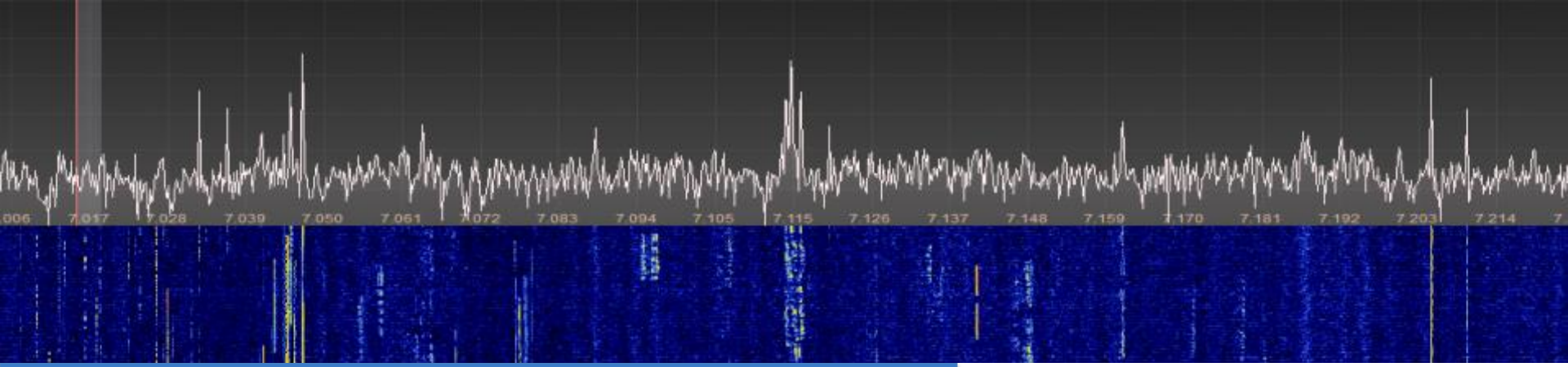


Anatomie d'un logiciel SDR

« Comment ça marche ?»

Sylvain AZARIAN - F4GKR
Congrès du REF – Brives – 2016

www.f4gkr.org

Anatomie d'un logiciel SDR

Objectifs de cette présentation :

- Comprendre le fonctionnement des logiciels SDR et les différentes étapes de calculs,
- Savoir quels matériels et logiciels sont adaptés à nos besoins et ce que cela implique;

- Faire un peu de prospective sur les évolutions que l'on peut prévoir

- Sans faire (trop) de Maths, en restant « amateurs »

(pardon pour les quelques raccourcis utilisés)

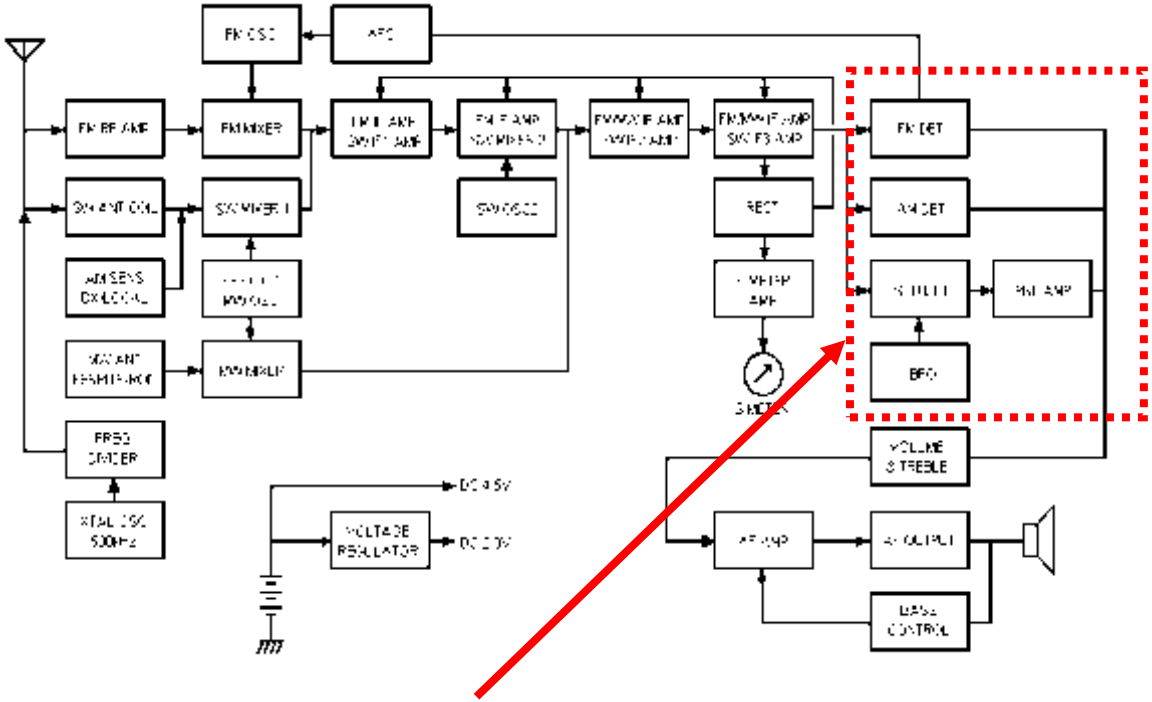
A l'origine du SDR: De plus en plus de fonctions…

Une technologie => Un chip

- GSM
- GPRS
- UMTS
- 4G
- WiFi
- BlueTooth
- WiMAX
- …

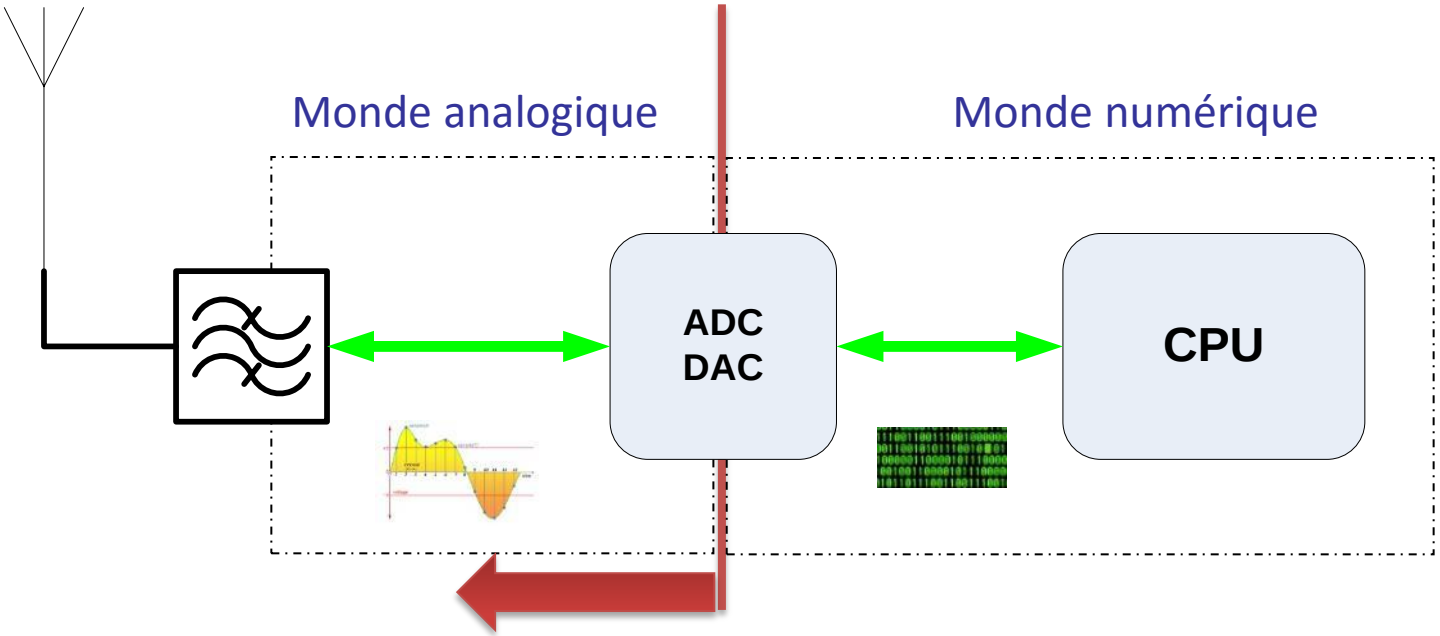
Ça ne "rentre" plus…

Electronique “grand public”



Un sous-ensemble par modulation

SDR : Architecture “idéale”

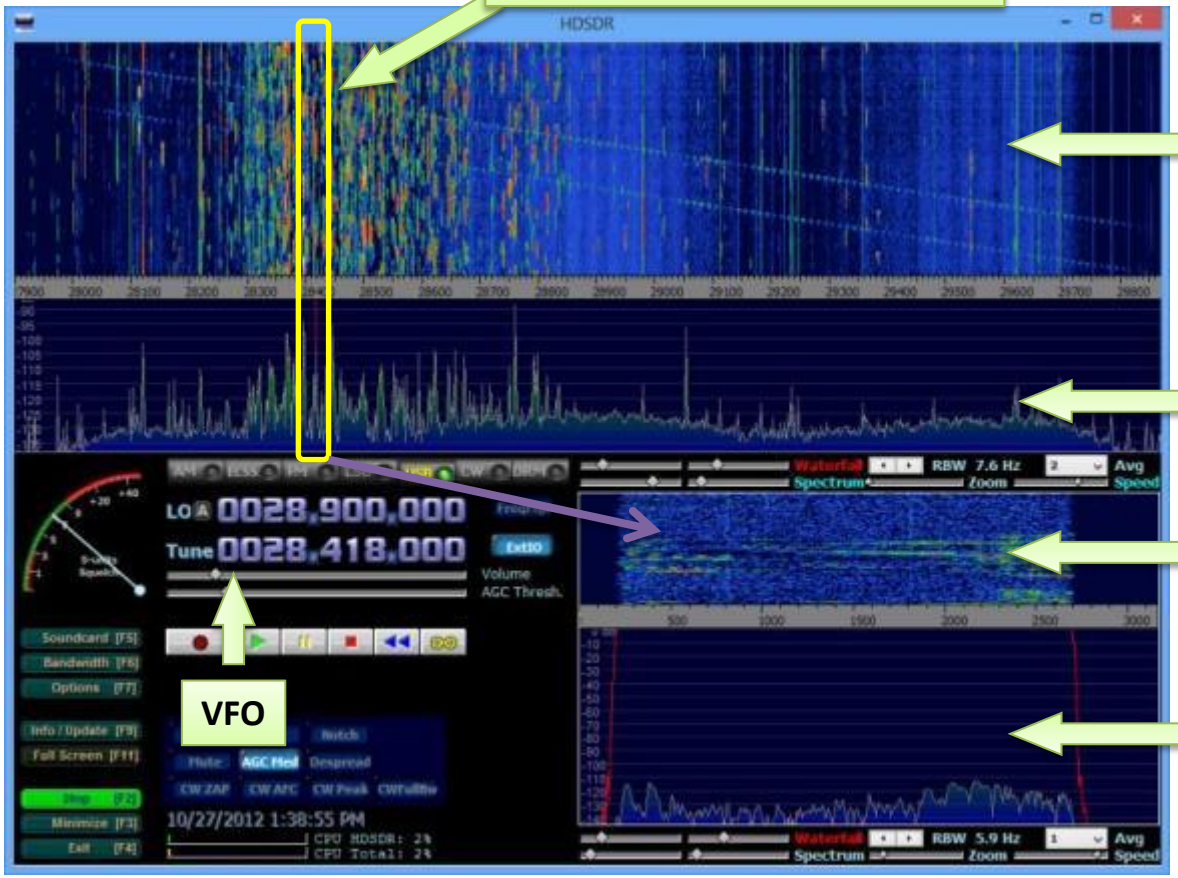


Les éléments qui composent le SDR idéal :

- **Une antenne et un peu de filtrage,**
- **Des circuits de conversion analogique/numérique,**
- **CPU + applications.**

Avec un « monde analogique » aussi réduit que possible

Logiciel SDR : exemple HDSDR



Zone d'intérêt (autour du VFO)

Waterfall entrée RF

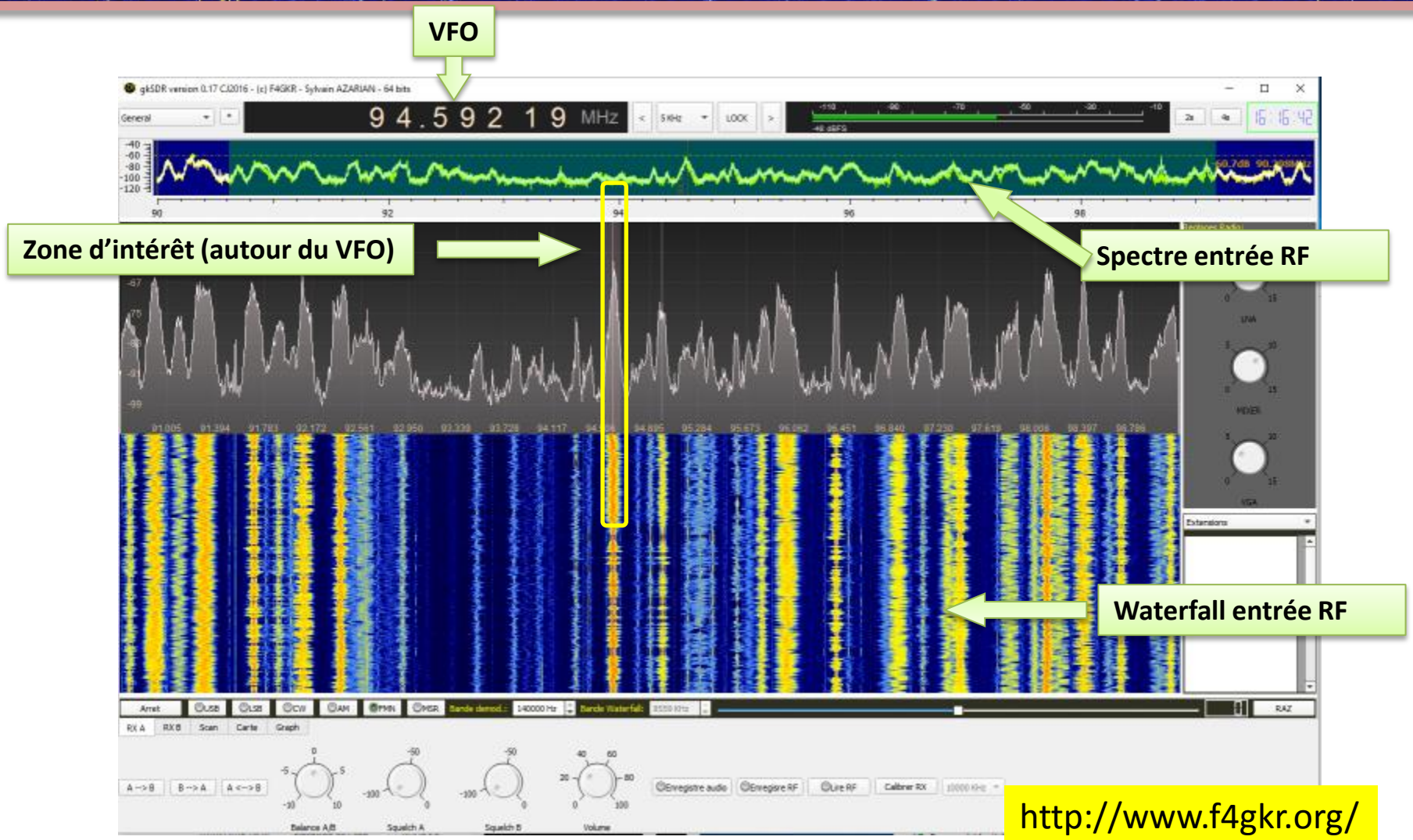
Spectre RF

Waterfall zone démodulée

Spectre zone démodulée

http://www.hdsdr.de/

Logiciel SDR : exemple gkSDR



1ère PARTIE

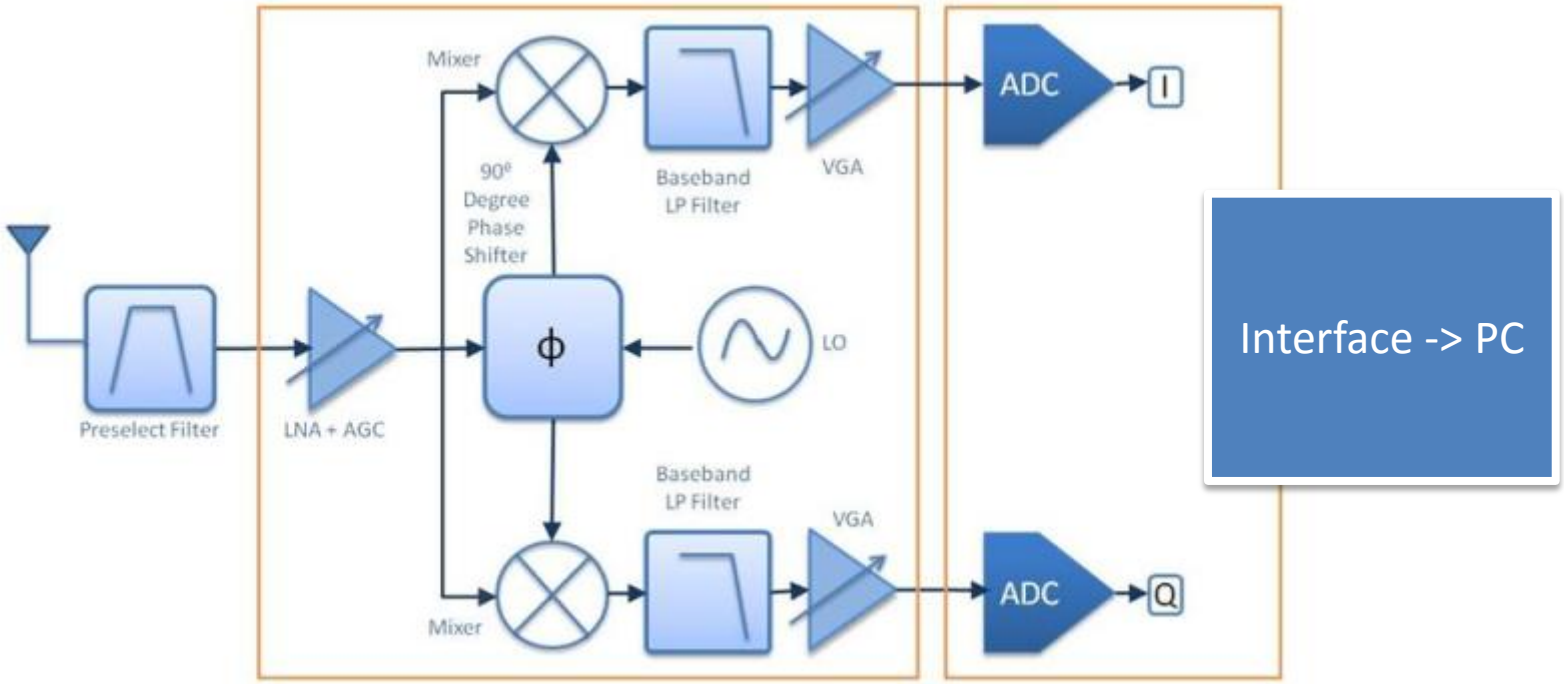
Aspects « hardware » avec un peu de Maths

