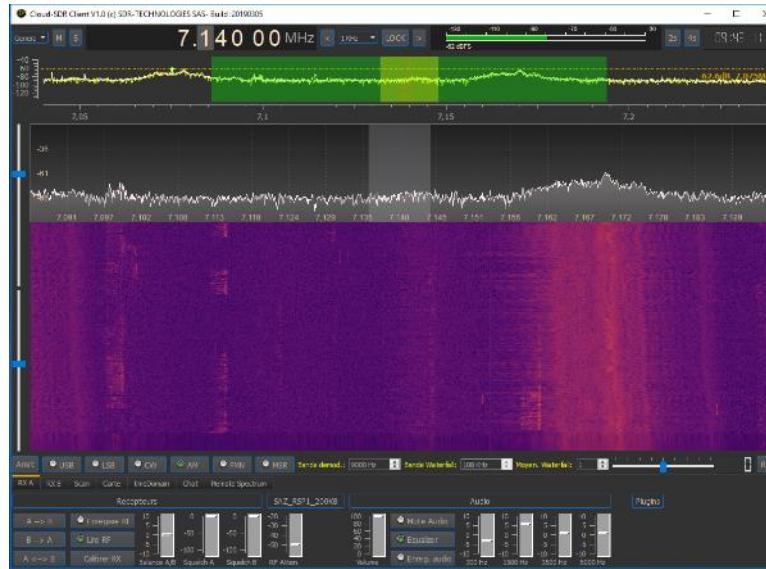


# SDR : La radio logicielle

Sylvain F4GKR



# Objectif de cette présentation



- Comprendre comment ça fonctionne et pourquoi c'est fait comme ça
- Présentation axée sur la réception

# Objectif de cette présentation

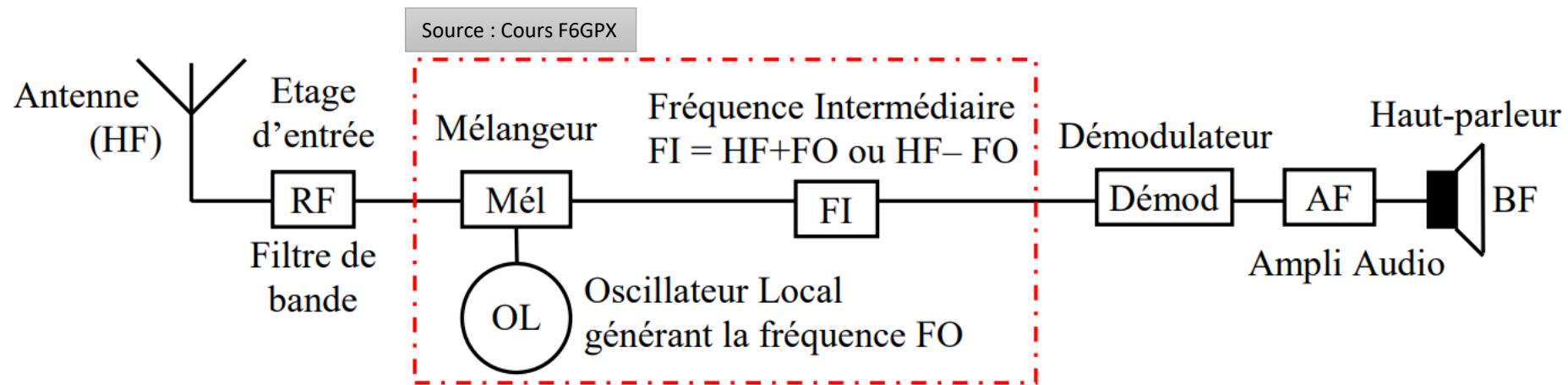
Avoir une meilleure compréhension de ce que ces « concepts » veulent dire quand on parle de SDR:

- Réception à conversion directe
- Echantillonnage complexe
- IQ
- FFT
- Waterfall
- Equilibrage IQ
- Echantillonnage direct
- FPGA
- Filtrage FIR

# Introduction

Retour aux sources: les bons synoptiques qu'on aimait tant

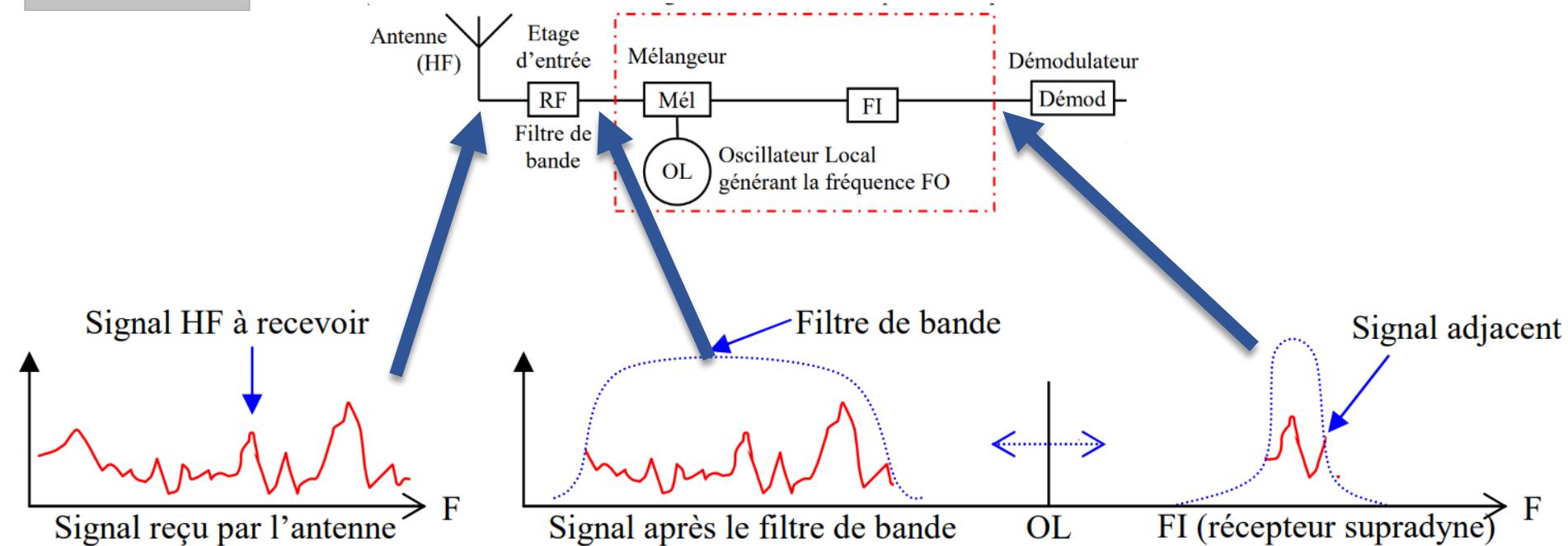
# Récepteur radio



Synoptique d'un récepteur radio analogique à simple conversion de fréquence

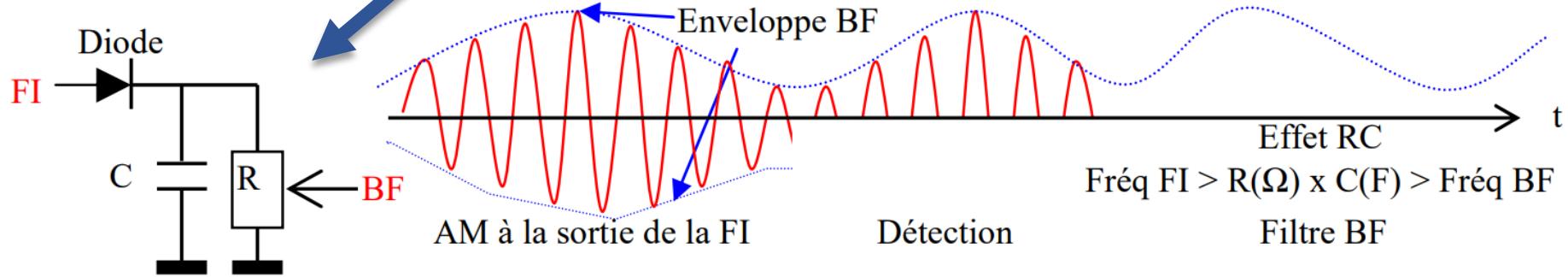
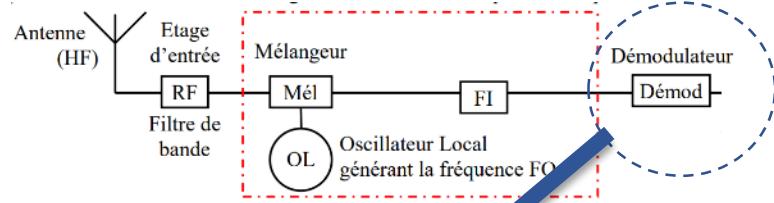
# Sélection du signal utile

Source : Cours F6GPX



- Filtrage puis transposition de fréquence
- Démodulation en « Fréquence Intermédiaire »

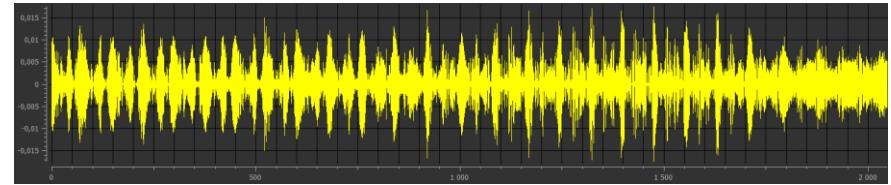
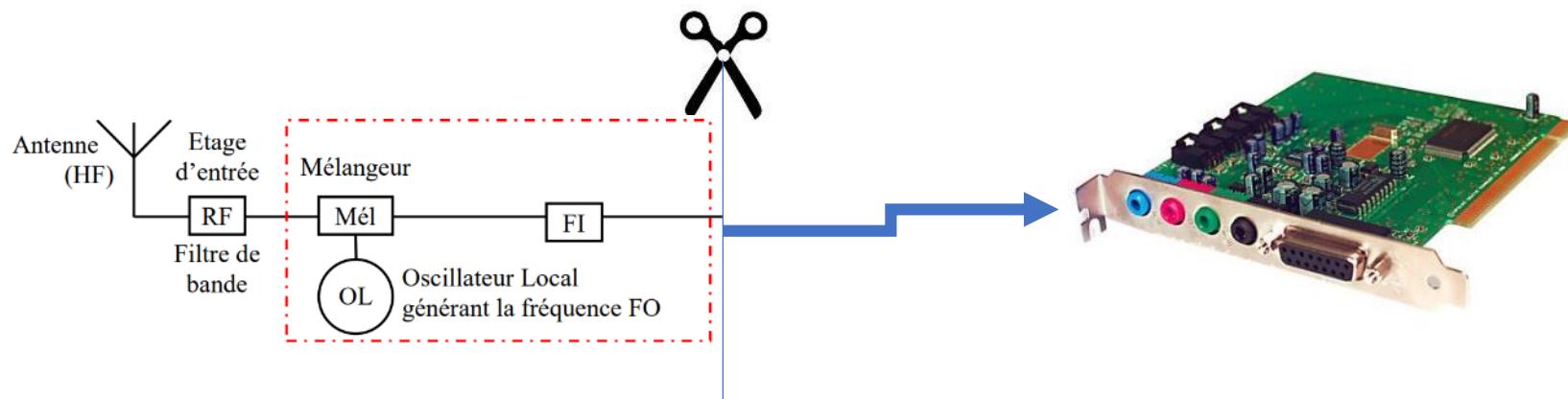
# Exemple : démoduler l'AM



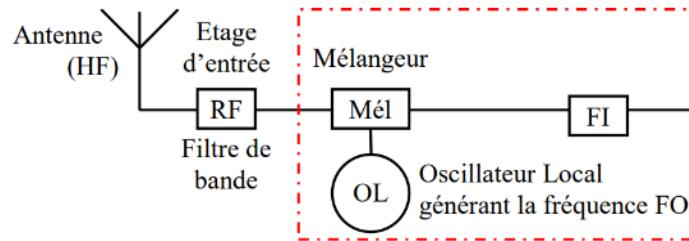
Démodulation d'amplitude par diode puis filtrage passe-bas (intégrateur RC)