Récepteur SDR

Principalement deux types d'architectures dans les équipements amateurs :

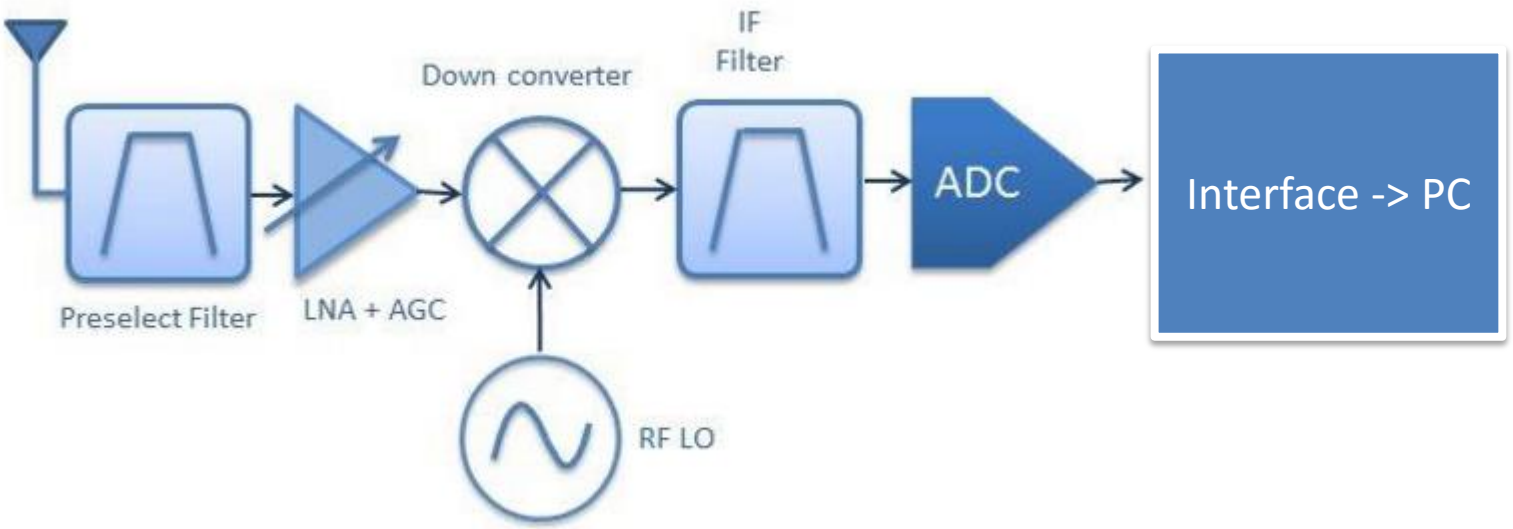
- A échantillonnage RF direct
- A conversion directe

SDR à conversion directe



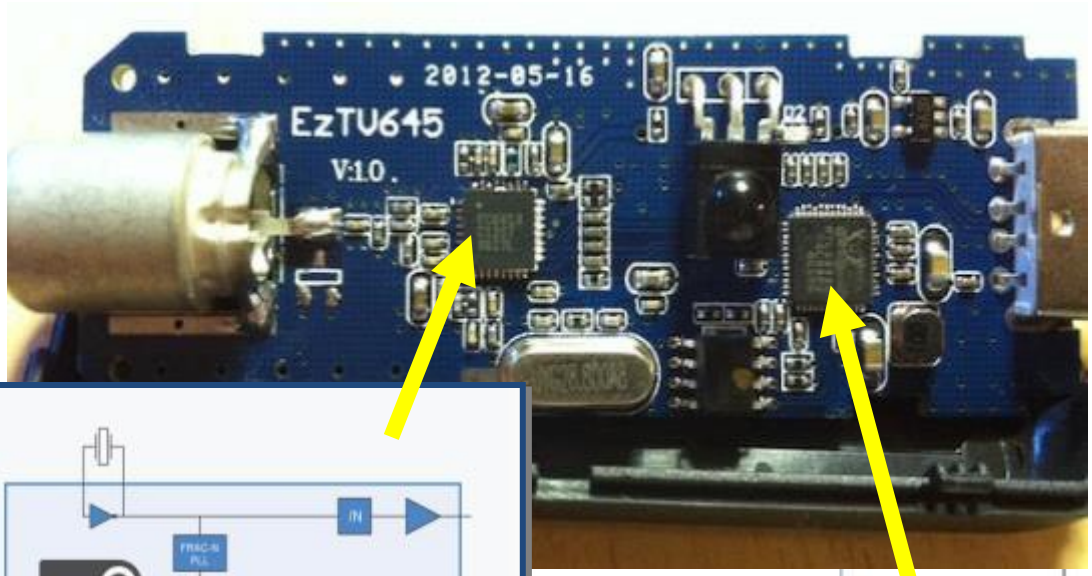
Un oscillateur local sélectionne la fréquence d'intérêt. Autour de cette fréquence, deux bandes sont « extraites » puis numérisées pour être envoyées vers l'ordinateur et le logiciel de traitement.

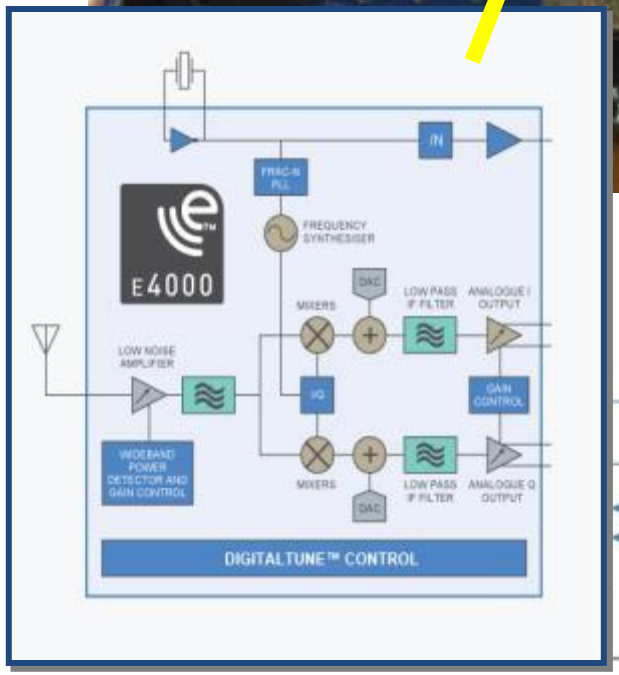
SDR à échantillonnage direct

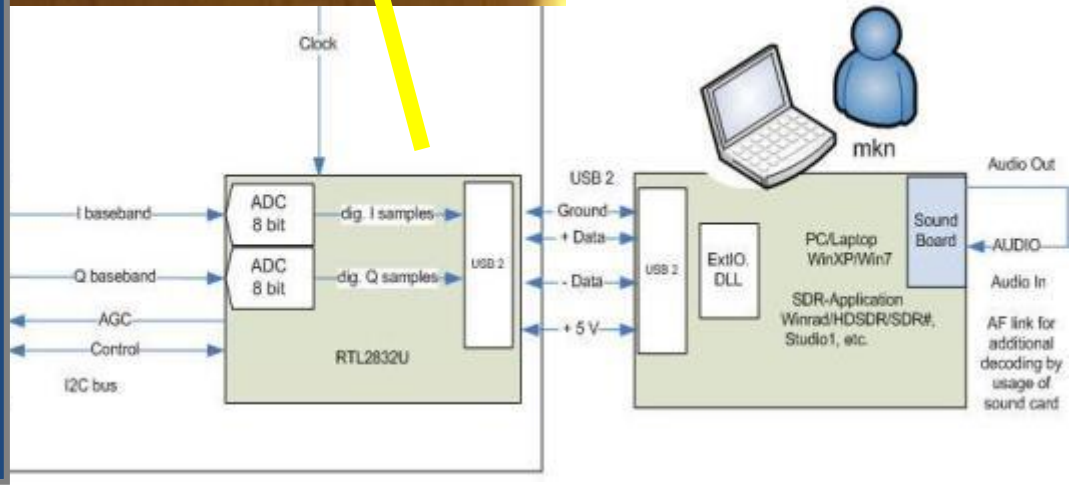


Un oscillateur local transpose le signal reçu. Après filtrage, le signal résultant est numérisé pour être envoyé vers l'ordinateur et le logiciel de traitement.

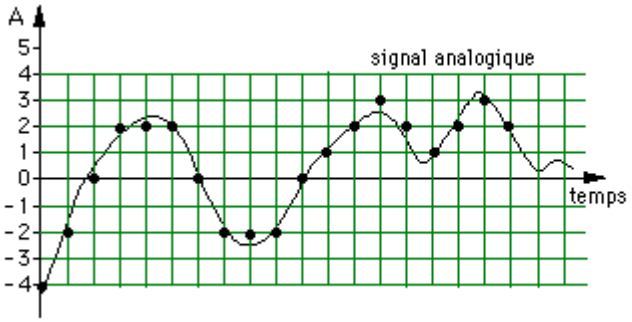
Exemple : clé USB RTLSDR







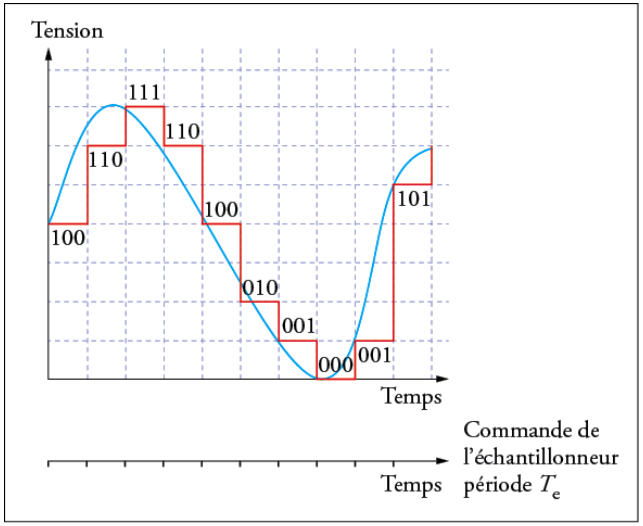
La conversion analogique / numérique

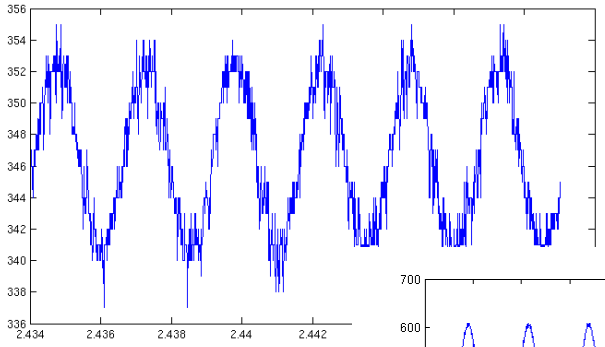


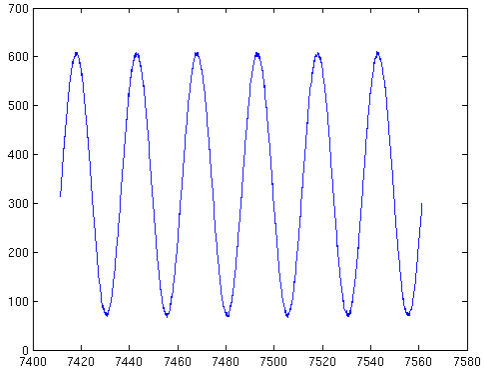
t	val
0	-4
1	-2
2	0
3	2
4	2
5	2
6	0
7	-2
8	-2
9	-2
	...

On va « mesurer » la tension entrante à des instants précis.

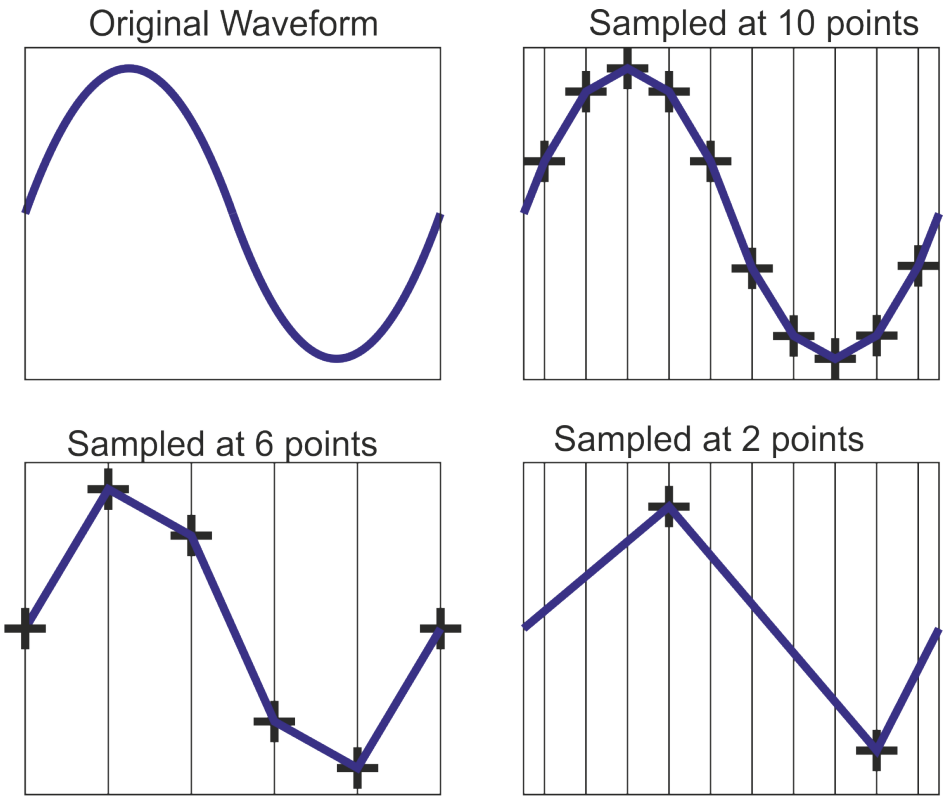
En fonction de la **résolution** du circuit (nombre de bits), on obtiendra une représentation plus ou moins « fidèle »







Nyquist… échantillonner à la bonne vitesse

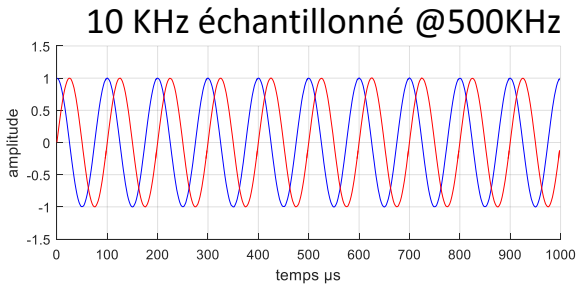


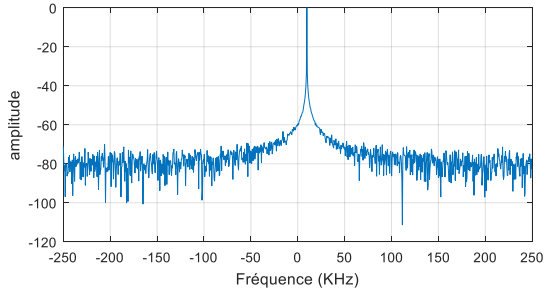
Quel est l'intervalle de mesure optimal ?

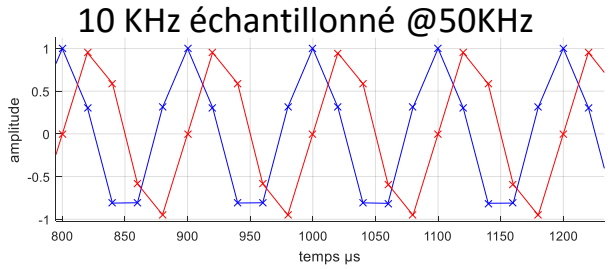
Intuitivement on se rend compte qu'à partir d'un certain nombre d'échantillons par unité de temps (fréquence d'échantillonnage), **les données supplémentaires acquises n'apportent plus d'information nouvelle**.

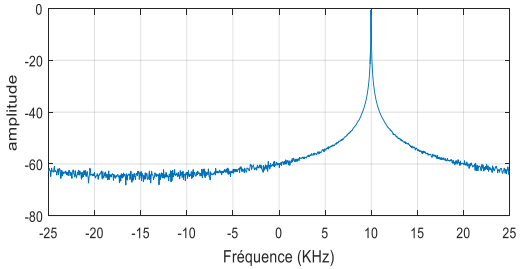
Quelle est la fréquence d'échantillonnage idéale ?

- Critère de Nyquist : **au moins 2.f_{MAX}**
  - 1 MHz de bande = 2 Millions d'échantillons / seconde

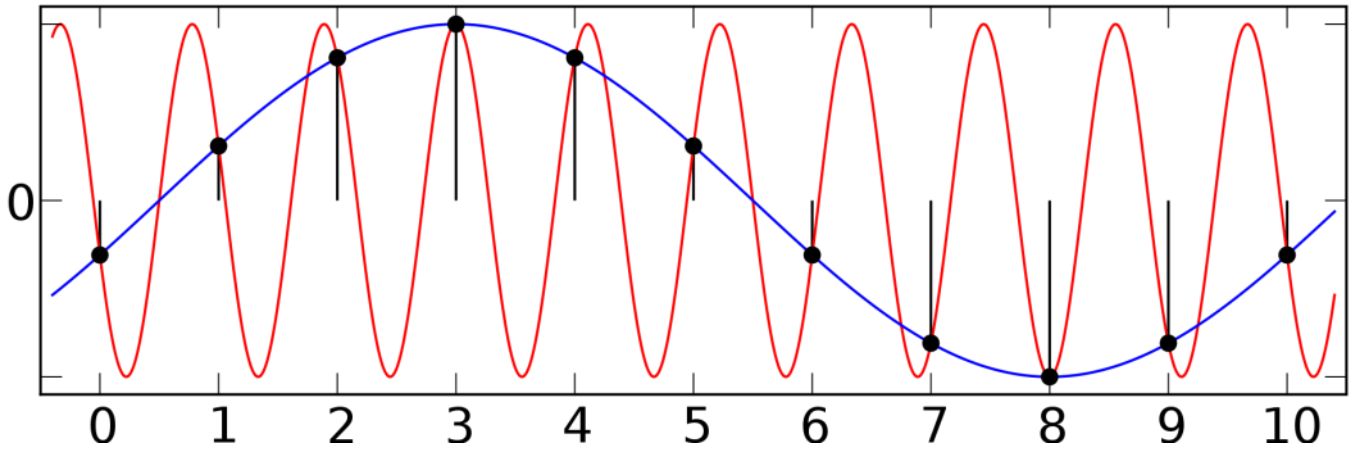






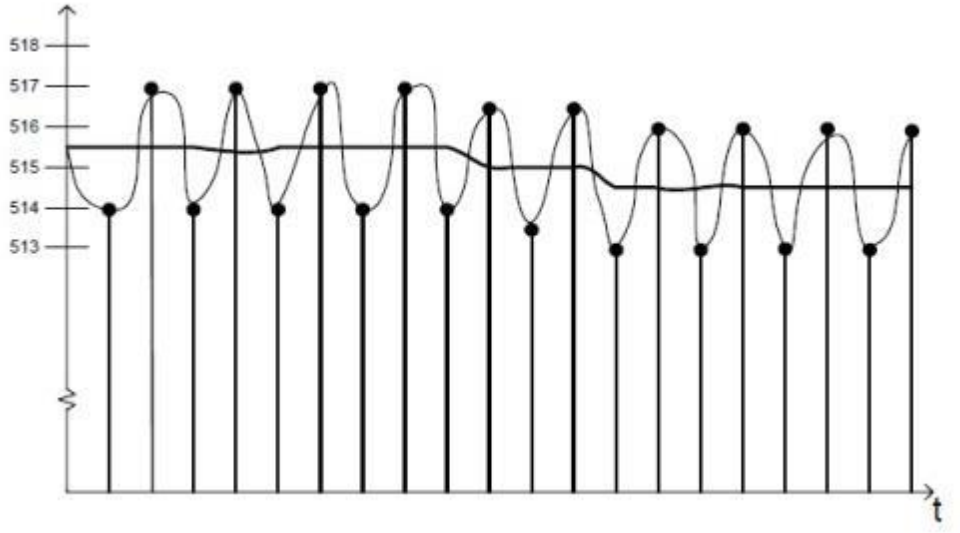


Sous-échantillonnage



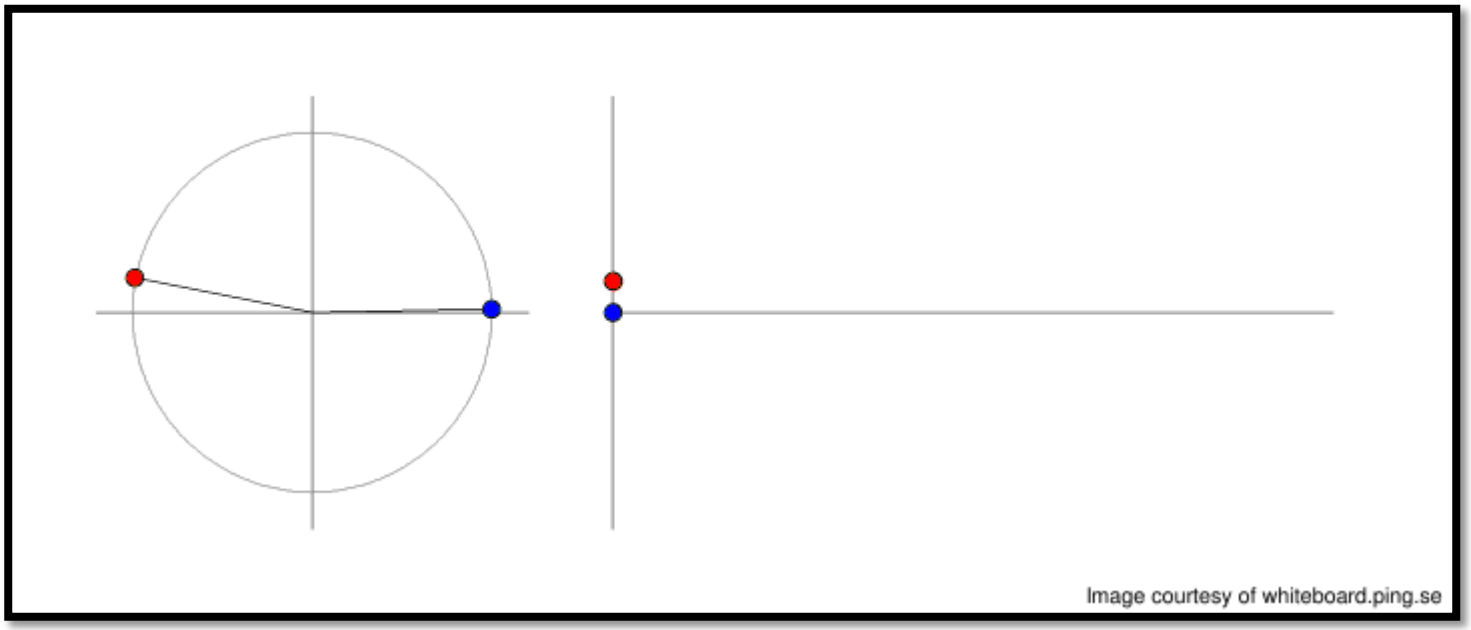
- Sous échantillonner fera apparaître des « alias », c'est-à-dire des signaux qui ne sont pas à leur « vraie » fréquence (phénomène de repliement)

Sur échantillonnage



- Le Sur échantillonnage permet de « gagner des bits supplémentaires » : $(2^n)^2$
  - Gagner 3 bits (n=3) => échantillonner 64 fois plus vite…
  - En pratique on le fait déjà : 2 MHz (clé SDR) ramené à 4 Khz = 2000/4=500… ~$(2^4)^2$ un peu plus de 4 bits « en plus »

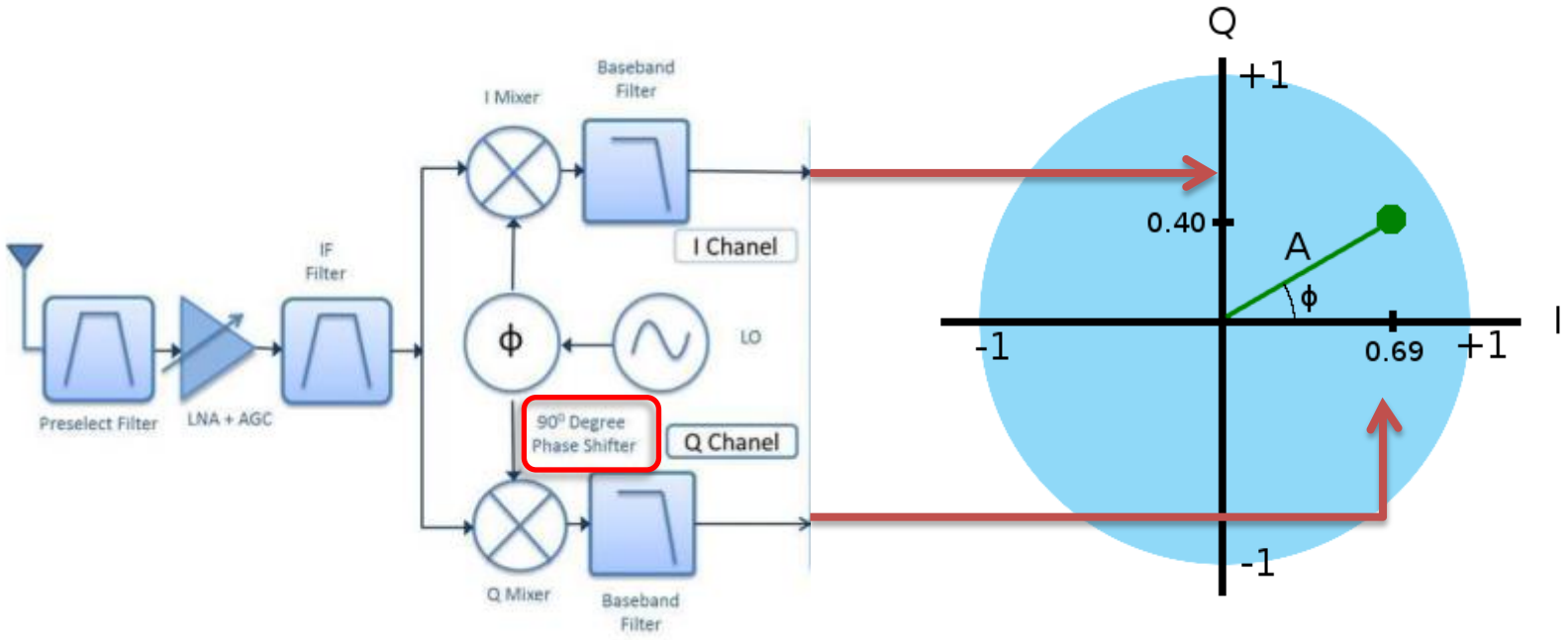
Fréquence négative ?



- Les deux signaux n'évoluent pas « dans le même sens » et pourtant leur représentation temporelle est identique

Signaux en quadrature

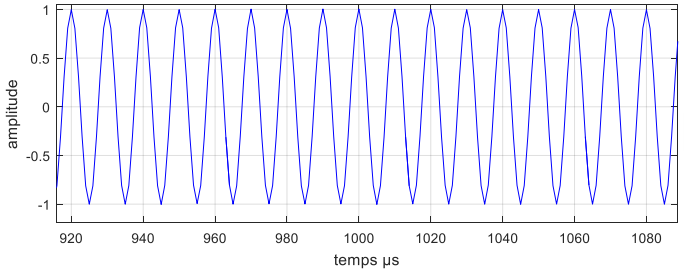
- La solution consiste à prendre deux échantillons du signal mélangé, mais avec deux OL déphasés de 90° : C'est l'échantillonnage « complexe » ou « IQ »

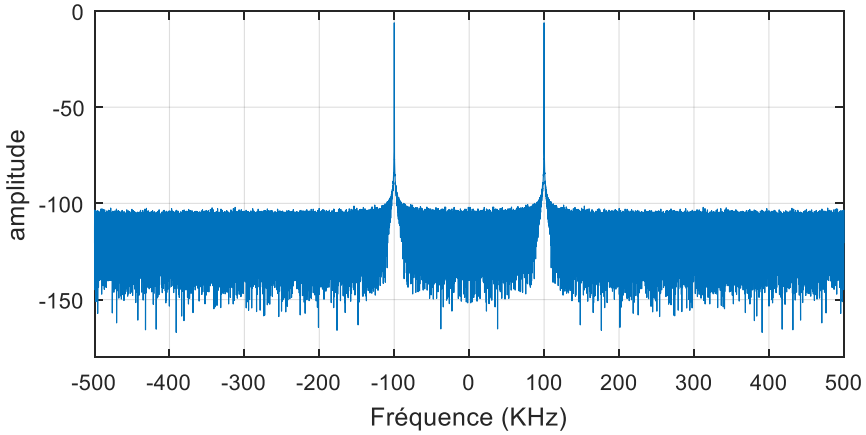


Remarque : pour les SDR à échantillonnage direct il y a une astuce pour retrouver I et Q

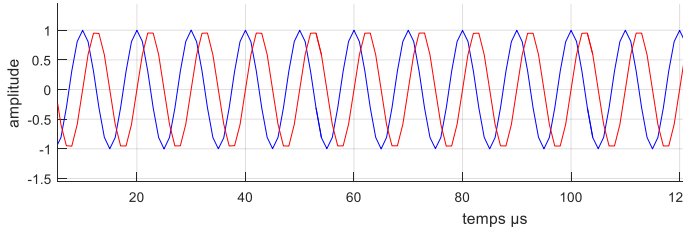
Spectre réel / spectre complexe

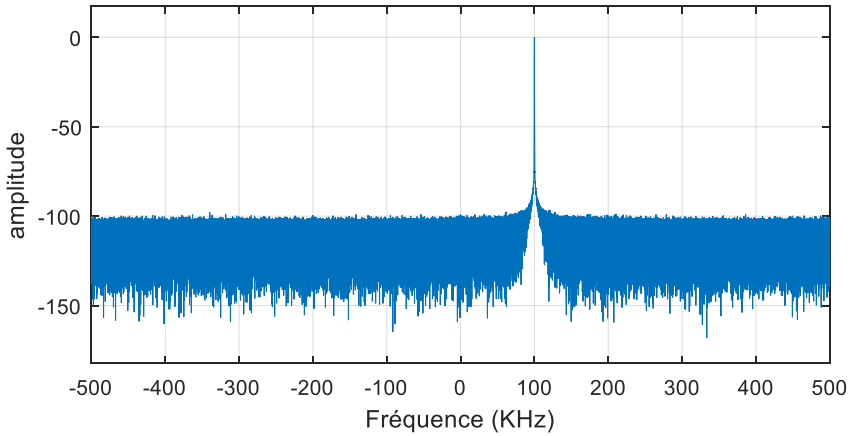
Sans quadrature (pas de I/Q)





Avec quadrature (I/Q)

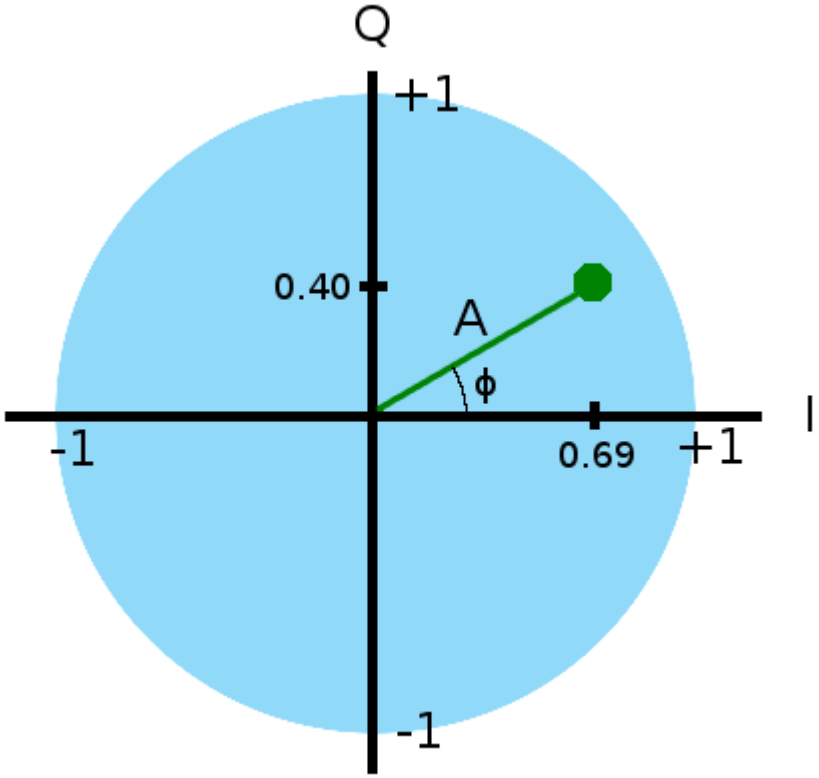




- Les signaux IQ permettent de connaître « le signe » du signal

En fait ça simplifie les calculs…

- Si on fait un peu de trigonométrie :