# Numériser la FI

Première approche : numériser le signal FI par une carte son d'un ordinateur  
Puis traiter le signal audio enregistré pour le démoduler

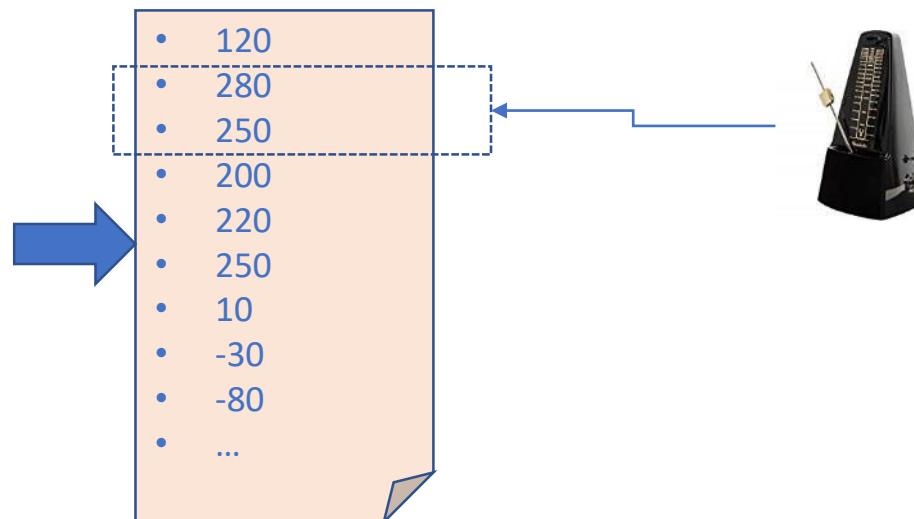
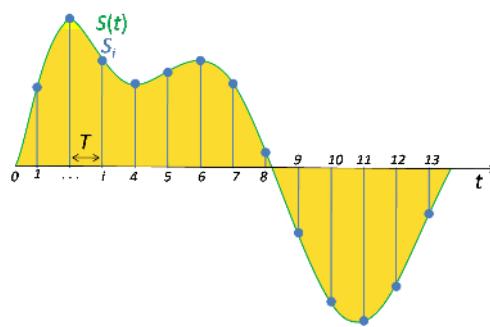


# Le processus de numérisation



- Echantillonnage : mesure régulière de la tension observée
- En entrée : un signal **continu**
- En sortie : un signal **discret**

# Le signal discret



Le « signal discret » n'est qu'une suite de nombre, représentant la tension mesurée à chaque « top » de l'horloge

# Quizz

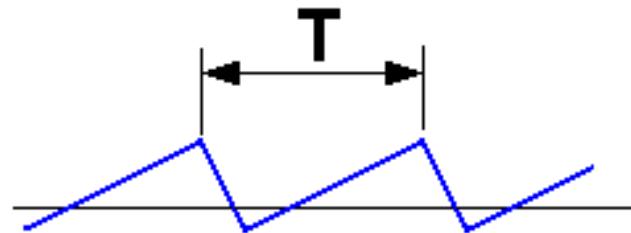
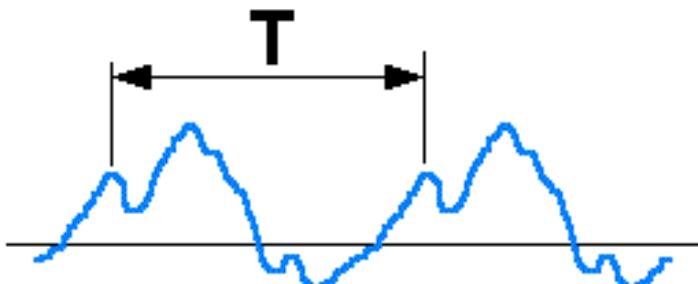
- Quelle précision de mesure faut-il (sur le voltmètre) ?
- Faut-il que le métronome soit régulier ?
- A quelle cadence faut-il régler le métronome ?
- Quelle est la bonne fréquence pour la FI (fréquence intermédiaire)
- C'est quoi « échantillonnage complexe » ? C'est quoi un « nombre réel »?

# Fréquence d'échantillonnage

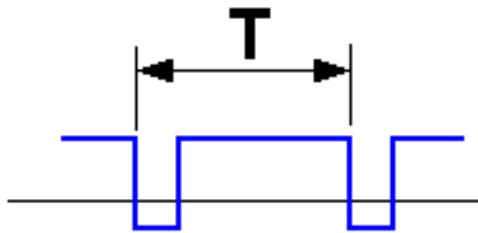
Quand ça plie et ça replie...

# Signal périodique

- Un signal est dit **périodique** si les variations de son amplitude se reproduisent régulièrement au bout d'une période  $T$  constante



On sait que le signal est périodique si on l'observe pendant « assez longtemps »

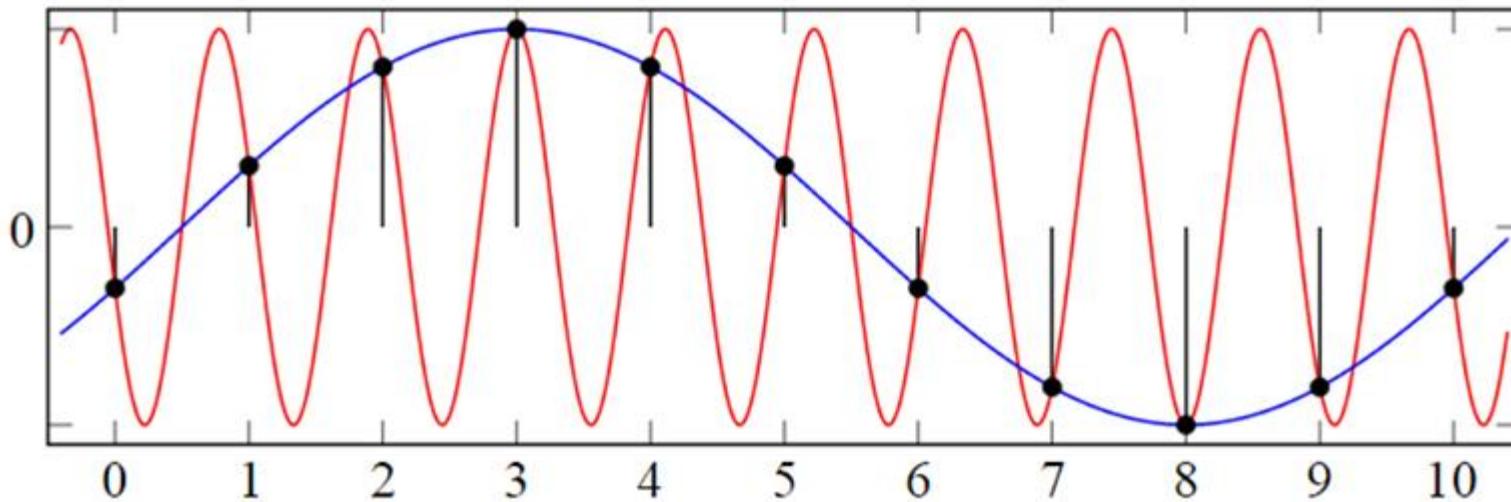


→ Si on observe pendant  $T$ , on ne peut pas être sûr que le signal se répète après... Pour une observation de durée  $T$ , la plus grande période observable est donc  $T/2$

# Nyquist et Shannon

- La durée de l'observation est fixée à  $T$
  - Le théorème de Nyquist-Shannon dit:
    - Si un signal  $x$  ne contient pas de fréquence supérieure à  $1/T$ , il est alors complètement déterminé si l'on prend **2 mesures pendant  $T$** .
    - Cette limite  $f=1/T$  est la fréquence de Nyquist
- Il faut **2 mesures** pour connaître un signal

# Aliasing (le repliement)



- S'il reste à l'entrée de l'ADC des signaux de fréquence supérieures à Nyquist, il seront numérisés et se retrouveront dans le spectre avec une fréquence apparente (alias) différente;
- D'où l'importance du filtre avant numérisation.

# Echantillonnage complexe

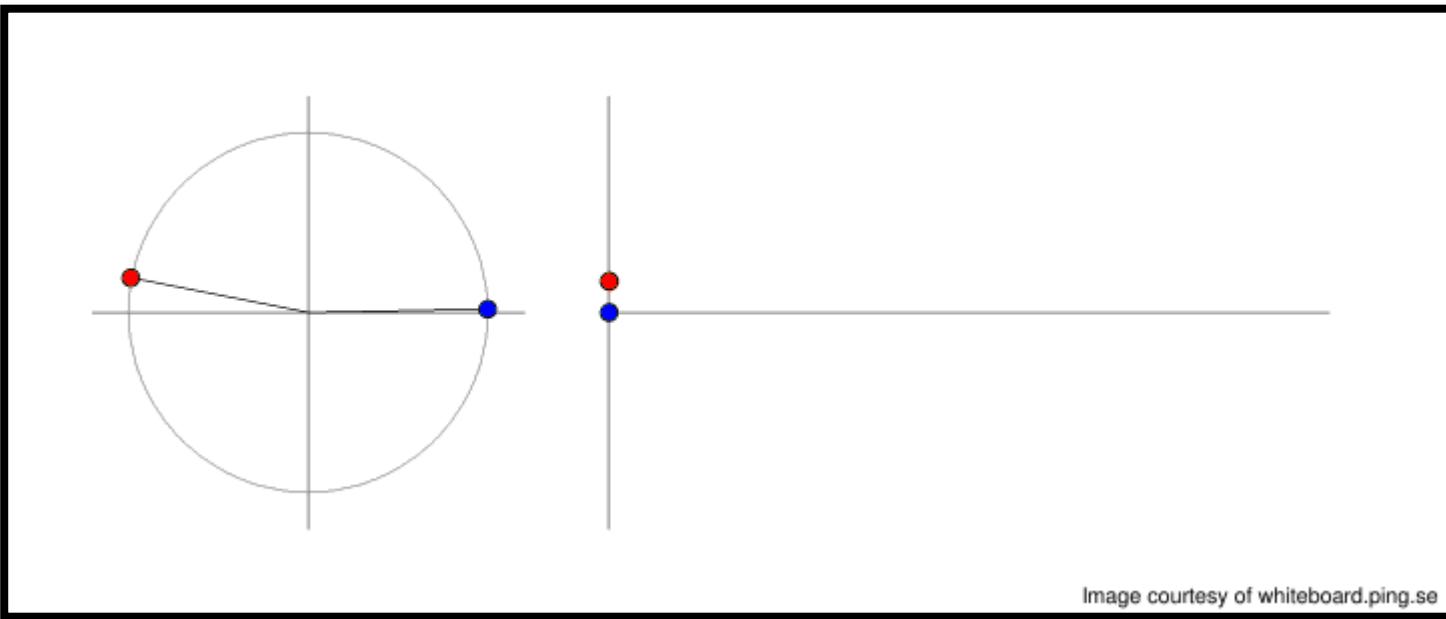
Les nombres imaginaires au secours d'un problème qui n'a pas de sens

# 2 mesures ?

- On a dit : Un échantillonnage régulier de  $2N$  mesures (échantillons) permet de décomposer le signal reçu en **N sinusoïdes** élémentaires
- Ça fonctionne mais... pas tout à fait (désolé)

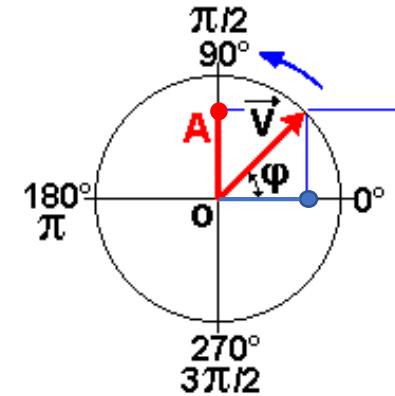
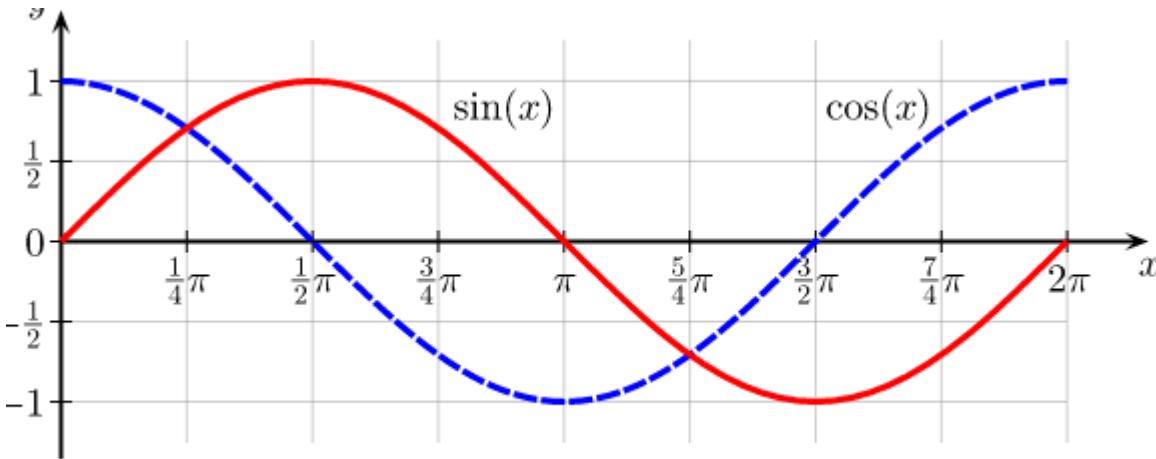


# 2 mesures... Oui ou non ?



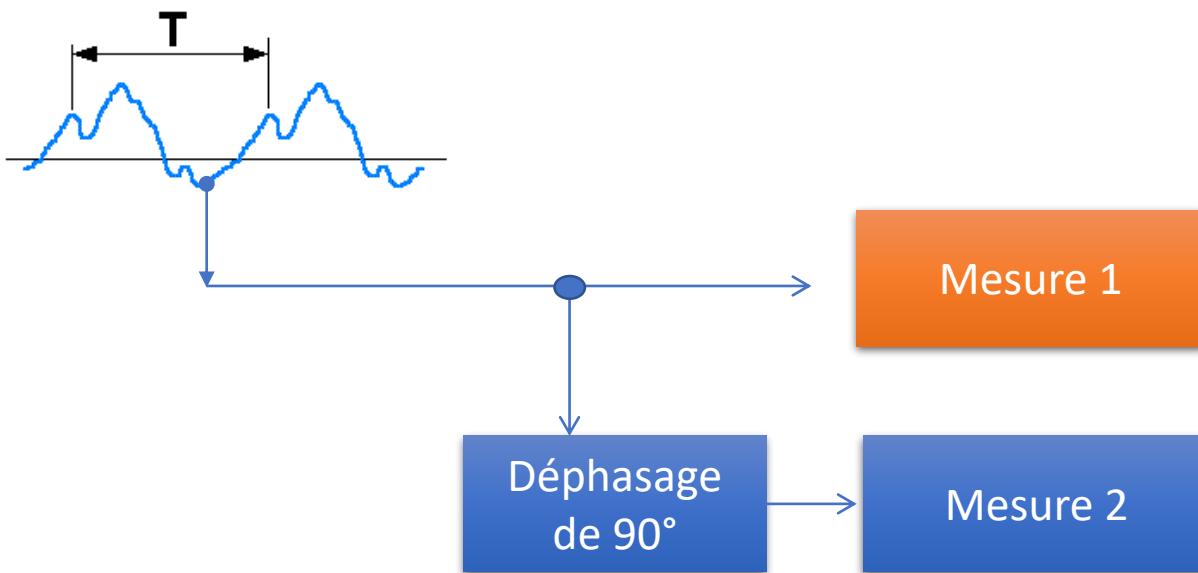
- Les deux signaux n'évoluent pas « dans le même sens » et pourtant leur représentation temporelle est identique (donc ça va pas)

# $\sin(x)$ et $\cos(x)$



- « sinusix » et « cosinusix » sont les mêmes courbes, mais déphasées de  $90^\circ$
- Si on mesure aussi (en même temps) le rouge ET le bleu, on a le sens de rotation

# En pratique on fait



- Le signal à quantifier est déphasé de  $90^\circ$  puis mesuré



# Signal réel ou complexe

- 2 mesures **consécutives** du signal : l'échantillon est « **réel** »
- 2 mesures **en même temps** du signal déphasé de  $90^\circ$  : l'échantillon est « **complexe** », on parle aussi d'échantillons « **IQ** »
- Si on garde la même cadence de mesure dans les deux cas, *on a « doublé » la fréquence maximum mesurable...*

BONUS

# En prime

- L'échantillonnage « complexe » permet de savoir « dans quel sens ça tourne »,
- 1 tour par seconde = 1 Hz



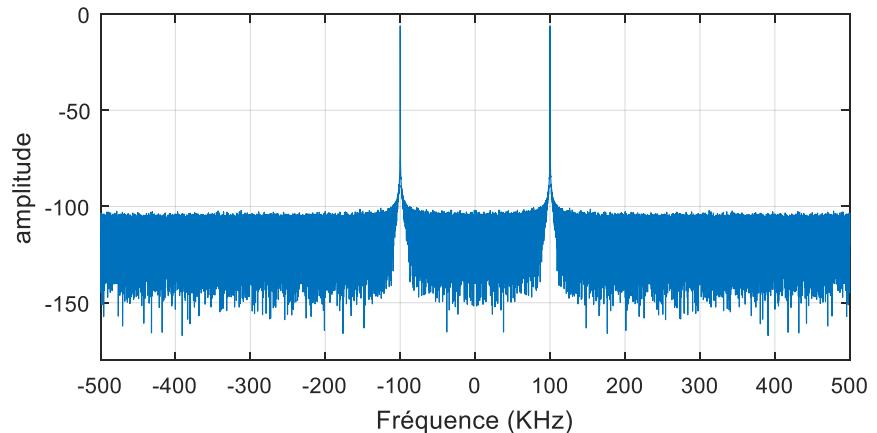
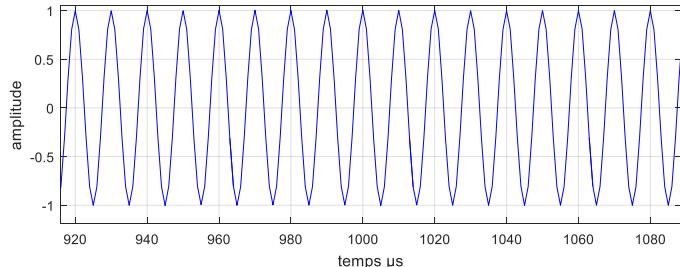
- « Du coup »... on a :
  1. Des fréquences « positives » (ça tourne dans un sens)
  2. Des fréquences « négatives » (ça tourne dans *l'autre* sens)

Signal complexe = « IQ » = signaux « en quadrature »

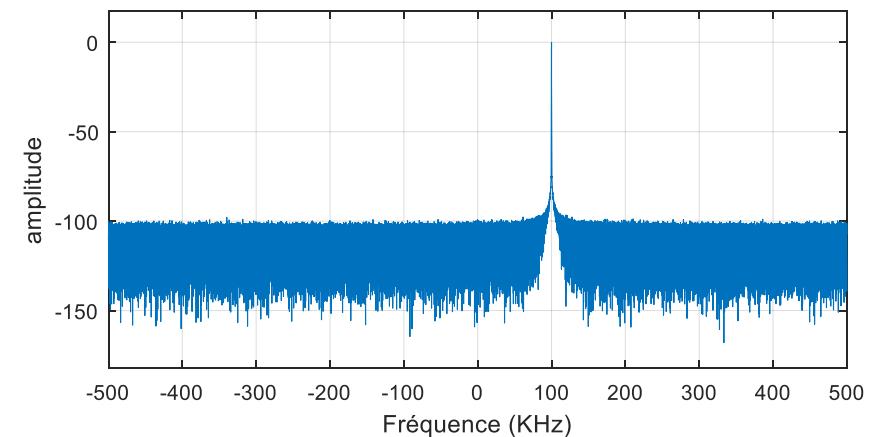
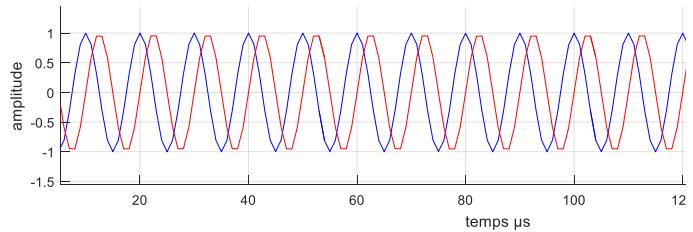
Ce sont les mêmes notions

# Spectre réel / spectre complexe

Signal réel (pas de IQ)



Signal complexe (IQ)

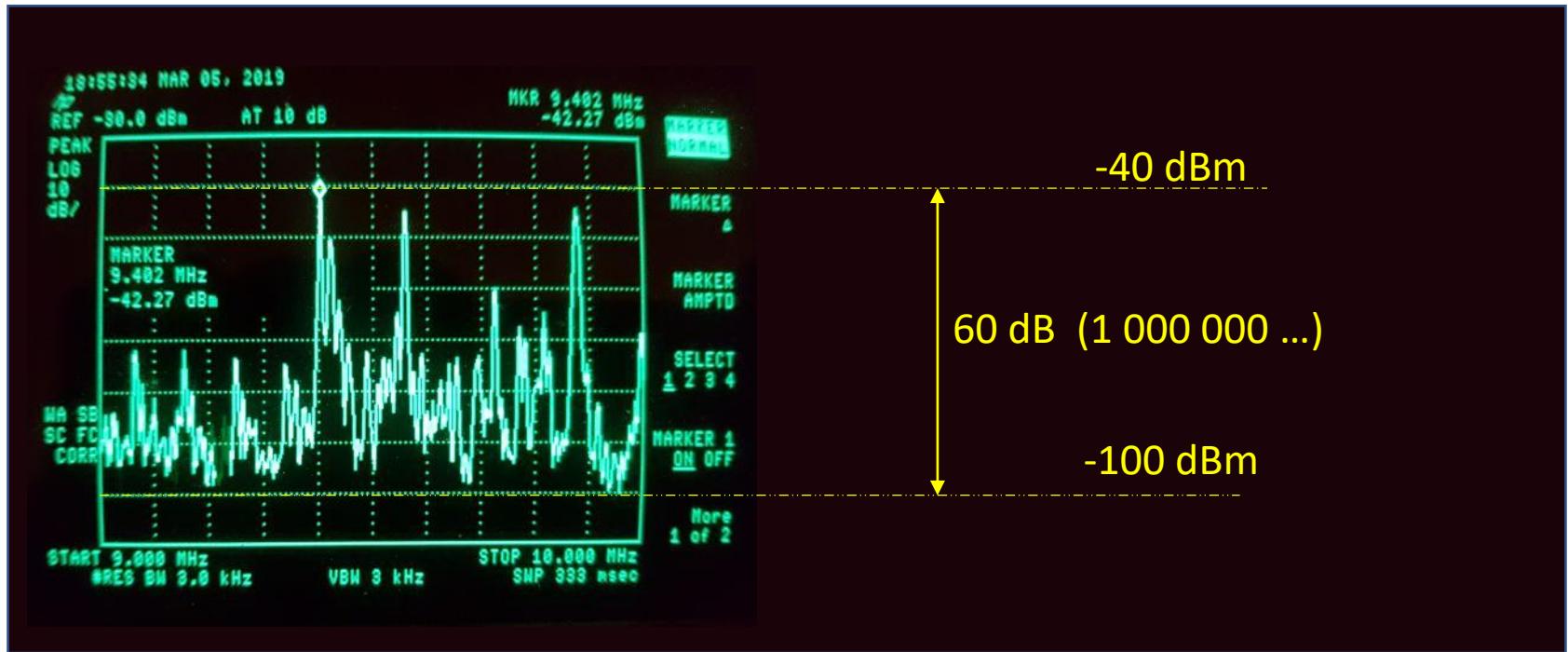


- Les signaux IQ permettent de connaître « le signe » de la fréquence (=dans quel sens ça tourne)