- Amplitude = longueur de A
- Angle ϕ

$$A = \sqrt{I^2 + Q^2}$$

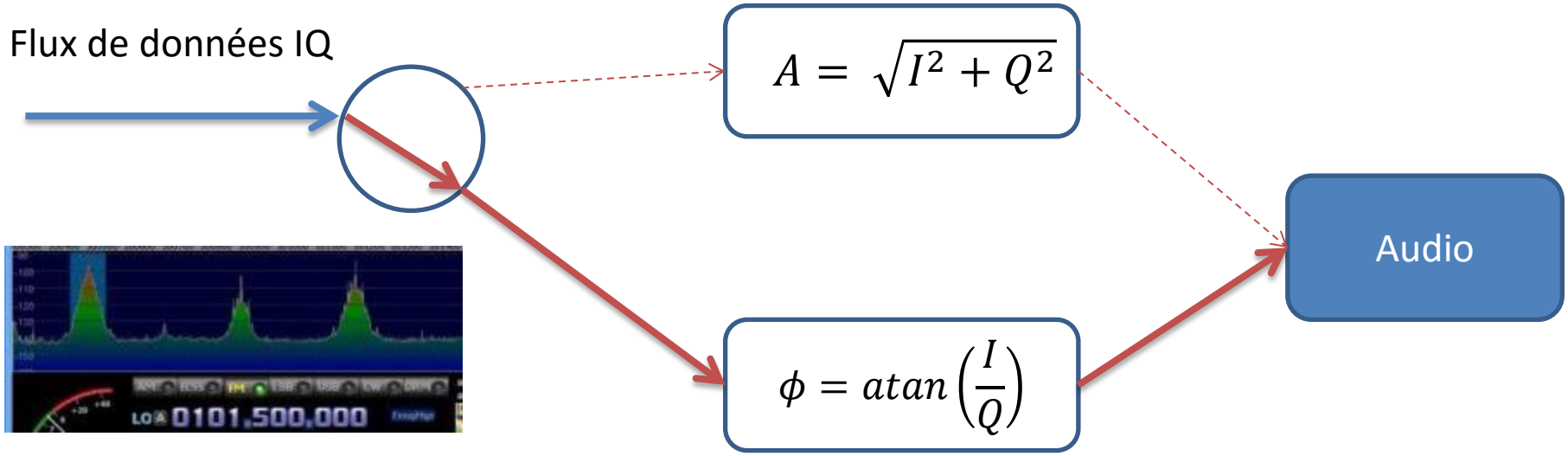
$$\phi = atan\left(\frac{I}{Q}\right)$$

Et donc ? AM … FM …

- **AM** = Amplitude Modulation = Variation de l'amplitude au cours du temps
  - Démoduler l'AM = calculer ***A*** à chaque instant et envoyer à la sortie audio

- **FM** = Modulation de Fréquence (de Phase)
  - Démoduler la FM = calculer ϕ à chaque instant et envoyer *la variation* à la sortie audio

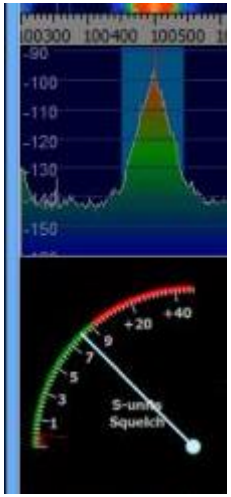
I et Q sont les ingrédients de base



- Chaque modulation correspond à un sous-programme de calcul dédié
- En fonction du choix de l'utilisateur le flux de données IQ est envoyé vers le bloc correspondant

Sert aussi pour…

- Le S-mètre :



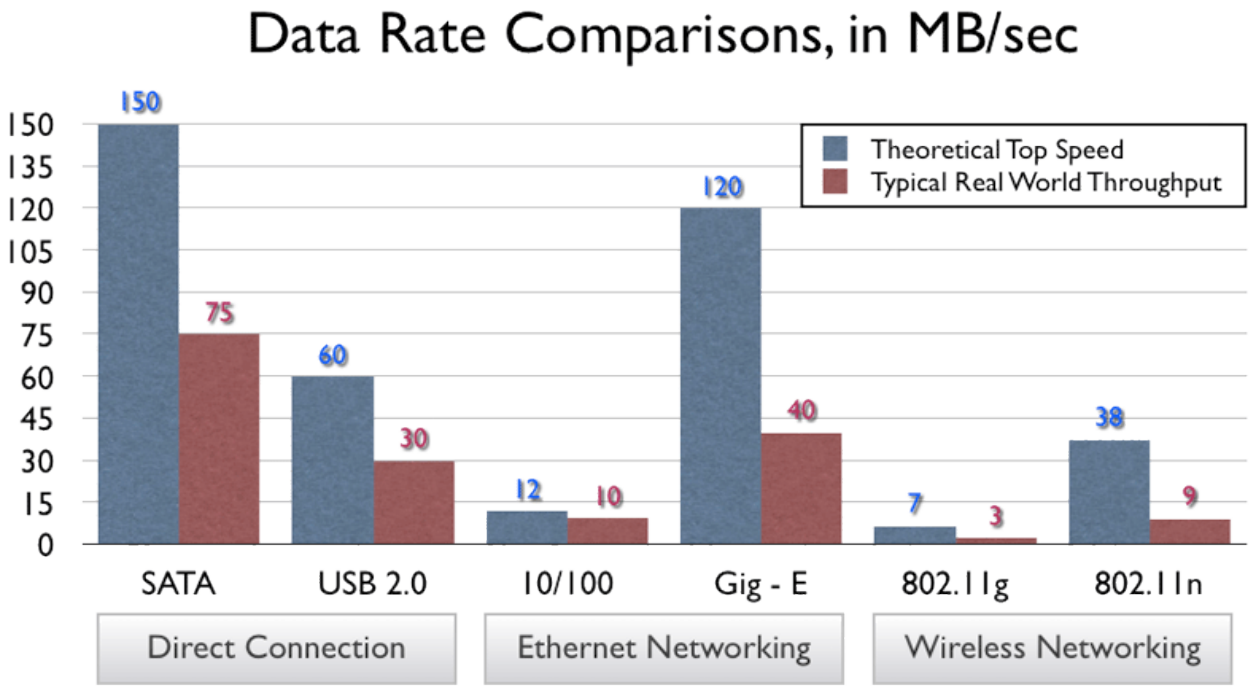
Dans la bande sélectionnée, on fait simplement la somme des amplitudes sur « un certain temps ».

Transférer les échantillons : USB

- L'USB est un **bus série** : 1 octet est transmis bit par bit entre le PC et le périphérique;
- Le débit théorique annoncé est en **bits/seconde**, mais il faut « enlever » toute la partie protocole / signalisation
- Le débit restant pour le SDR est donc très réduit…

- Et on doit souvent transmettre au minimum 2 octets par échantillon (IQ sur 8 bits)

Quels sont les débits réalistes ?



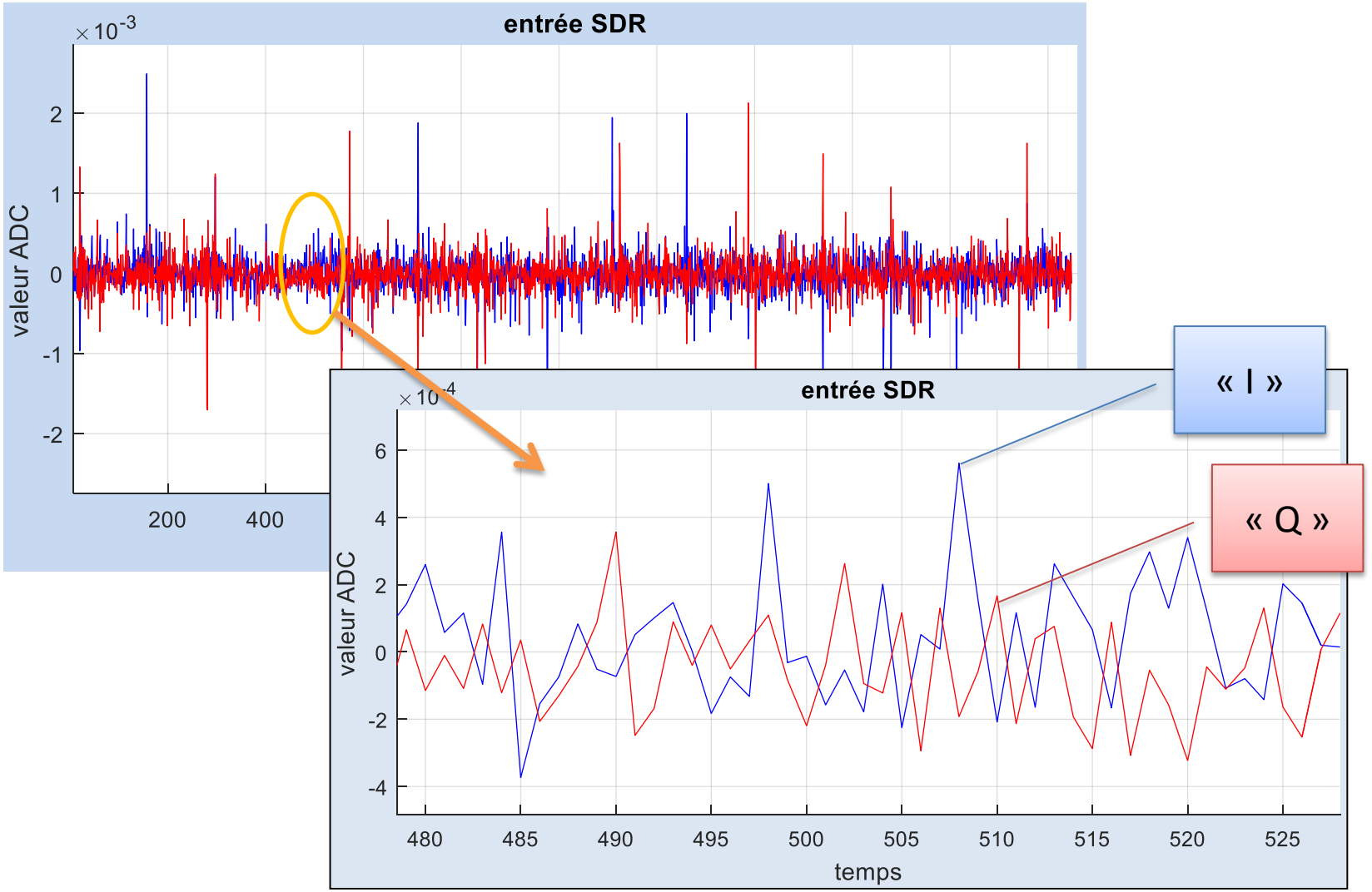
- 30 MB = 15 MHz de bande IQ à 8 bits = 7,5 MHz de bande IQ à 16 bits

USB3.0 et USB3.1

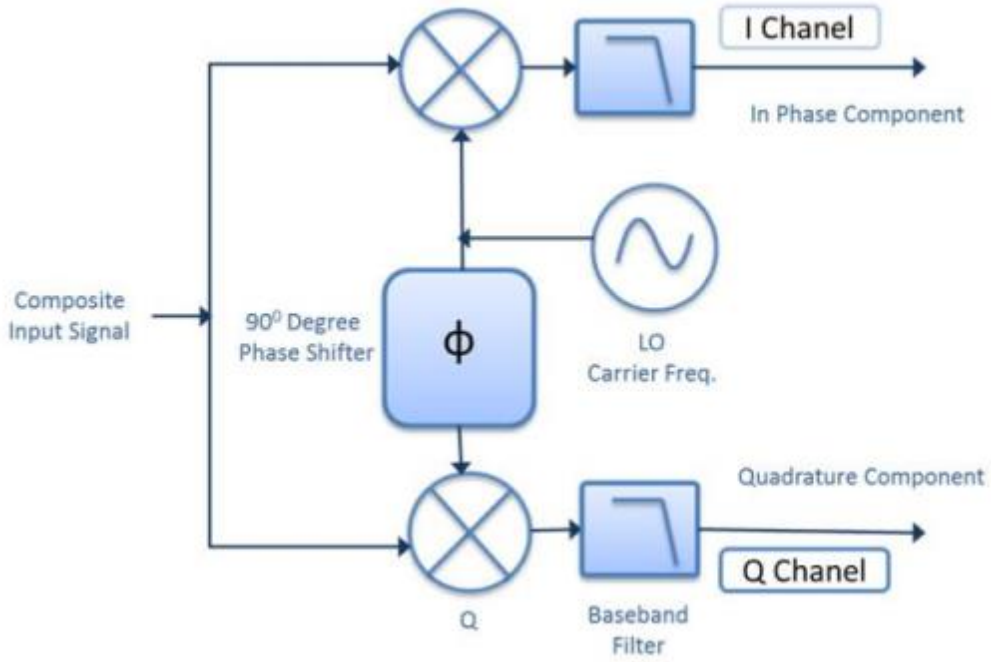
	Version	Speed	Bits/sec	HD movie 25GB
	USB 1.1	Low speed (LS) Full speed (FS)	1.5 Mbps 12 Mbps	~9.25 hours
	USB 2.0	High speed (HS)	480 Mbps	~14 mins
	USB 3.0	SuperSpeed (SS)	5 Gbps	~70 sec
	USB 3.1	SuperSpeedPlus (SSP)	10 Gbps	~35 sec

- 5 Gbps = 5 Giga bits per seconds = environ **300 millions de IQ @ 8 bits** = environ 150 millions de IQ @ 16 bits = 150 MHz avec une « super résolution »
- Problème : avoir le PC qui va assez vite pour les calculs suivants…

Résumé : Ce que l'on a en entrée



Pour la suite de cette présentation



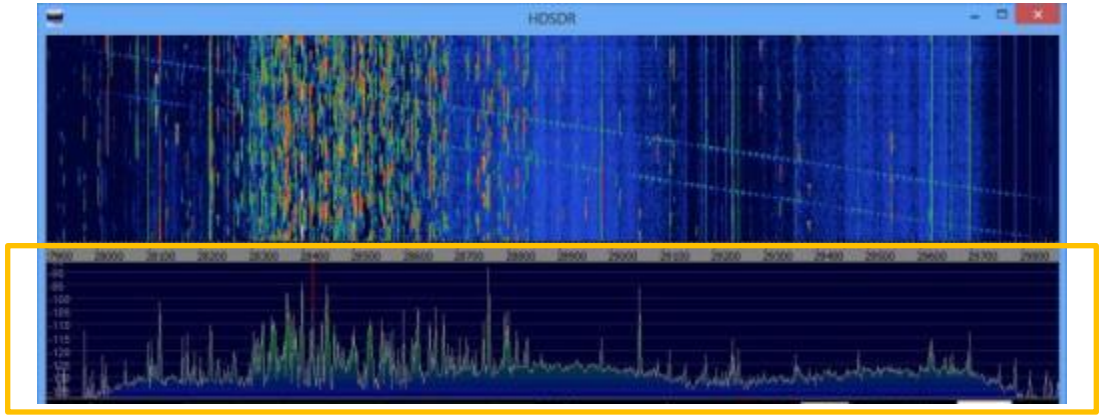
On supposera récepteur de type « conversion directe » envoyant I et Q numérisés séparément

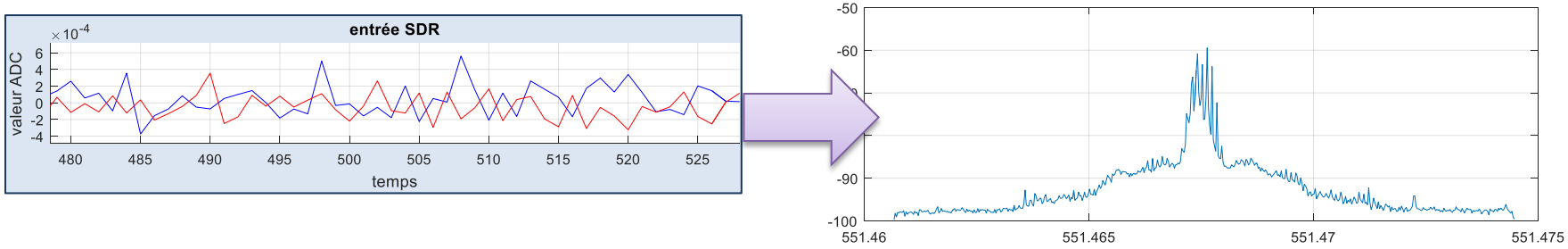
2ème PARTIE

Logiciels SDR : principes de fonctionnement
Algorithmes utilisés

(et surtout pourquoi il faut un PC puissant)

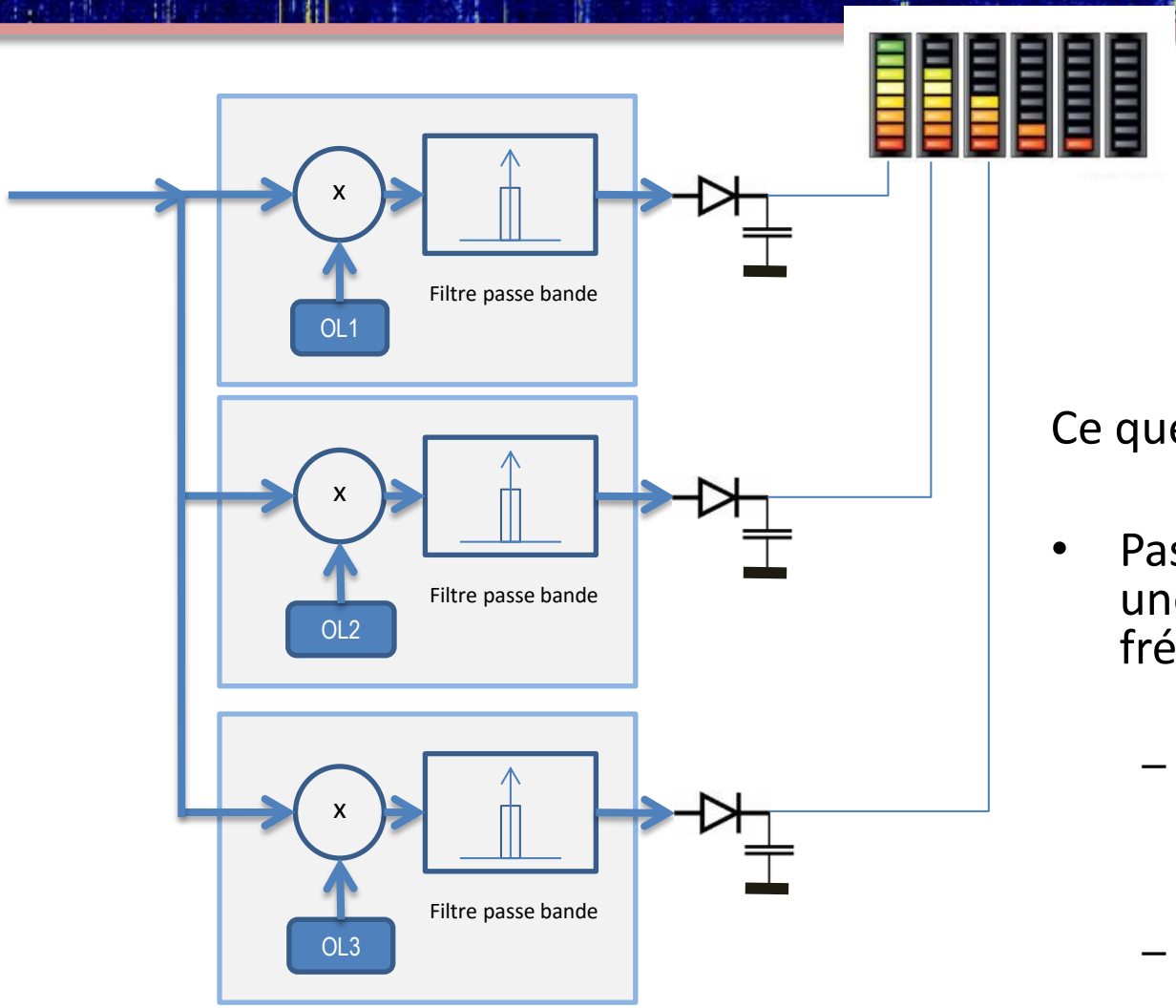
Affichage du spectre





- Transformation « temporel » -> « fréquentiel »
- Comment est-ce réalisé ?

La transformée de Fourier



Ce que permet la FFT :

- Passer d'un signal temporel à une représentation fréquentielle
  - En entrée : une séquence d'échantillons successifs à pris à une cadence constante
  - En sortie : l'amplitude et la phase pour chaque fréquence

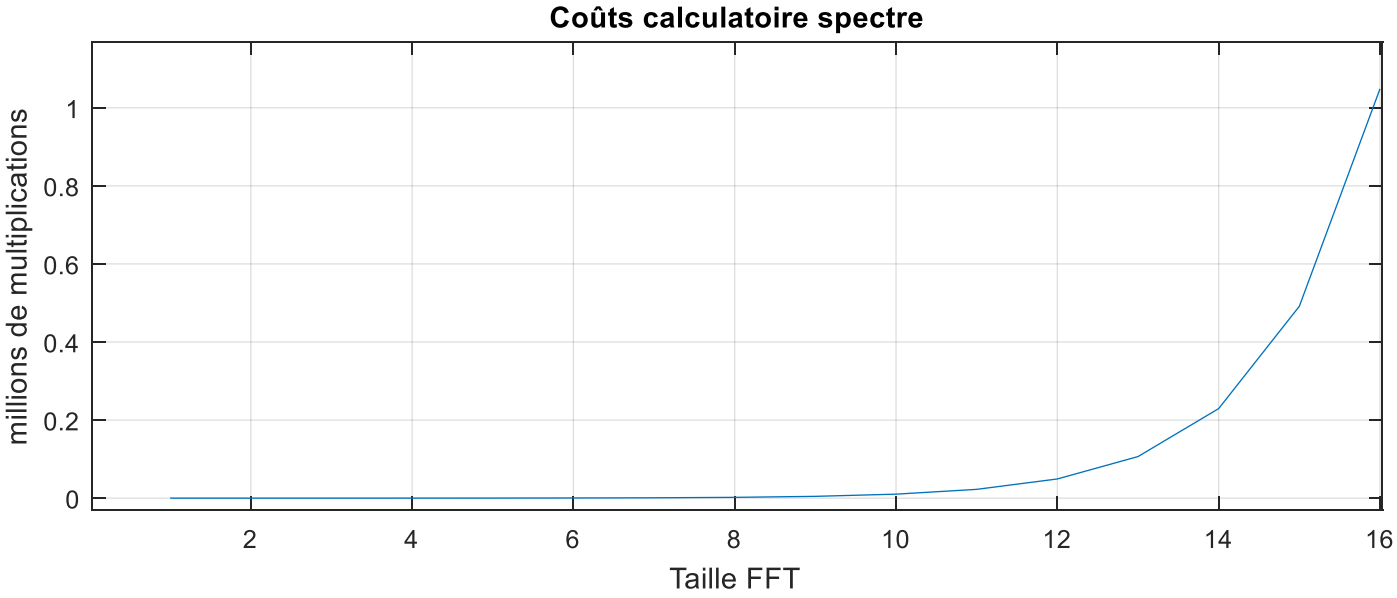
La transformée de Fourier *rapide* (FFT)

- Algorithme de calcul de la transformée optimisé pour des suites de valeurs (des échantillons). Ne requiert « que » $n.\log_2(n)$ multiplications
- Optimisé pour des tailles puissances de 2 (..512,1024,2048 etc.)

- Exemples : SDR à 2 MHz
  - n=1024, $\log_2(n)$=10 il faut environ 10240 multiplications
  - n=4096, $\log_2(n)$=14 il faut environ 57344 multiplications

  - n=1024 donne une résolution de 2M/1024 = 1,95 KHz
  - n=4096 donne une résolution de 2M/4096 = 488 Hz

Quelle limite « raisonnable » ?

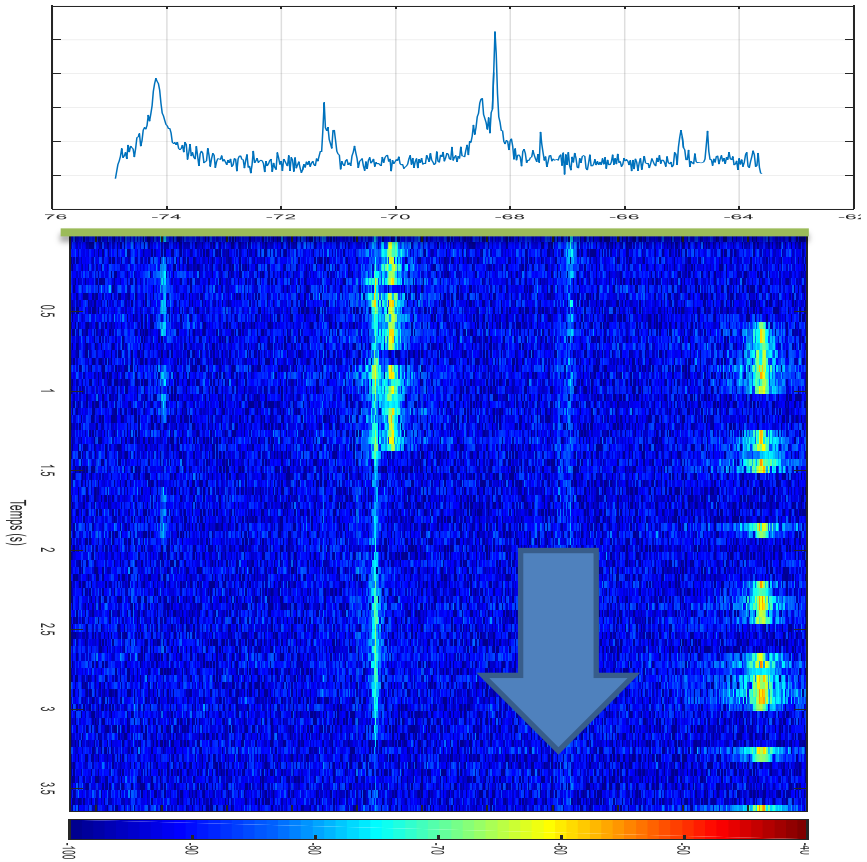


30 Hz de résolution avec 2MHz de bande impose 1 million de multiplications…

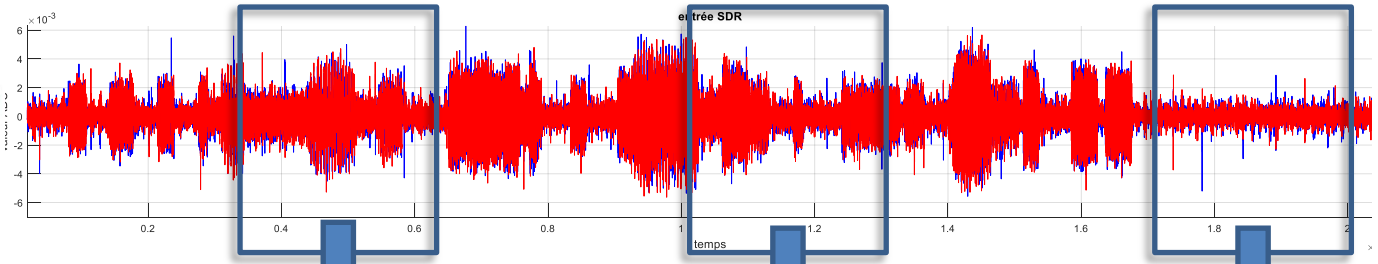
Avoir une résolution de 10 Hz sur 10 MHz de bande… est peu réaliste sans CPU spécial ou algorithme spécial

Le Waterfall (chute d'eau)

- Juste un empilage de spectre…
- On calcule des FFT en continu en prenant des « morceaux » du signal qui arrive du récepteur
- On peut aussi moyenner plusieurs FFT pour faire « sortir » des signaux faibles



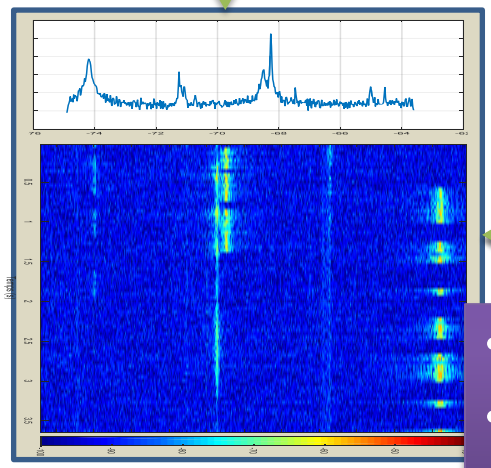
En résumé jusqu'à présent



FFT

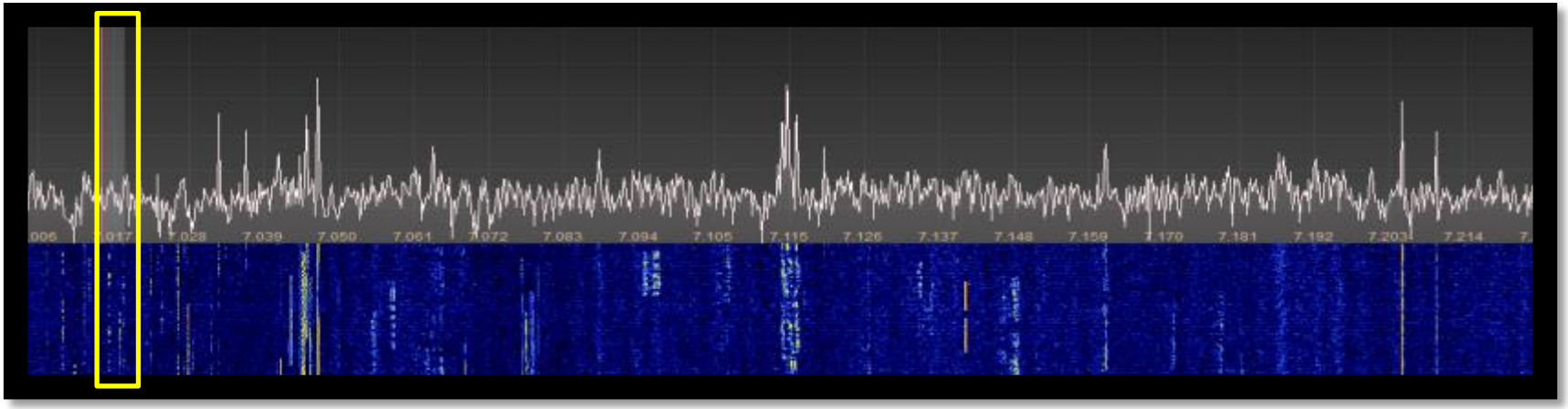
FFT

FFT



- Flux découpé en morceaux
- Calculs en continu
- A chaque coup, $n.\log_2(n)$ opérations

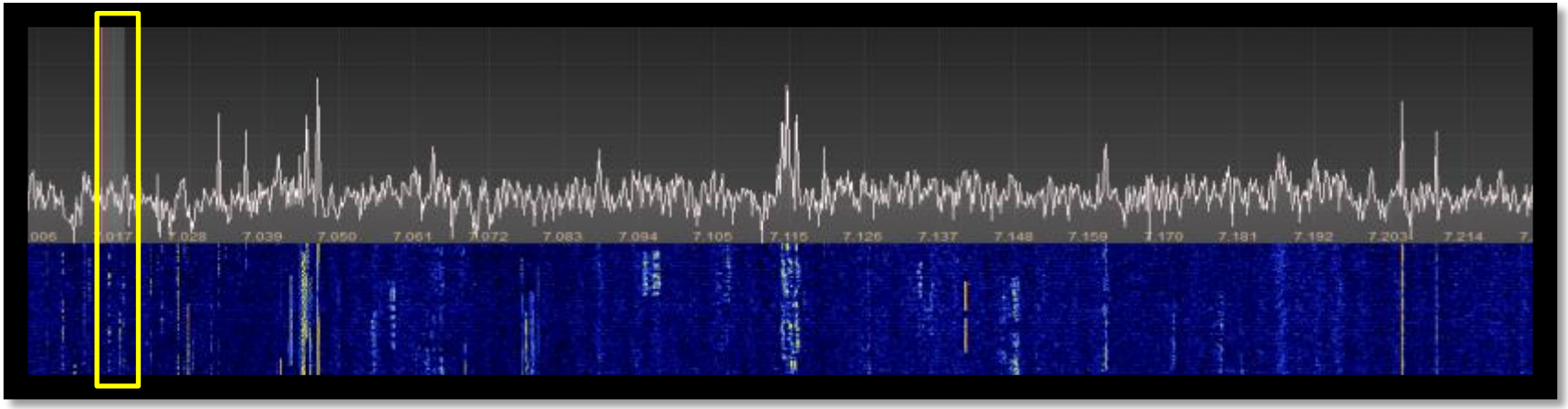
Sélectionner la bande à démoduler

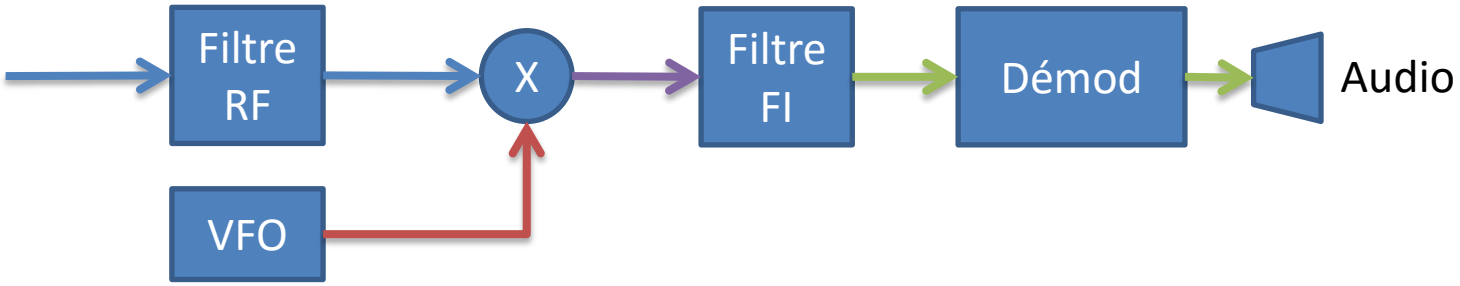


- On veut « extraire » une partie de la bande reçue (par exemple 3300 Hz en BLU) pour ensuite extraire l'audio (démoduler)