# La numérisation des signaux

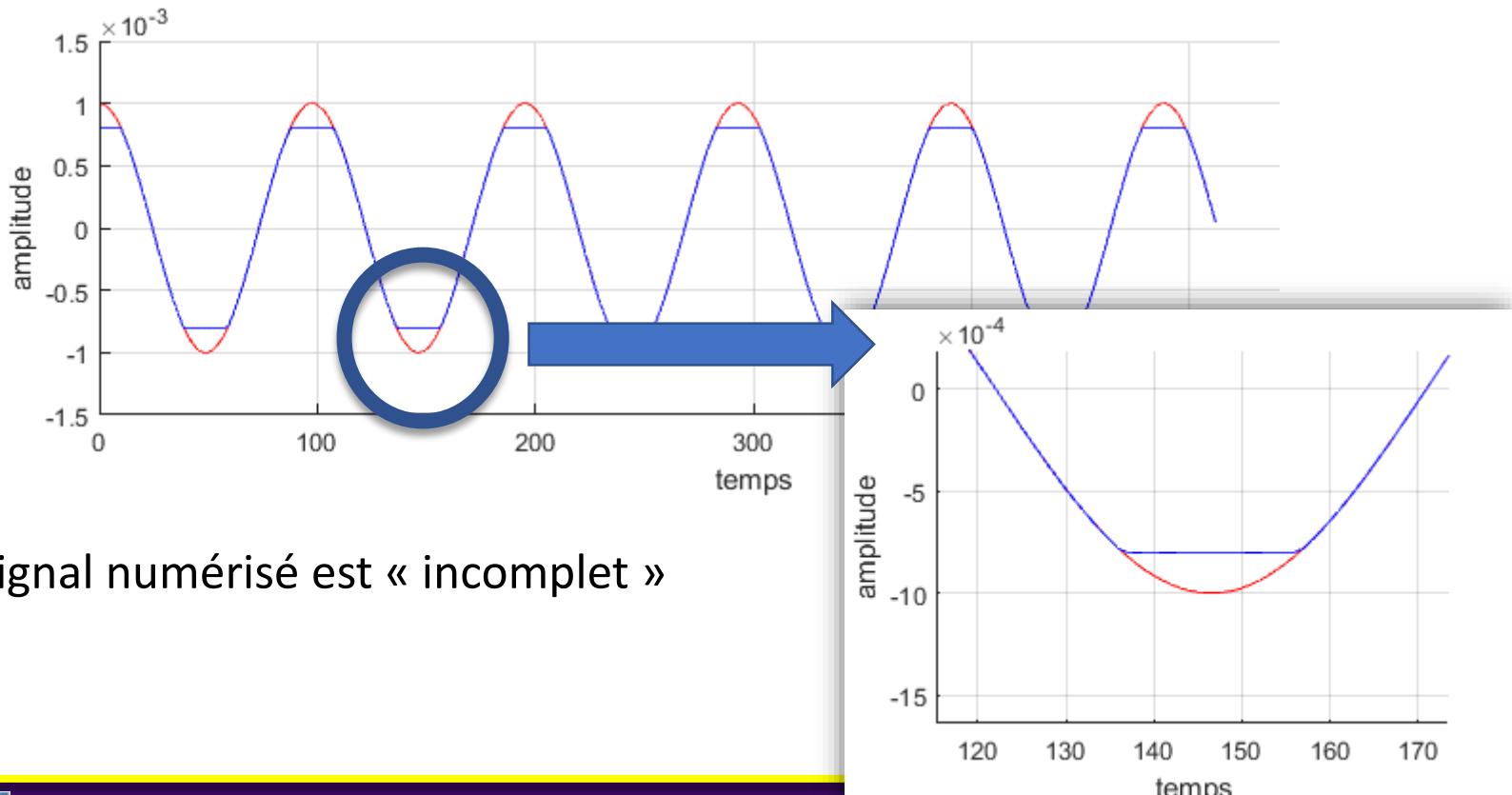
Ne pas se laisser éblouir par les forts pour voir les petits

# Mesure au « bout du coax »



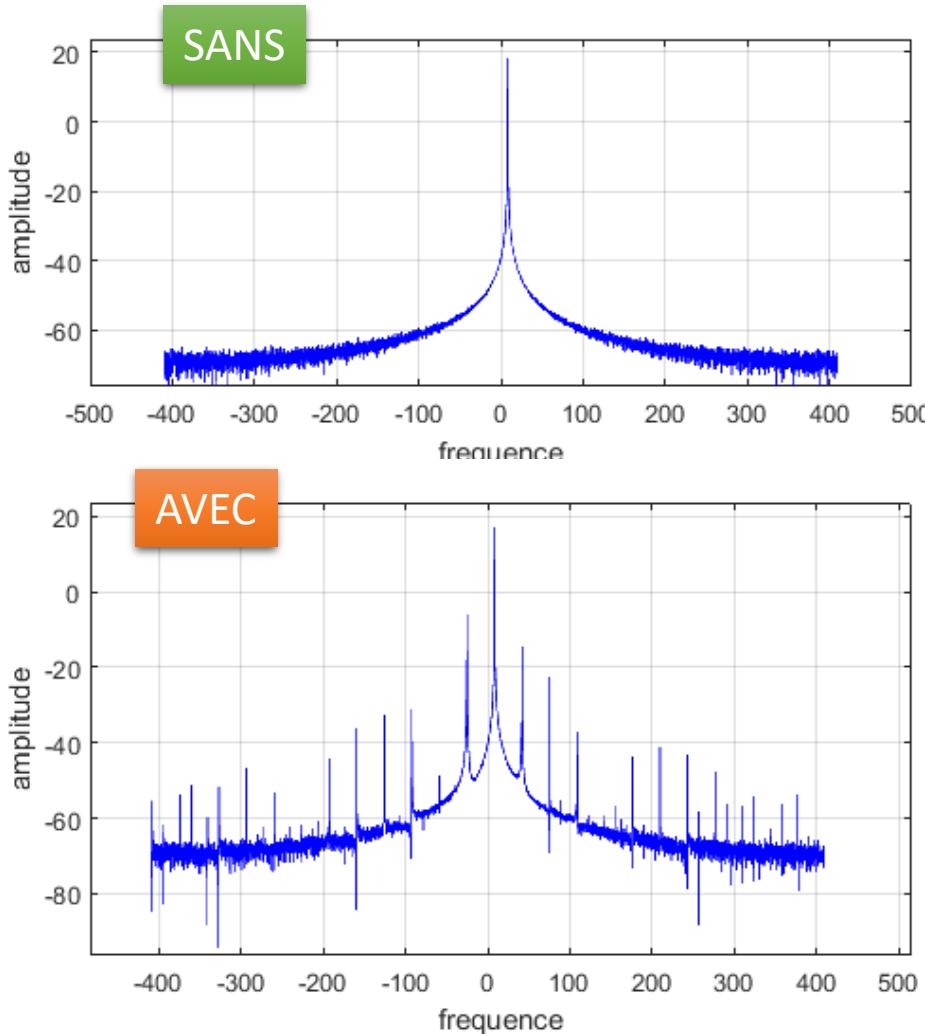
# Saturation à l'entrée

Gain trop important ou signal entrant trop fort avec saturation de l'amplificateur, ou dépassement  $V_{REF}$  du convertisseur ADC :



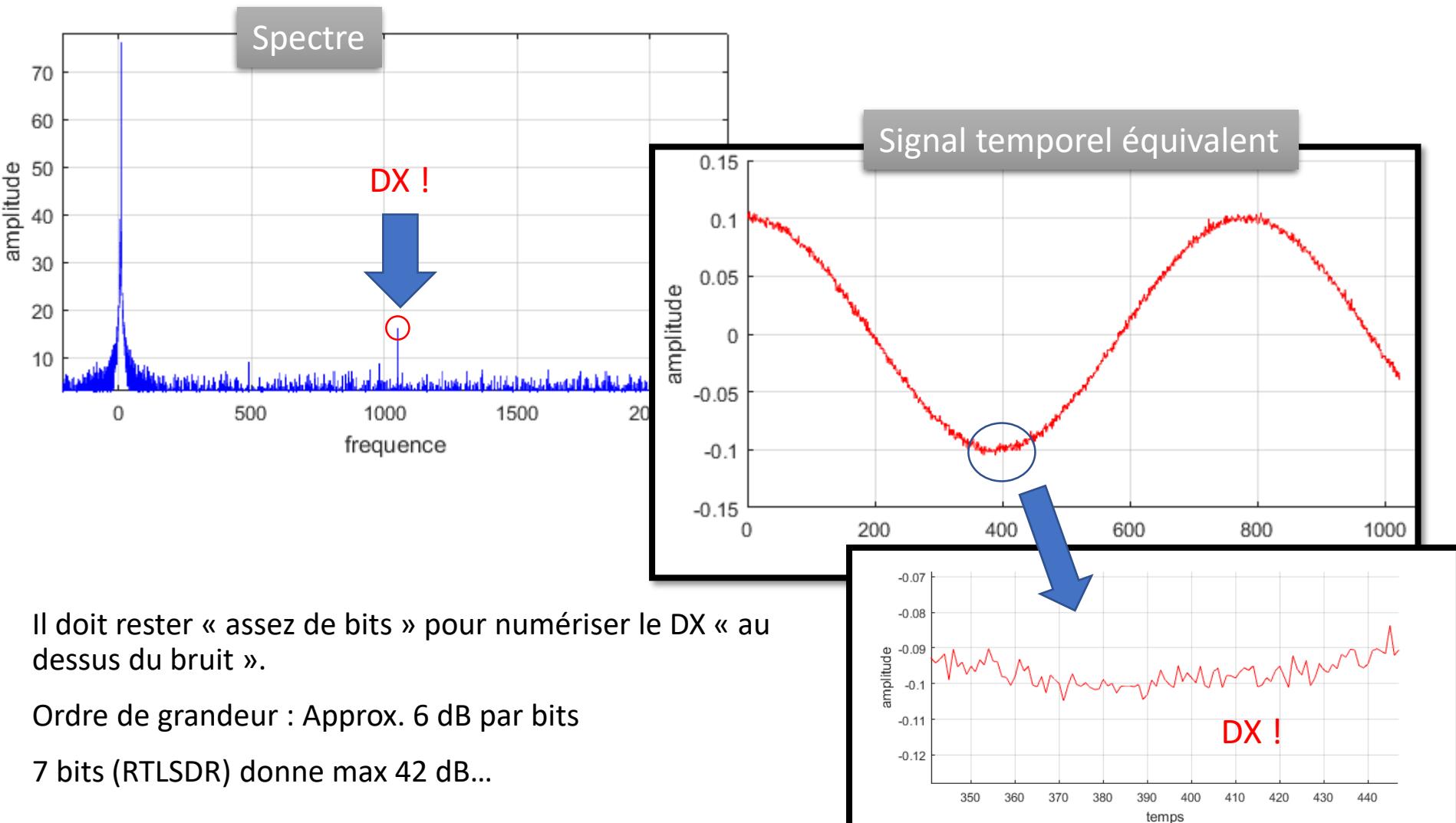
Le signal numérisé est « incomplet »

# Conséquences de la saturation

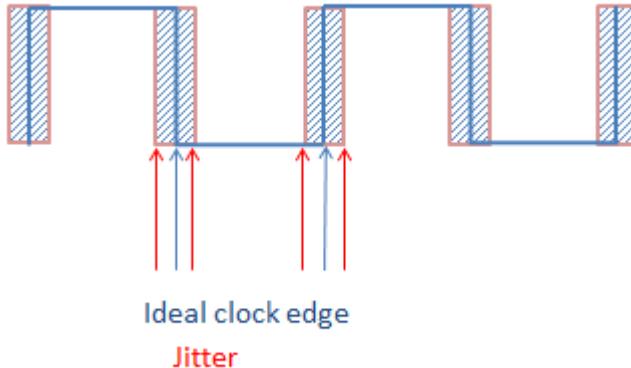


- Apparition de signaux « fantômes » sur le spectre,
- Apparition de porteuses audibles après démodulation