- Deux premières étapes :
  1. « Tuner » sur la bonne bande,
  2. Appliquer un filtrage passe-bande

Sur un récepteur analogique

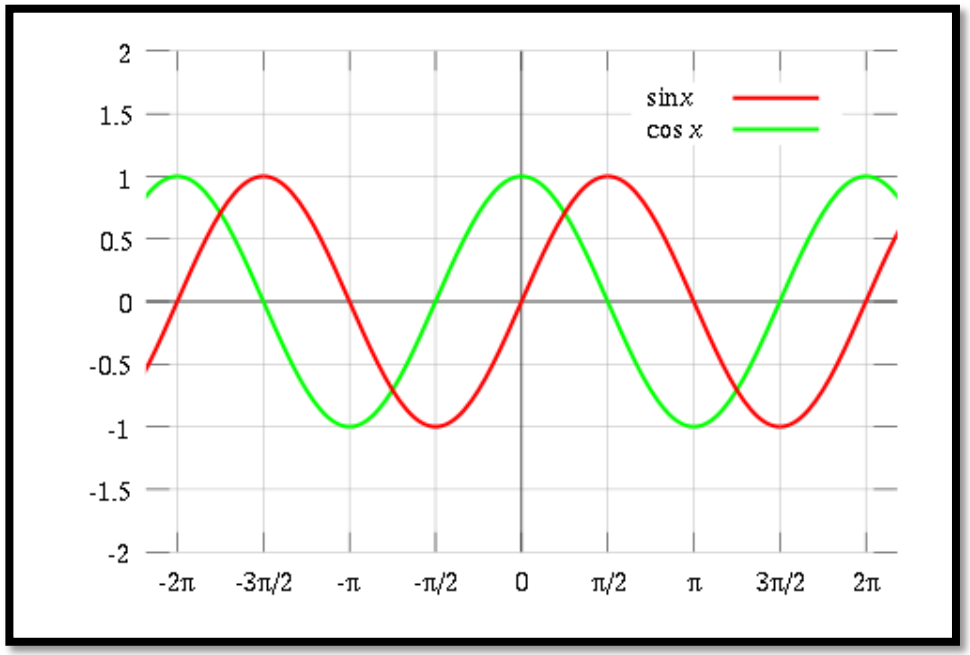




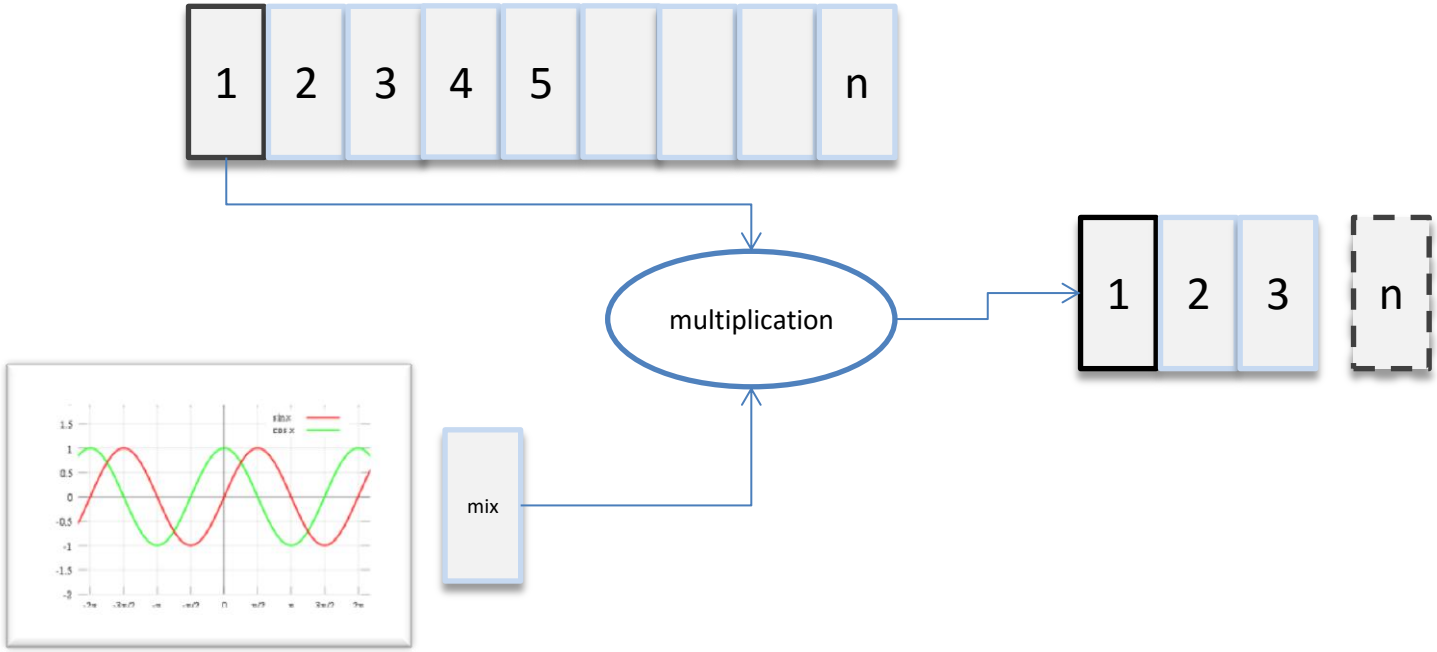
- La sélectivité dépend de la qualité des filtres
- Le bruit de phase / stabilité du VFO est critique

Le VFO

- L'objectif est de générer un signal de mixage le plus propre possible
- Très simple dans un SDR grâce à l'emploi de deux formules « simple » :
  - cos(x) et sin(x)
  - Exactement déphasées de 90°

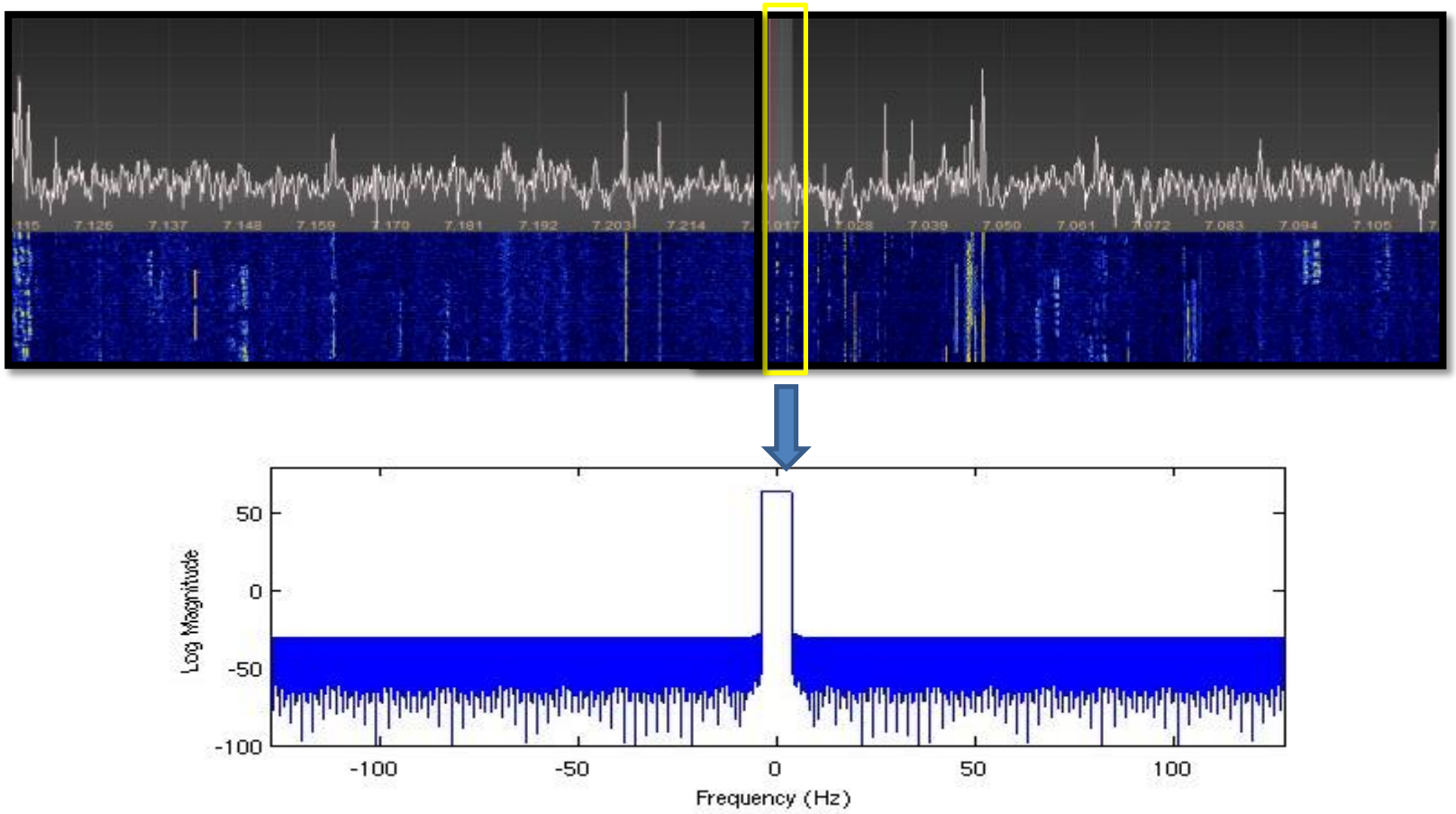


Le mixage



- En sortie de ce calcul; le spectre est « décalé » et recentré sur la fréquence de mixage
- Une multiplication par échantillon + calcul sin/cos

Extraction de la bande « utile »

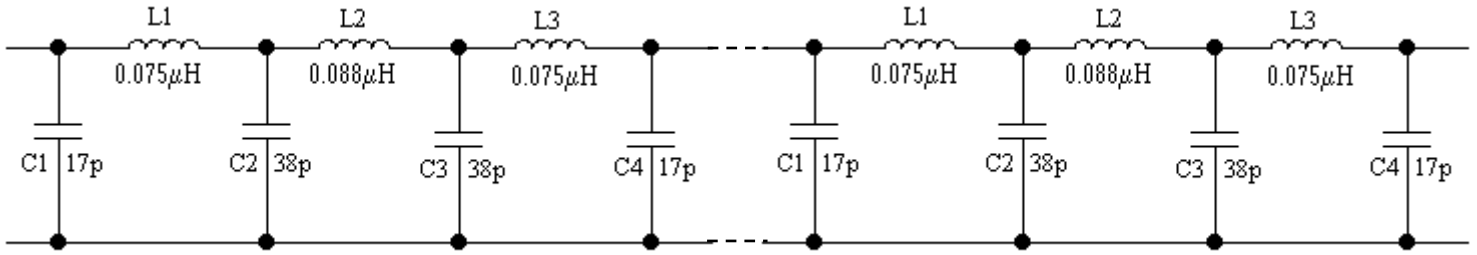


On a donc besoin d'un filtre qui va atténuer tout ce qui est « hors bande »

Introduction au filtre FIR

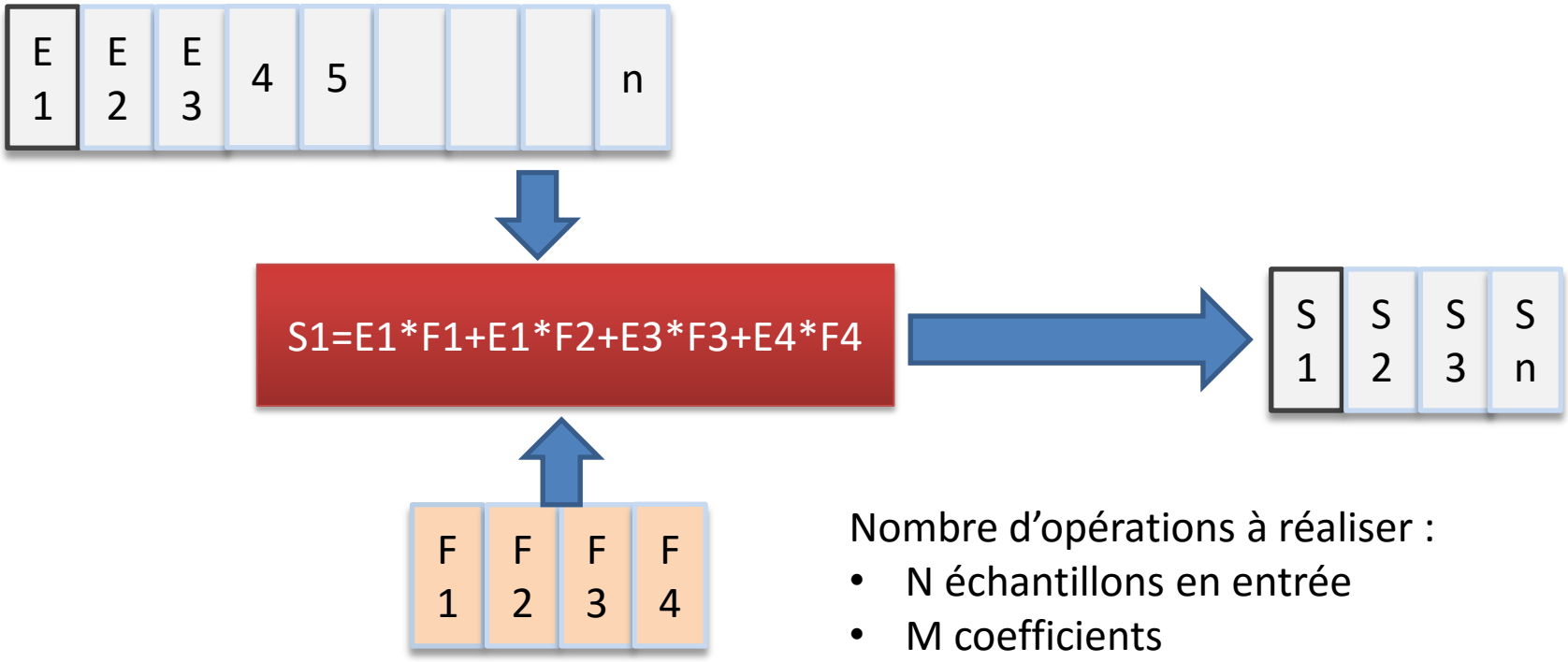
- FIR = Finite Impulse Response Filter

- En pratique : un « **certain nombre de coefficients** » bien choisis pour permettre de supprimer les composantes « hautes fréquences » (au-delà de la fréquence de coupure)
- Comme en analogique : *plus c'est long plus c'est bon*
  - Il faut beaucoup d'étages pour être sélectif (ou utiliser un filtre à Quartz)



Le filtrage « FIR »

- Opération répétée **pour chaque échantillon**



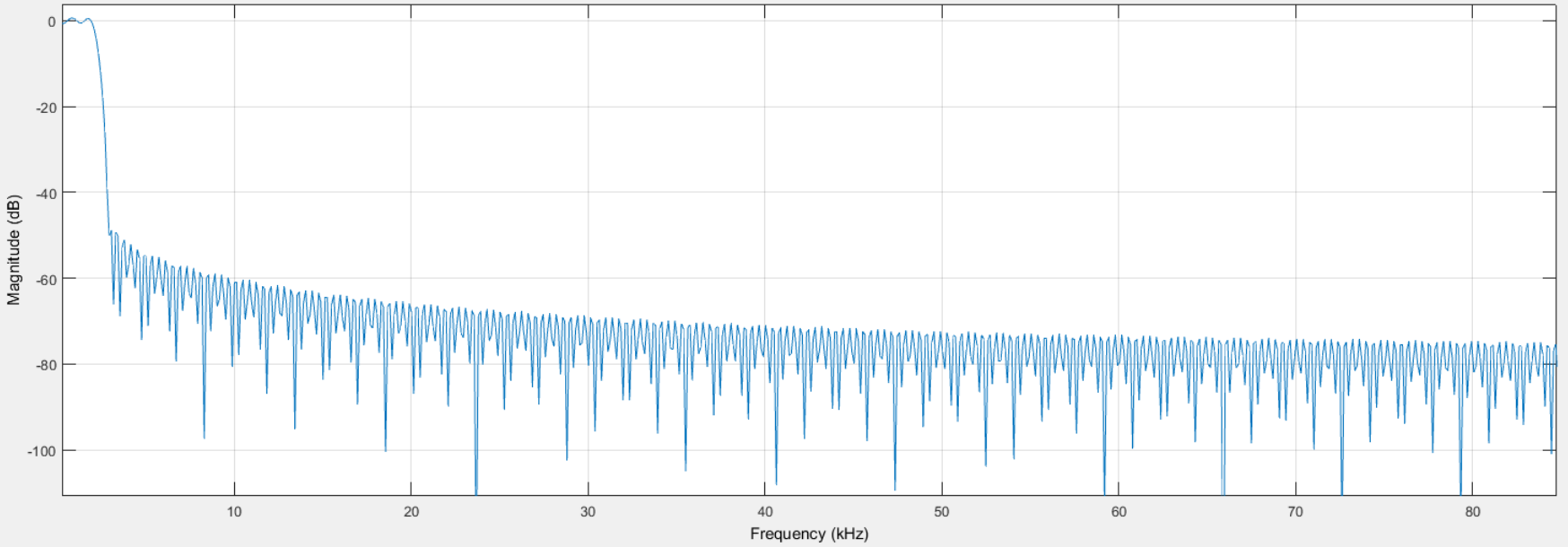
Filtre à 4 coefficients

Nombre d'opérations à réaliser :

- N échantillons en entrée
- M coefficients
- Alors on fait **N*M multiplications et M additions** **(que l'on ne compte pas…)**

Avoir la bonne longueur…

- Combien de coefficients faut il ?