# Dynamique



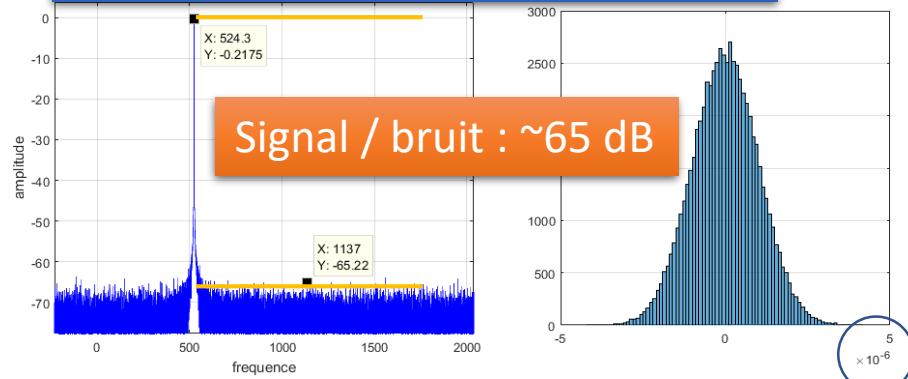
# Horloge de numérisation : *Jitter*



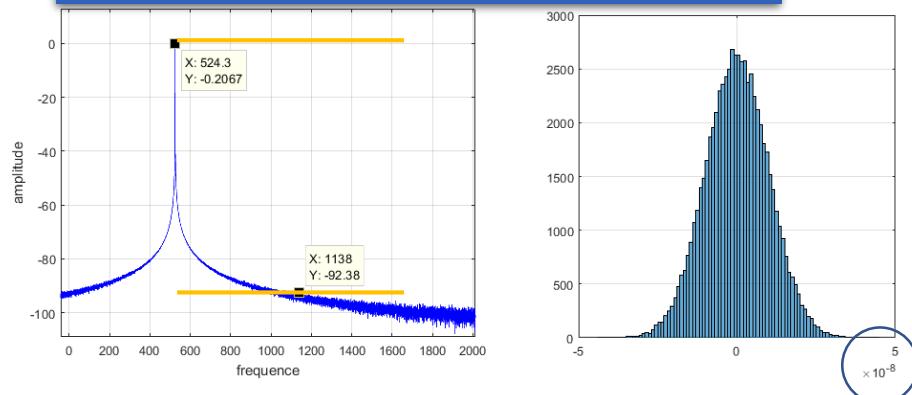
Le bruit d'horloge d'échantillonnage se traduira par une « prise de mesure » à un autre instant que celui qui était prévu.

Il vaut mieux une horloge stable qu'une horloge juste...

$\pm 5 \mu\text{s}$  d'erreur sur  $\text{Fclock}=100 \text{ KHz}$  :



$\pm 5 \text{ ns}$  d'erreur sur  $\text{Fclock}=100 \text{ KHz}$  :



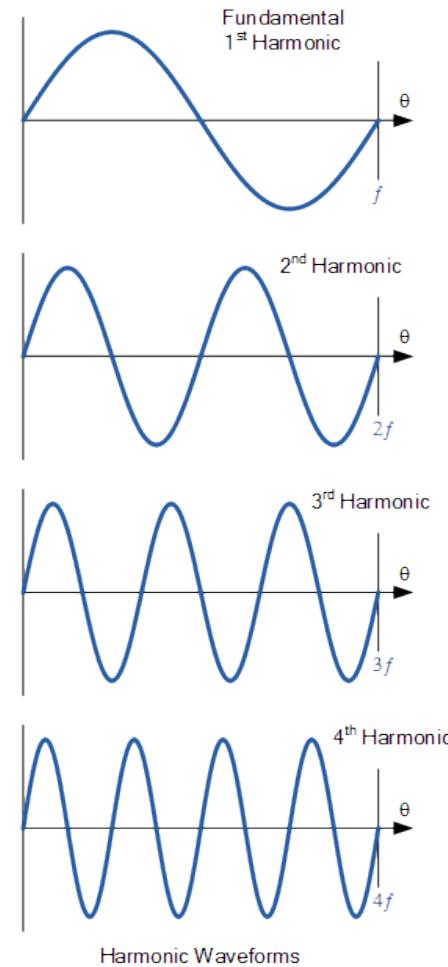


# Le spectre de Fourier

Temps, fréquence et vice-versa

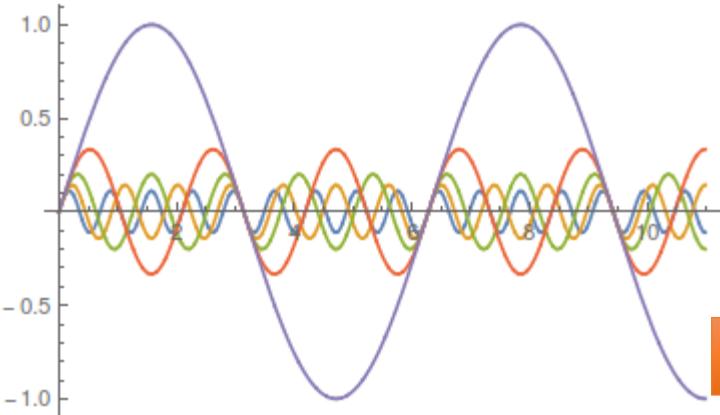
# Harmoniques

Une harmonique d'un signal  $s$  est un signal dont la fréquence est un multiple entier de la fréquence de  $s$

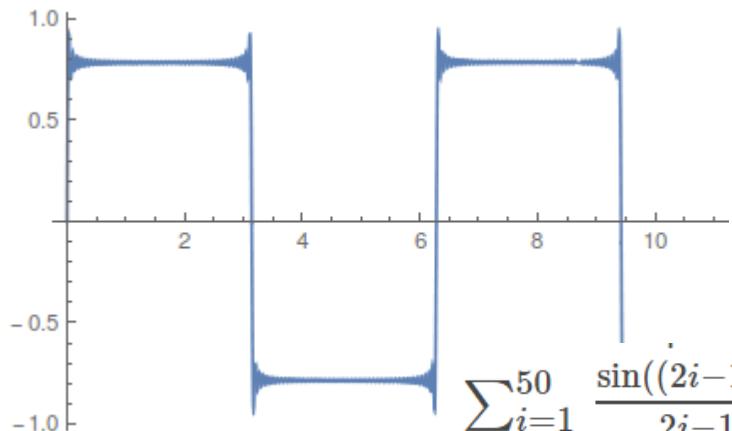
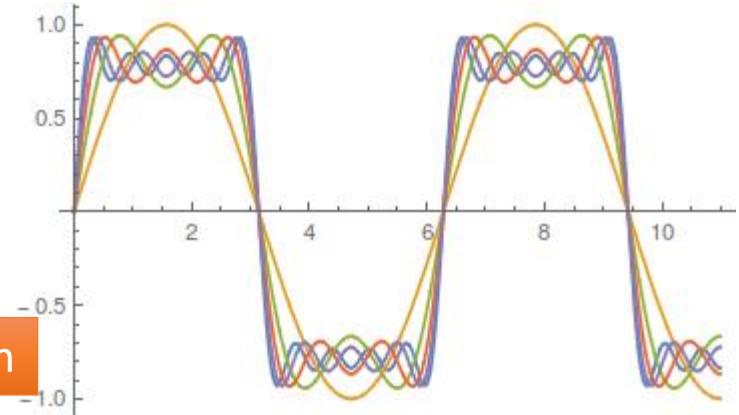


Harmonic Waveforms

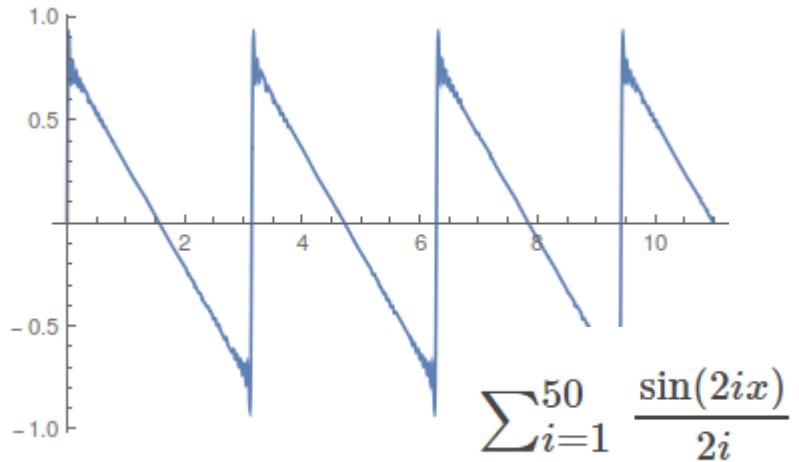
# Combinaison des harmoniques



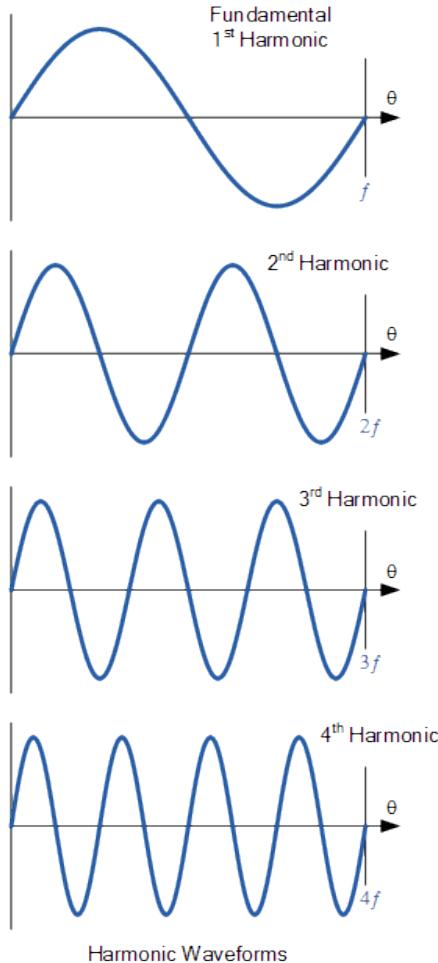
combinaison



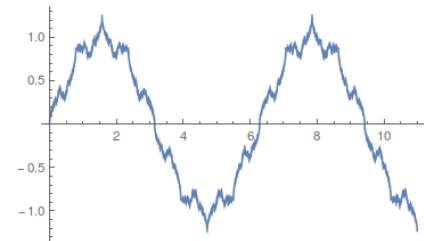
Source: <http://hawaiireedlab.com/wpress/?p=1693>



# Synthétiser un signal



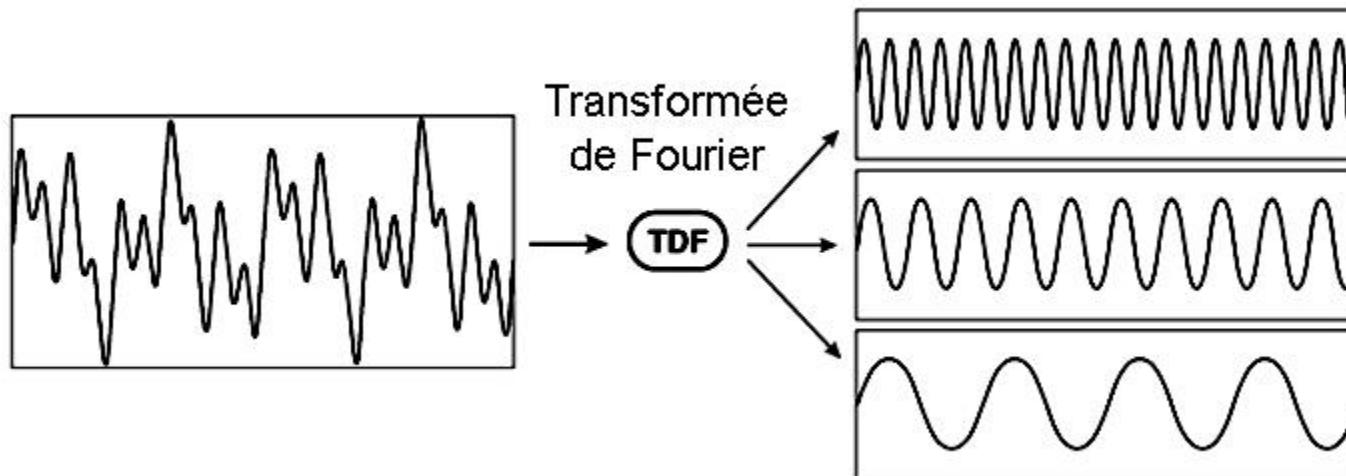
En combinant « comme il faut » des signaux sinusoïdaux\* harmoniques, on peut « synthétiser » d'autres signaux périodiques (mais pas des sinusoïdes)



\*Remarque : on peut aussi utiliser autre chose que des sinusoïdes...

# Décomposer un signal

A partir d'un signal d'entrée, retrouver les « ingrédients » qui ont servi à le générer  
C'est le processus inverse de ce qui a été vu précédemment



La transformée de Fourier (joseph) permet de faire cette décomposition

# Synthétiser/Décomposer

Représentation temporelle

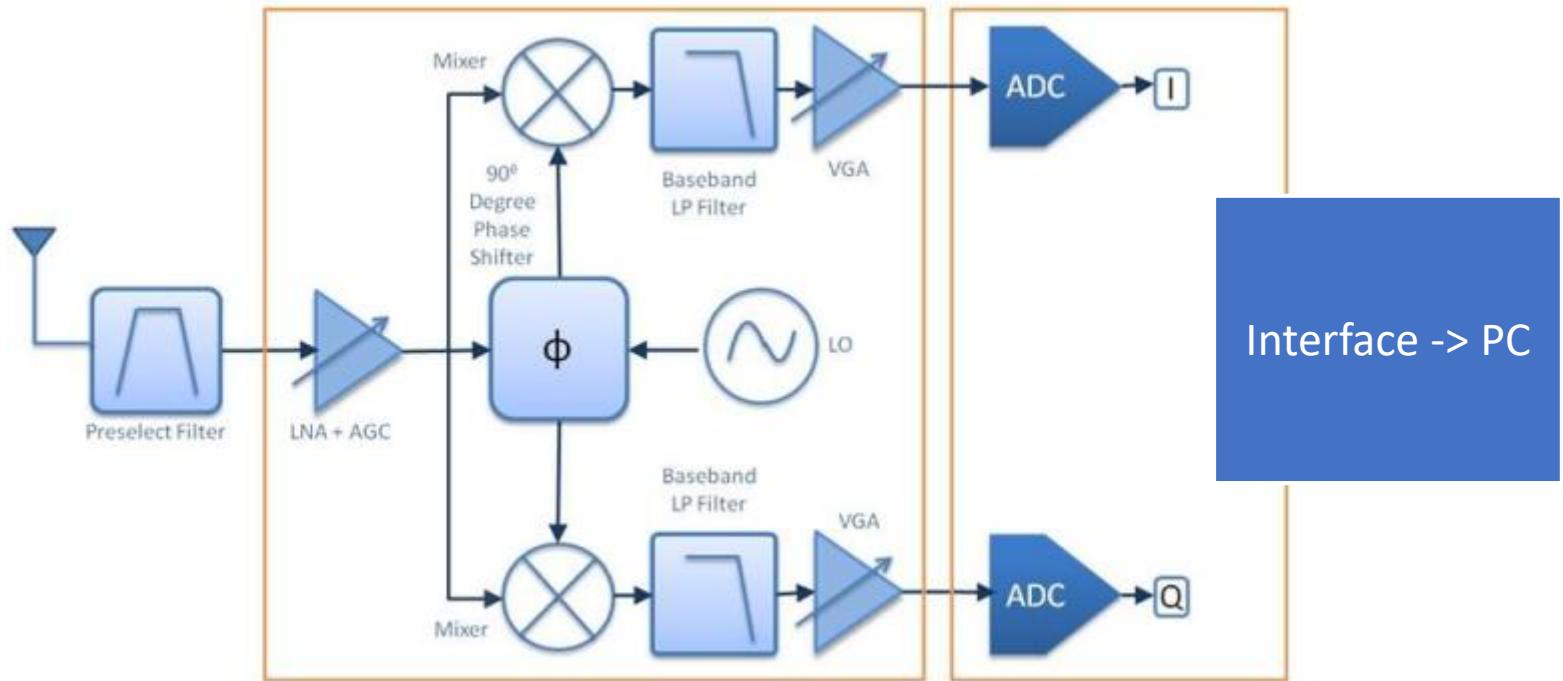
*Transformée de Fourier*

Représentation  
fréquentielle

# Architecture des SDR

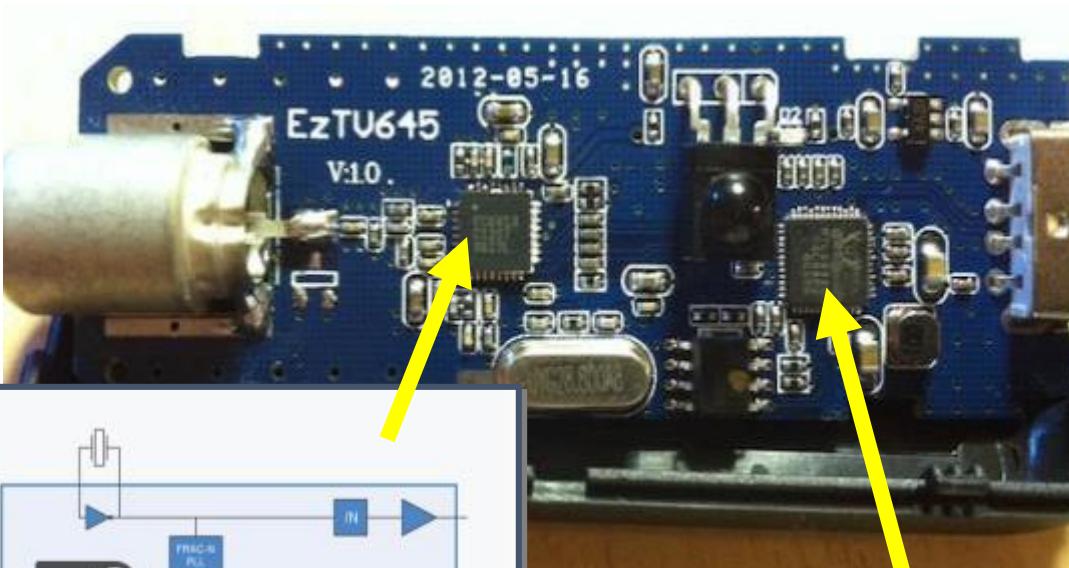
Finalement l'électronique sert quand même à quelque chose

# SDR à conversion directe

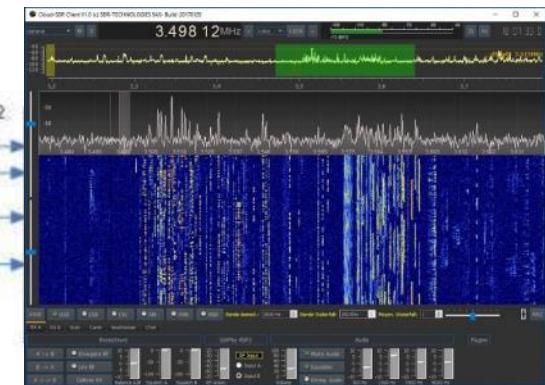
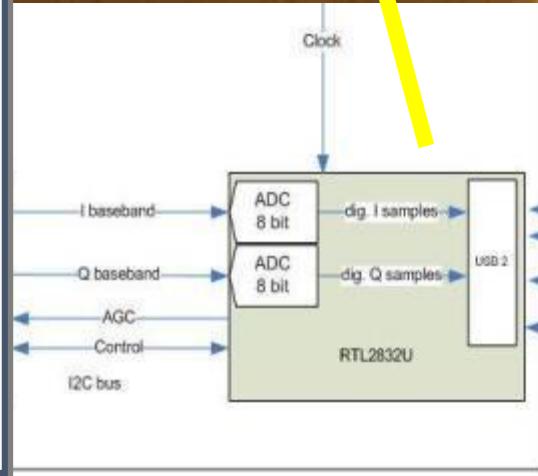
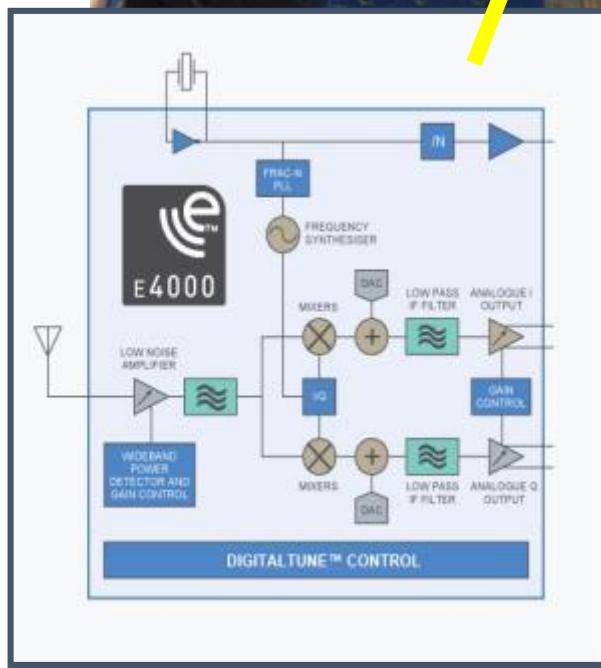


Un oscillateur local sélectionne la fréquence d'intérêt. Autour de cette fréquence, deux bandes sont « extraites » puis numérisées pour être envoyées vers l'ordinateur et le logiciel de traitement.