Exemple ici : 2MHz de bande en entrée, on veut 3KHz de bande

Il faut… 5060 coefficients

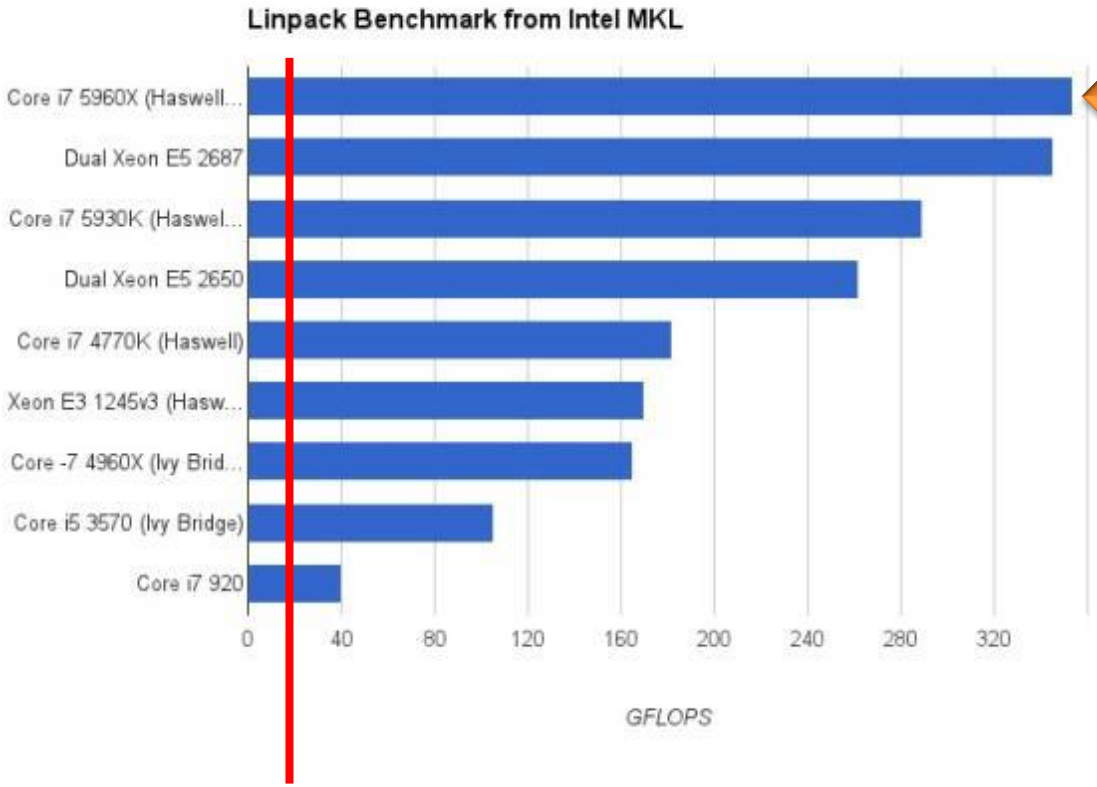
5060 coefficients, et alors ?

- Avec 2 millions d'échantillons par seconde (IQ…) , on a donc 4 millions de valeurs nouvelles par seconde à traiter
- Chaque valeur doit être multipliée par un filtre de longueur 5060
- Ça fait donc : (4 million)x(5060) = 20,2 Milliards de multiplication par seconde = 20 « gigaFlops »

Pour « à peine » 50 dB de réjection hors bande… avec tout juste 2MHz de bande en entrée…

20 Giga Flops pour filtrer ?

A part sur un processeur « haut de gamme », on consomme la majorité de la puissance



Prix au 01/05/2016 : 1200 €

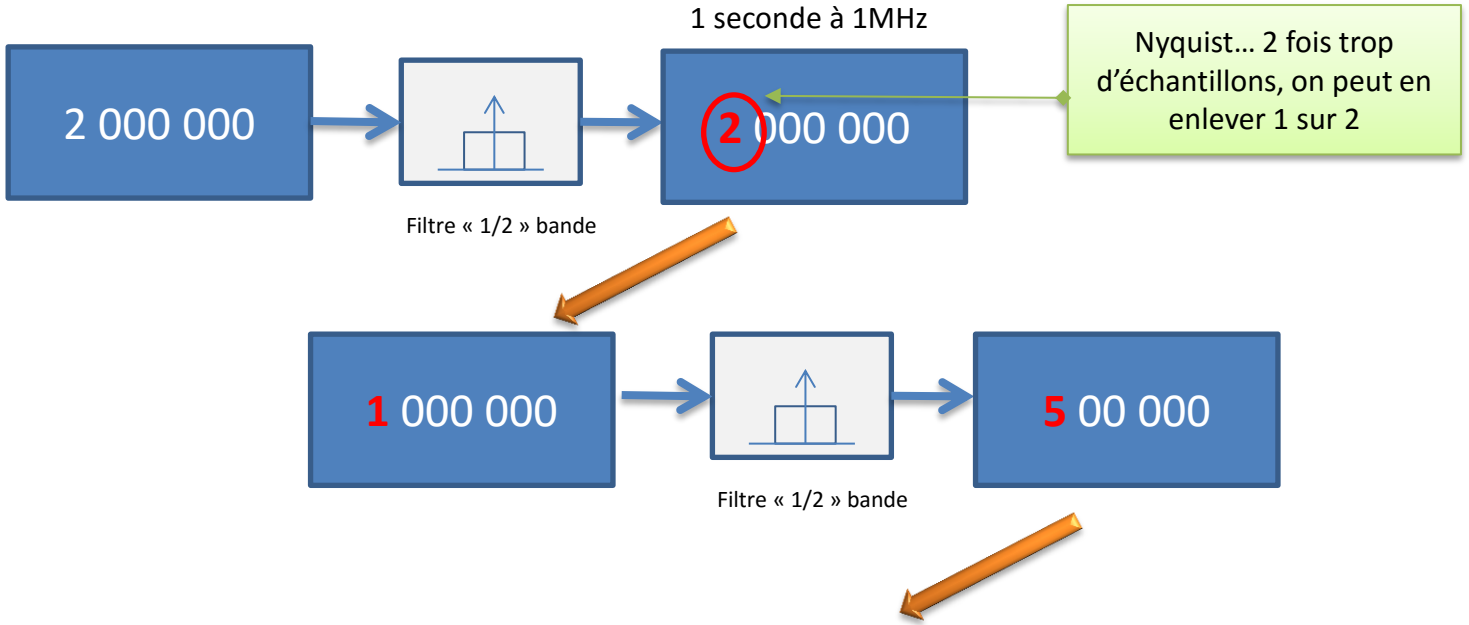
Conclusion… faut pas faire comme ça !

Diviser pour régner

- **Nyquist** dit « au moins 2fmax »
  - 1 seconde à 2MHZ = 2000000 échantillons
  - 1 seconde à 4KHz = 4000 échantillons
  - Après le filtrage on a 200000/4000=500 (au lieu de 2), on est large…

- L'astuce consiste à enchaîner des réductions par 2 = la décimation

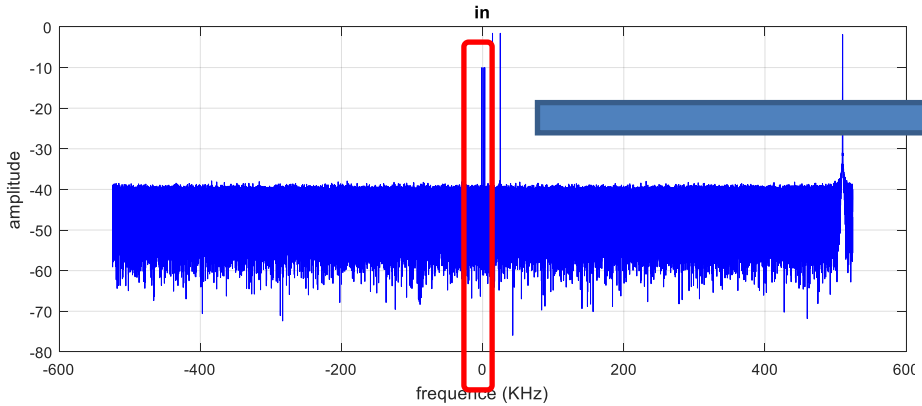
Décimation

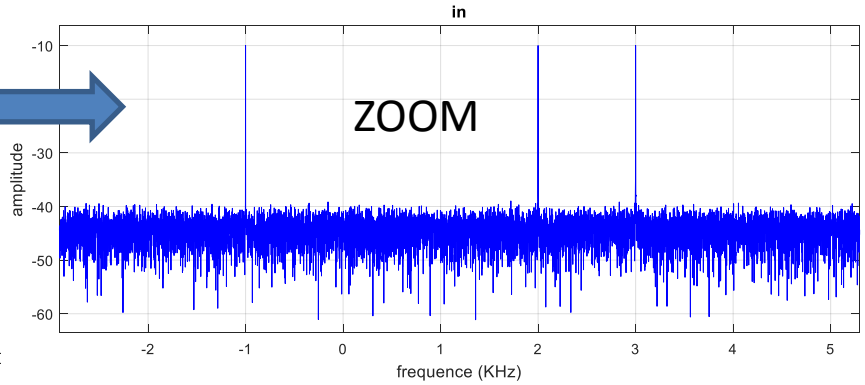


- En répétant cette opération et avec des filtres bien choisis (de plus en plus sélectifs) on arrive à une bonne réjection et des temps de calcul raisonnables

- Par exemple dans gkSDR : Premier filtre à 7 coefficients, puis 11, …. , puis 43 et 51 pour terminer

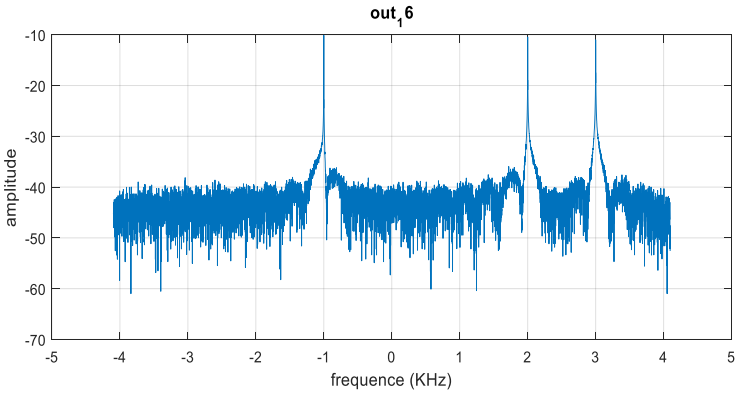
Exemple de traitements



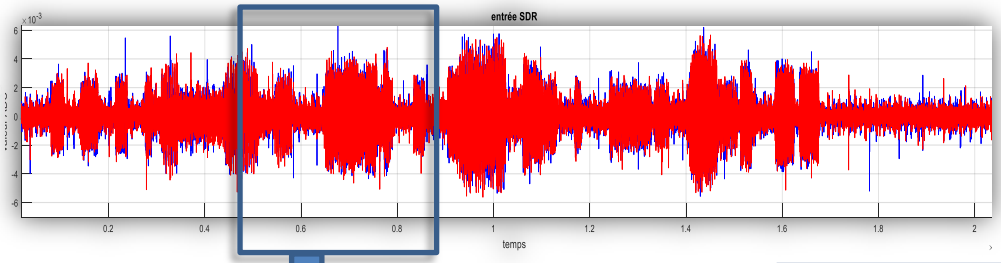


Extraction de 8KHz de bande dans 1MHz (décimation par 128) :

- **7 étages** de décimation par deux en cascade,
- Sur 1 seconde de signal, utilise **34 millions** de multiplications (au lieu des « milliards » nécessaires si on y va « sans réfléchir ») gain > 1000 !
- Tout le problème est ici de dimensionner correctement les étages de filtrage

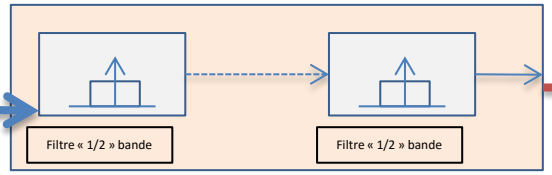


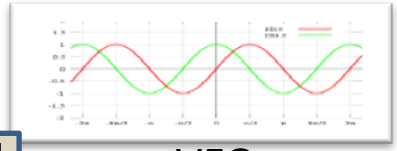
En résumé jusqu'à présent…



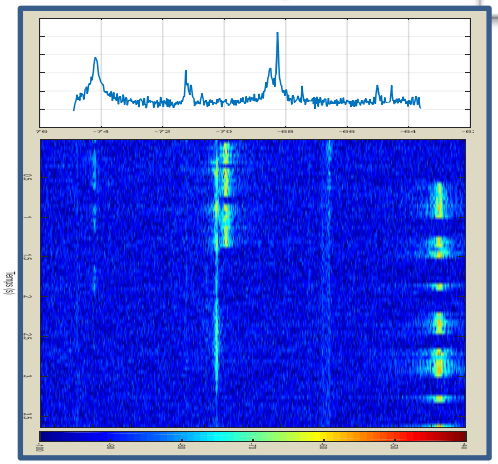
FFT

mix





VFO



$$A = \sqrt{I^2 + Q^2}$$

$$\phi = atan\left(\frac{I}{Q}\right)$$

Démodulateurs

Audio

Transfert des données entre blocs

Dans un équipement « non logiciel » les données RF sont acheminées par une ligne de transmission (coax, piste etc.)

- Dans un logiciel SDR, les données entrent **par paquets** (USB, réseau, etc.) et sont envoyées par paquets à la carte son, au disque dur
- La difficulté est que les paquets répartis entre les différentes étapes du traitement entraînent un **temps de latence**.

- A titre d'information :
  - La majorité des « API » audio utilisent des buffers de quelques Ko (2048 par exemple)
  - Si les données audio ne sont pas envoyées au bon moment, ça s'entend (craquements)
  - Il faut donc dimensionner les traitements « à l'envers »… 1000 échantillons audio à 48 KHz représentent 1/48ème de seconde, soit sur un SDR à 2MHz environ 42000 echantillons, et 200 000 pour 10 MHz de bande

3ème PARTIE

A quoi ressemblera le logiciel SDR de demain ?

Les prochains SDR ?

- Aujourd'hui les limites principales sont :
  - La puissance de calcul des ordinateurs « personnels »,
  - Le débit des interfaces d'entrée / sortie,

- On aimerait bien :
  - Plus de bande,
  - Plusieurs récepteurs en même temps (un par bande OM ?)
  - Plusieurs antennes

Plus de puissance de calcul

- **Le jeu sur PC** représente un marché important, gourmand en 3D, en calcul rapides… Les fabricants de processeur ont investi massivement dans des processeurs spécialisés pour l'affichage 3D
- Progressivement ces processeurs spéciaux se sont « ouverts » pour y faire fonctionner des codes « non graphiques » : les **GPU**

Cartes GPU Nvidia ou ATI

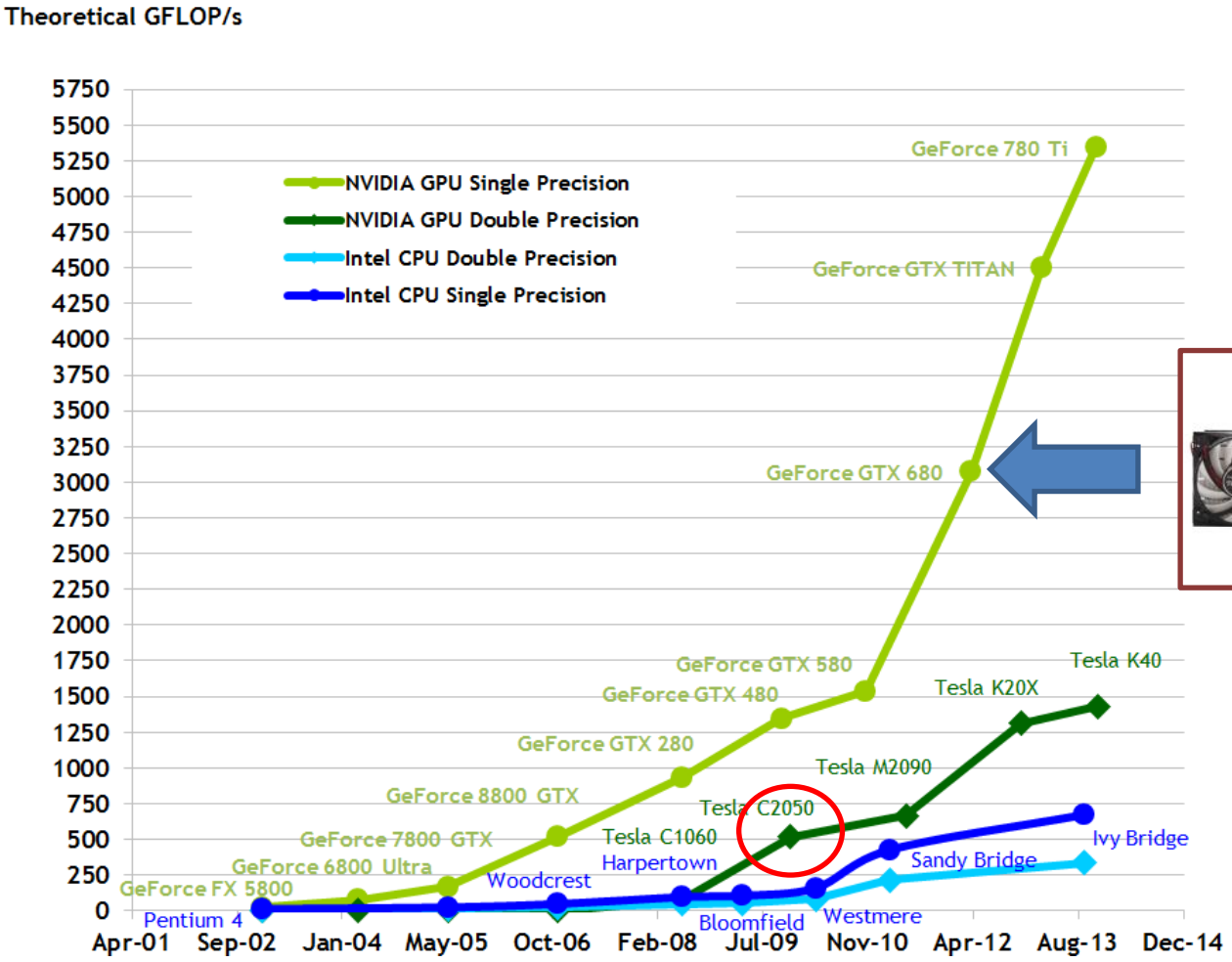
Certaines cartes GPU (ici NVIDIA TESLA) ne comportent même plus de sortie vidéo.
Des outils de développement spécifiques sont maintenant disponibles (CUDA, OpenCL)