# Exemple : clé USB RTLSDR

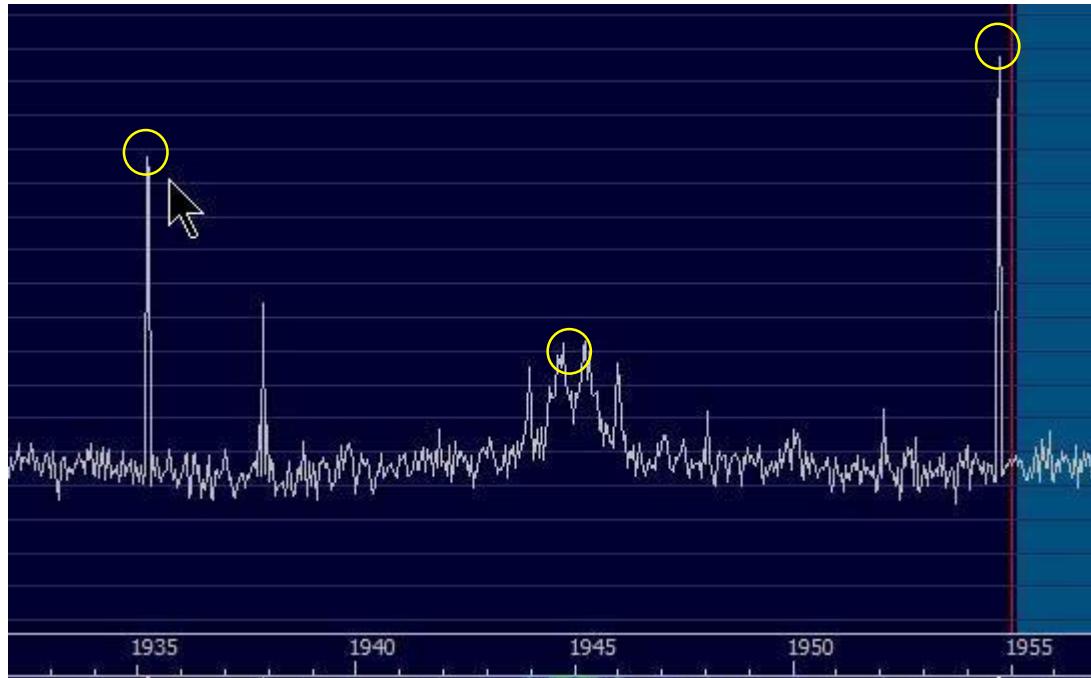


2 ADC « 8 » bits  
Lien USB  
μProcesseur intégré

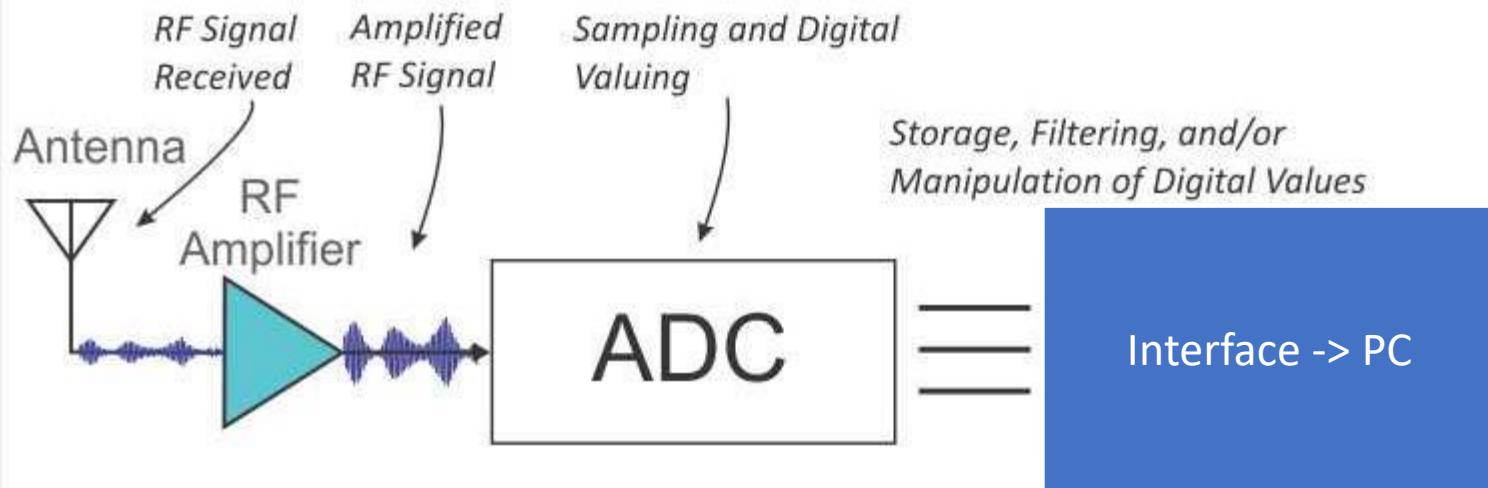


# Points faibles

- 2 voies = Peu probables qu'elles soient identiques...
- Fuite de l'oscillateur local

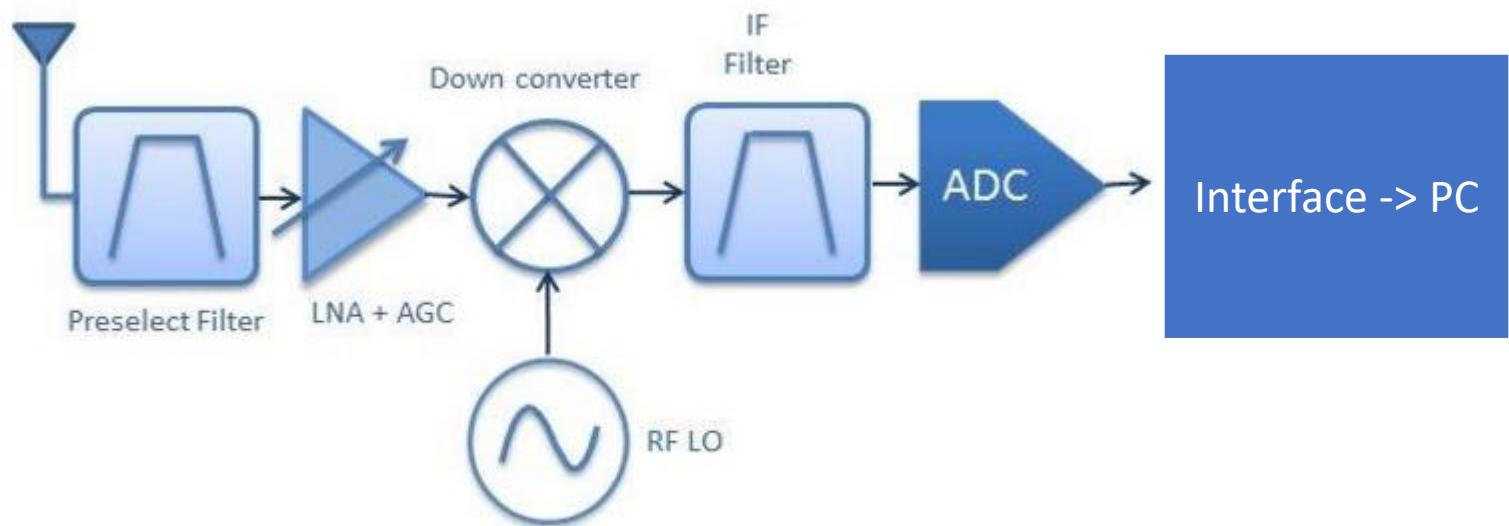


# SDR à échantillonnage direct



Après amplification le signal résultant est numérisé pour être envoyé vers l'ordinateur et le logiciel de traitement.

# SDR à échantillonnage direct



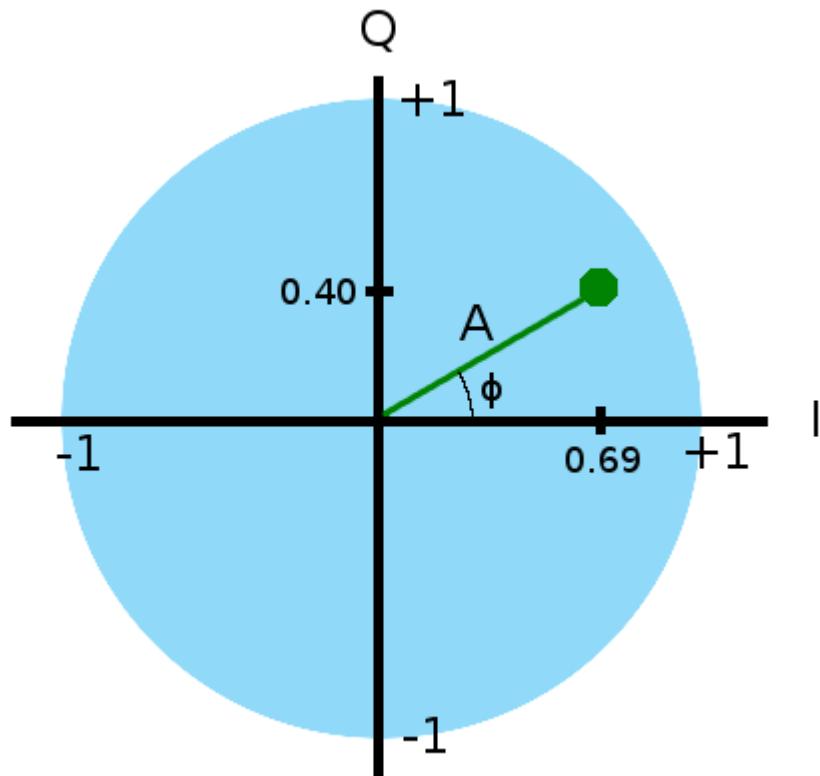
En pratique, bien souvent un oscillateur local transpose le signal reçu. Après filtrage, le signal résultant est numérisé.

# Logiciels « SDR »

Bon... et ça ressemble à quoi tout ça à la fin ?

# Les complexes qui simplifient

- Si on fait un peu de trigonométrie (Pythagore):



- Amplitude = longueur de A
- Angle  $\phi$

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

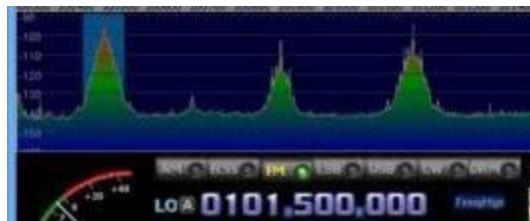
$$\phi = \text{atan} \left( \frac{I}{Q} \right)$$

# 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  $\phi$  à chaque instant et envoyer *la variation* à la sortie audio