Exemple NVIDIA

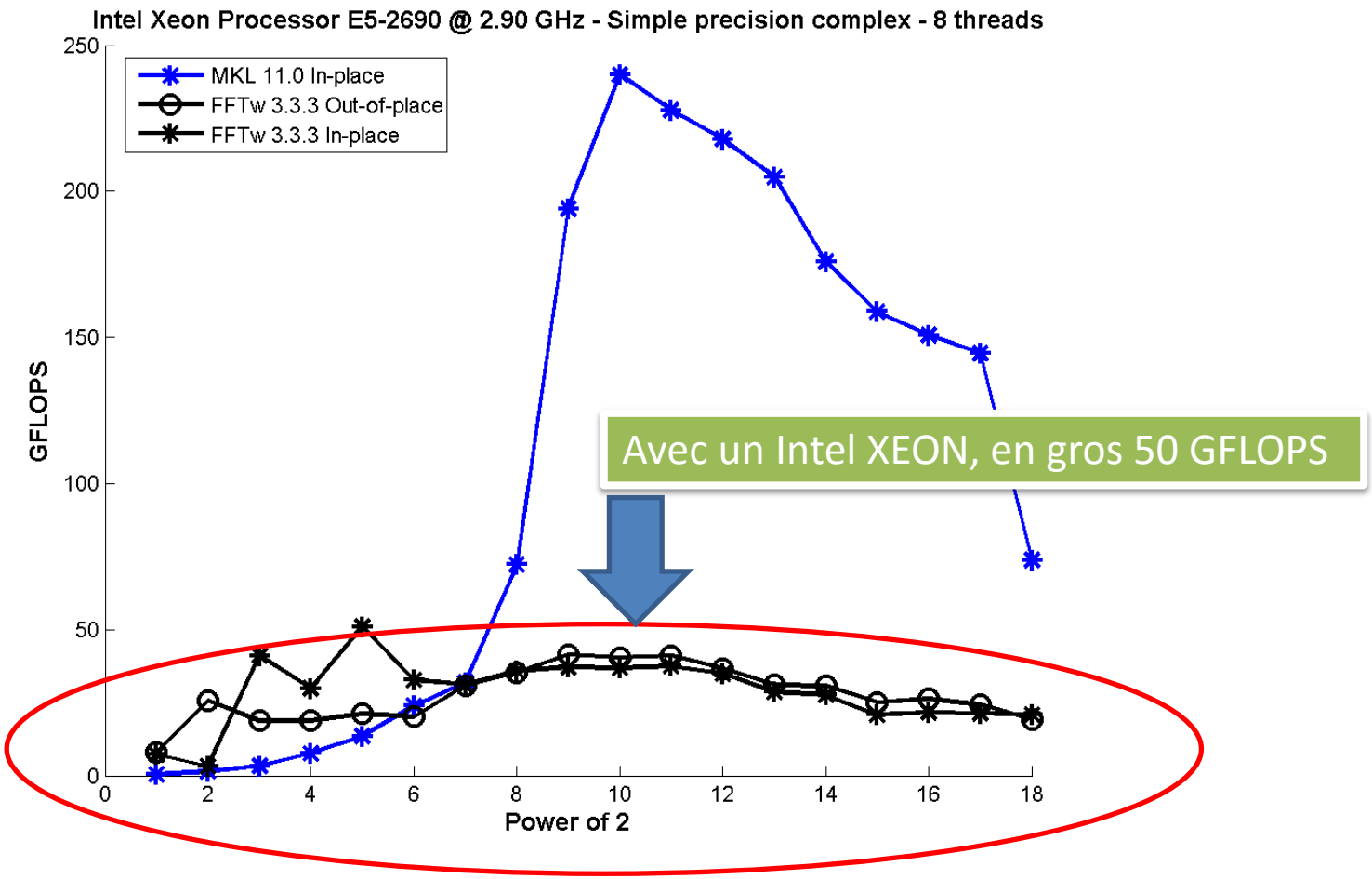
- Carte TESLA C2075 (déjà très ancienne… sur Ebay se trouve à moins de 200 euros)
  - **448** processeurs de calcul à 800 MHz,
  - 6 GB de mémoire embarquée sur la carte,
  - Capable de transferts à 5 Gb/s avec le processeur principal
- Carte TITANX : **3072** processeurs à 1GHz !!!
  Pour 1100 euros environ (neuf, au 01/05/2016)

L'arrivée des GPU

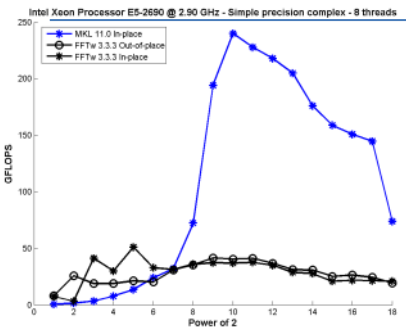
1 GFLOP/s = **1 milliard** de multiplication « en virgule flottante » par seconde

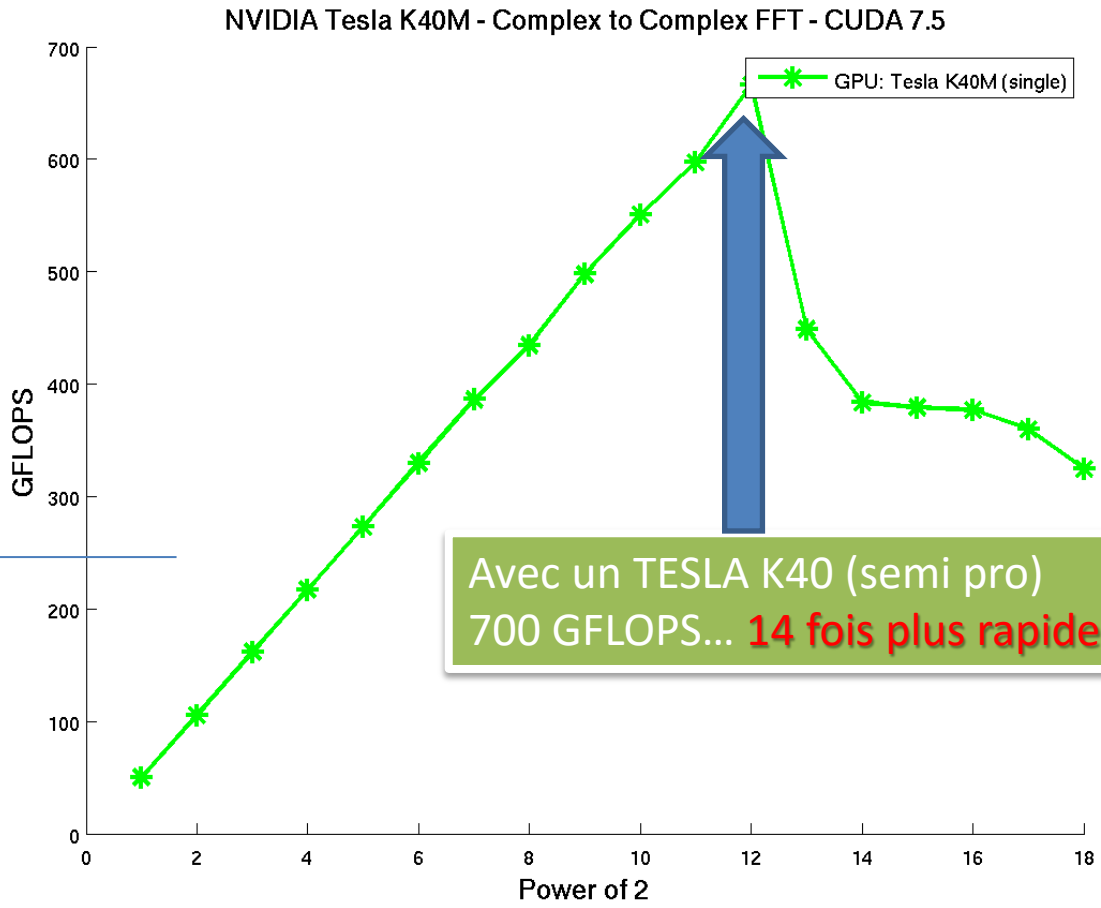


Souvenez vous de la FFT…



FFT sur GPU


Intel Xeon Processor E5-2690 @ 2.90 GHz - Simple precision complex - 8 threads
MKL 11.0 In-place
FFTw 3.3.3 Out-of-place
FFTw 3.3.3 In-place
GFLOPS
Power of 2


NVIDIA Tesla K40M - Complex to Complex FFT - CUDA 7.5
GPU: Tesla K40M (single)
GFLOPS
Power of 2
Avec un TESLA K40 (semi pro)
700 GFLOPS… 14 fois plus rapide !

Pour quoi faire ?

- Le ***waterfall*** ultra haute résolution en temps réel,
- Le filtrage de la bande utile grâce à des algorithmes spécifiques plus adaptés au « calcul massif »
- Plusieurs sous-bandes en même temps à partir de récepteur SDR large bande (AirSpy à 10 MHz etc).

- Pas encore dispo dans les logiciels amateur (bientôt dans gkSDR… en test ☺), sauf pour l'affichage du waterfall dans certains logiciels.

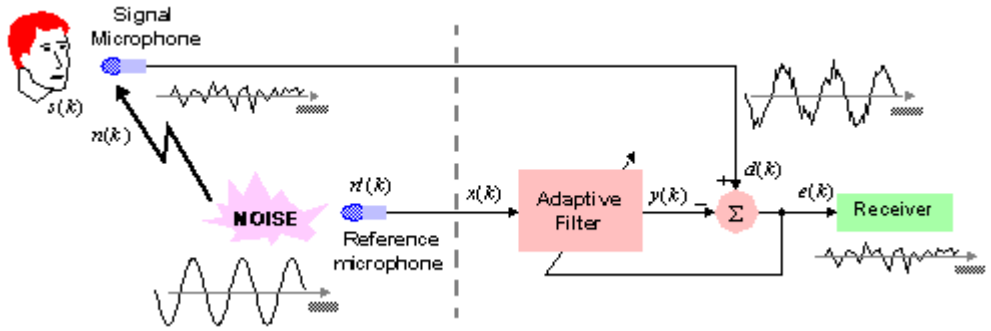
Plusieurs voies de réception

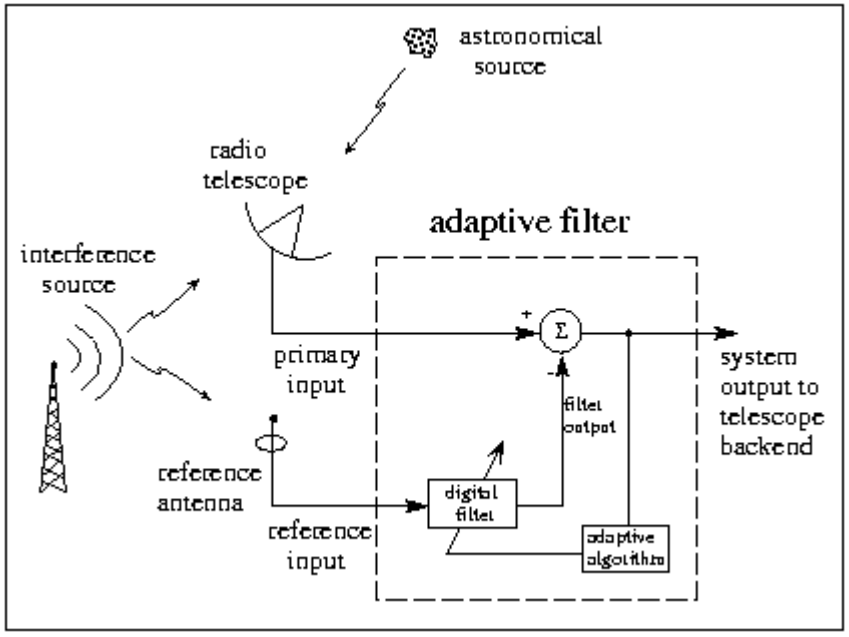
Le surplus de puissance permet également d'imaginer l'entrée dans le monde amateur de techniques « pro » :

- Filtrage automatique évolué,
- Formation de faisceau par le calcul

Filtrage adaptatif

- Exemple en audio (facile vu la bande) : suppression de bruit ou d'écho



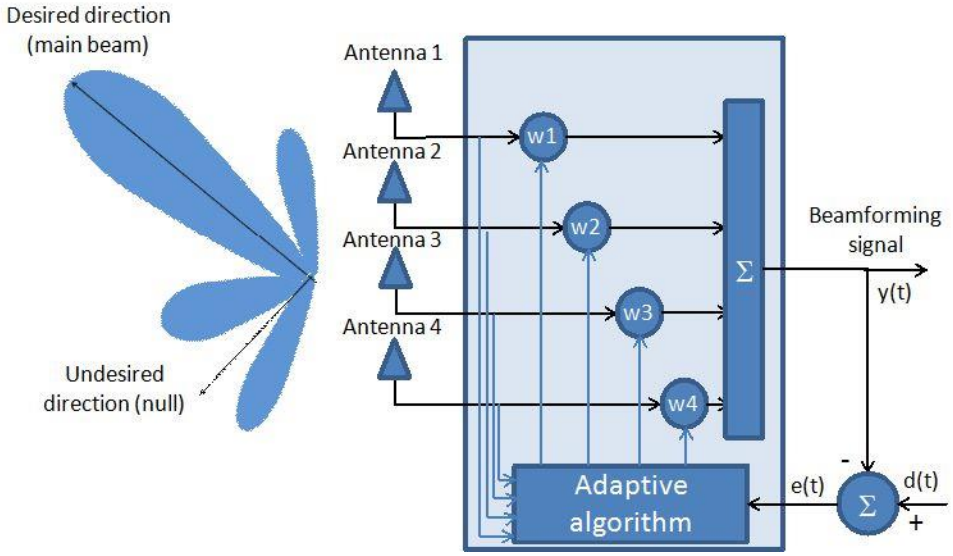


Appliqué en « radio astronomie » en POST-TRAITEMENT, sur les données enregistrées…

Le GPU permet de le faire « en live »

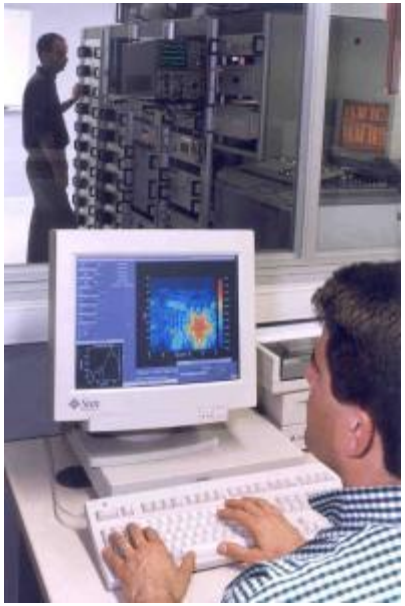
FFC

- Formation de faisceau par le calcul (FFC) : synthétiser une antenne directive à partir de « N » antennes fixes



Pensée pour les collègues Onera

Photos ONERA



« Quelques récepteurs SDR sous une forêt d'antennes décamétriques »

Pour aller plus loin

Lire la doc de CuteSDR par Moe Wheatley AE4JY

- https://sourceforge.net/projects/cutesdr/files/doc/CuteSDR102.pdf/download

« Software Defined Radio for the Masses » de AC5OG, 4 parties, également traduit dans RREF :

1. https://www.arrl.org/files/file/Technology/tis/info/pdf/020708qex013.pdf
2. https://www.arrl.org/files/file/Technology/tis/info/pdf/020910qex010.pdf
3. http://www.arrl.org/files/file/Technology/tis/info/pdf/030304qex020.pdf
4. https://www.arrl.org/files/file/Technology/tis/info/pdf/021112qex027.pdf

FIN !!!

Merci de votre attention !