# Démoduler = calculer

Flux de données IQ



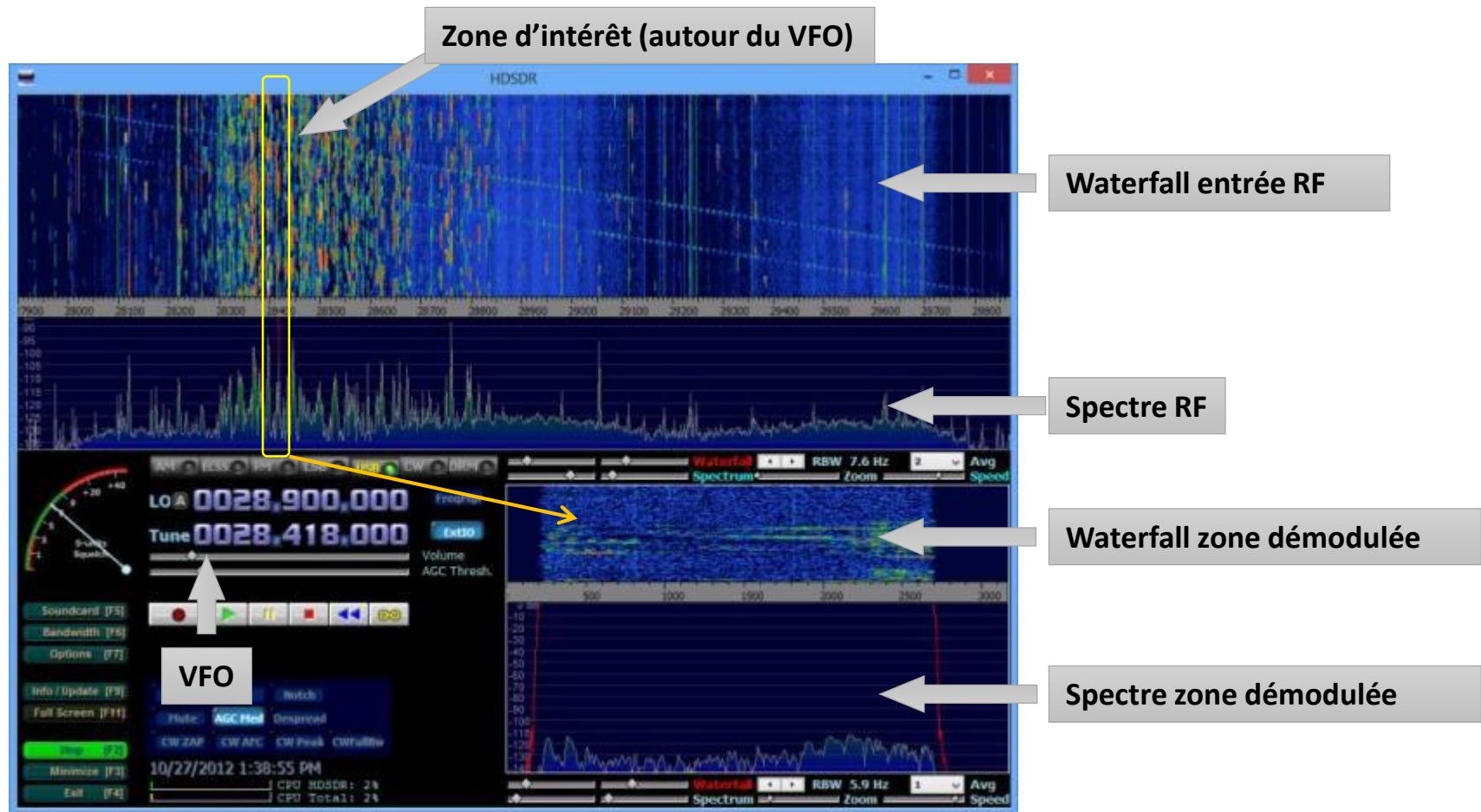
$$A = \sqrt{I^2 + Q^2}$$

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

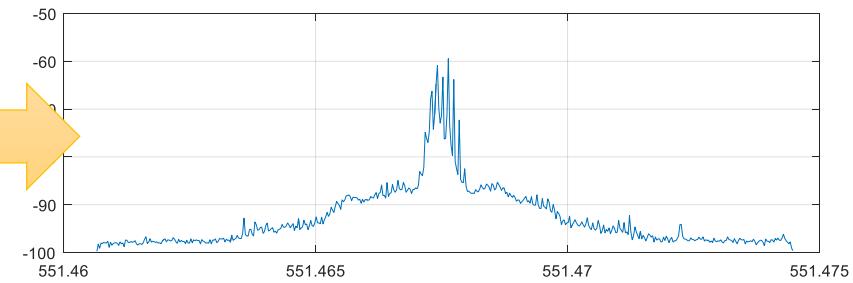
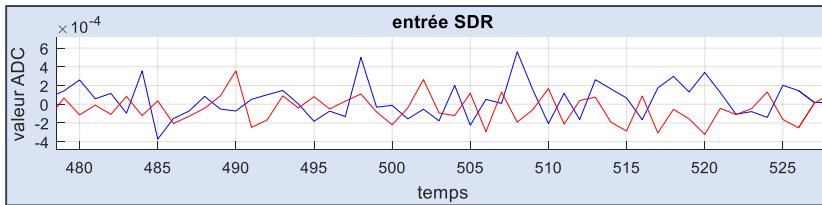
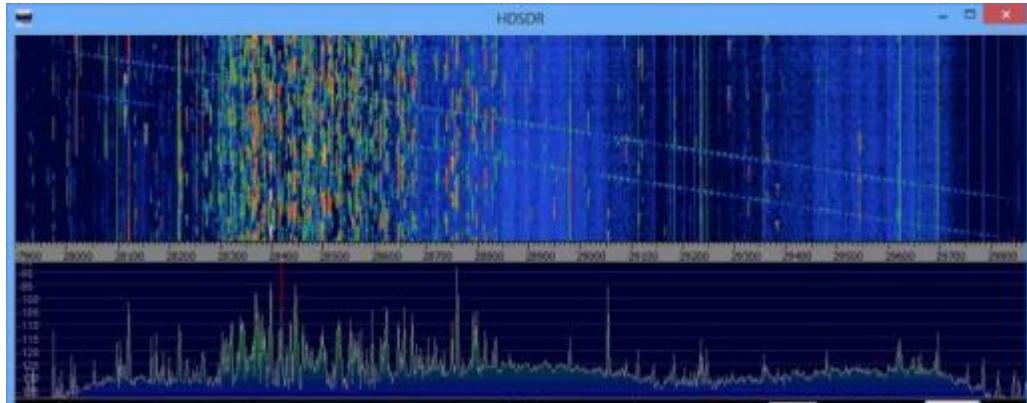
Audio

- 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

# Logiciel SDR : exemple HDSDR

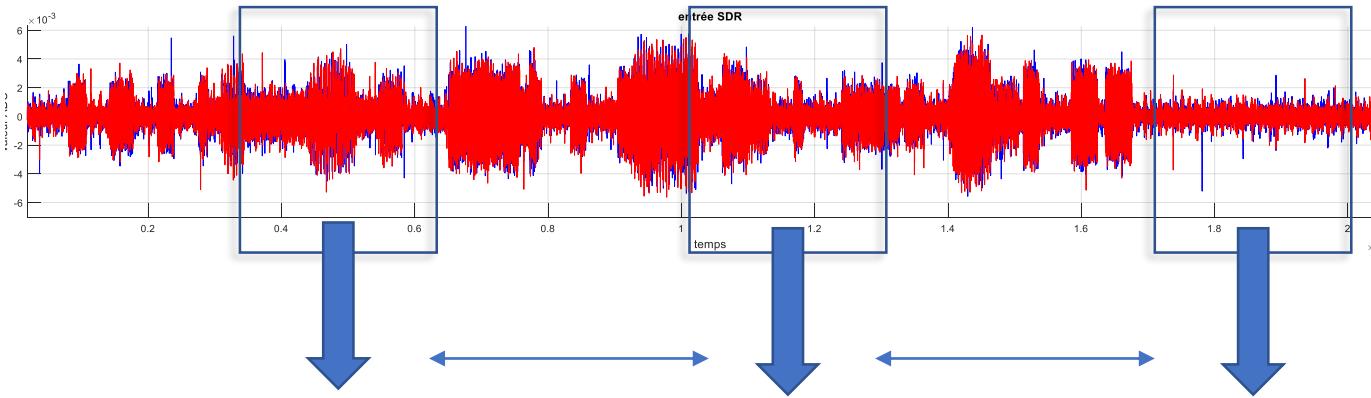


# Affichage du spectre



- Transformation « temporel » -> « fréquentiel » grâce à la transformée de Fourier

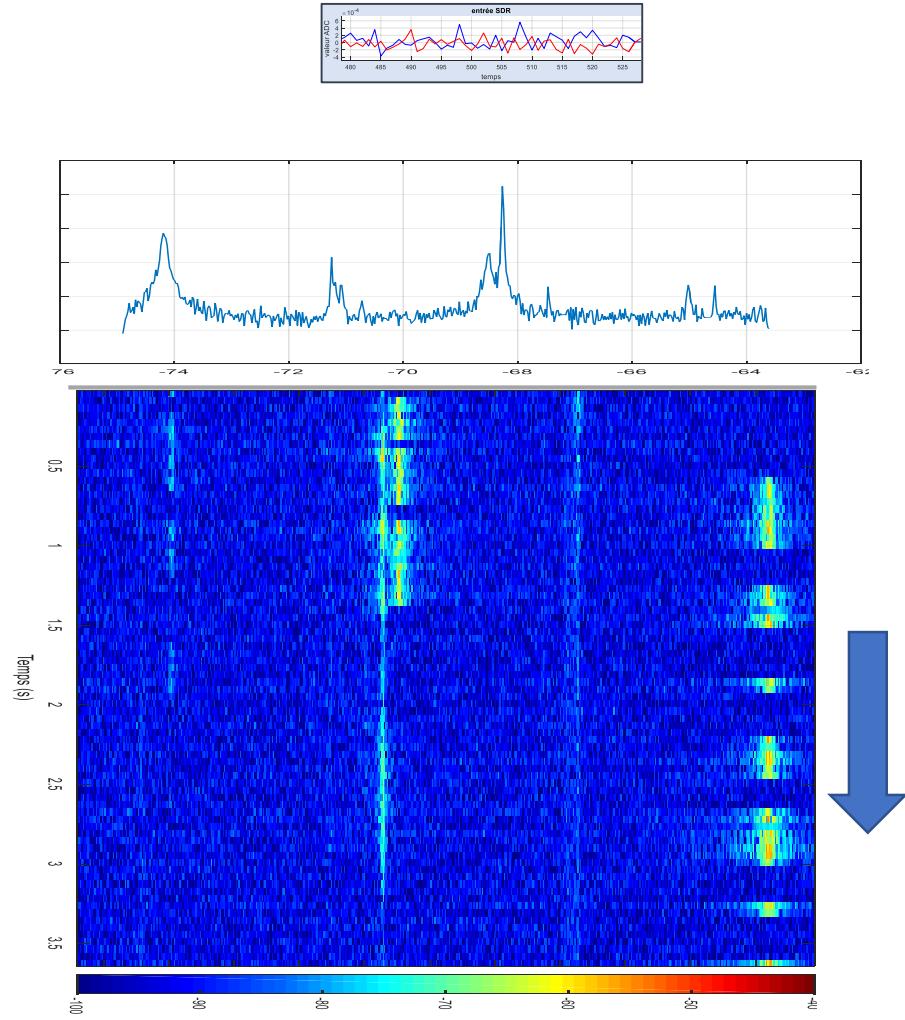
# Le waterfall et la FFT



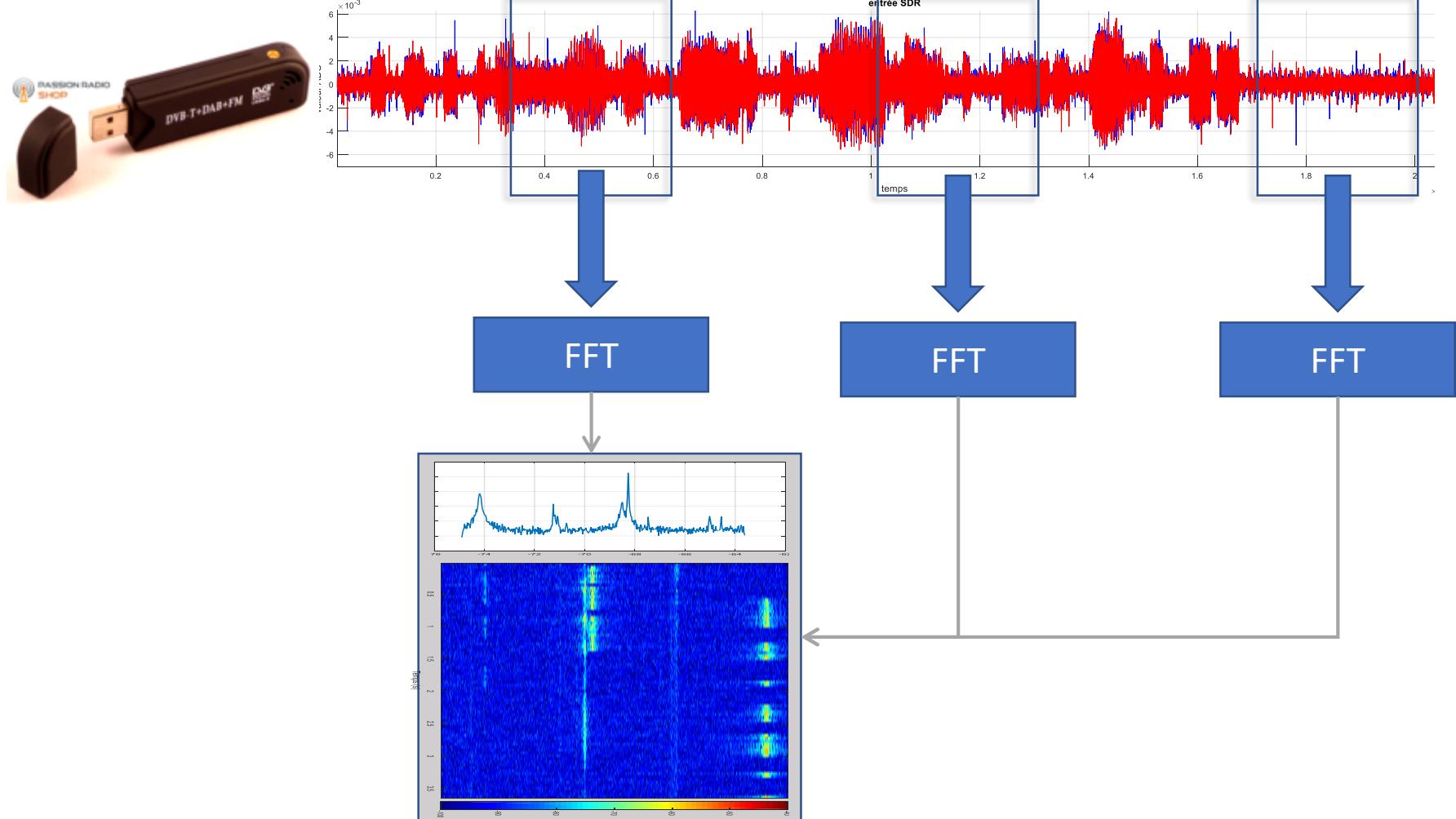
- Comme le calcul du spectre est assez gourmand en cycles CPU, on traite généralement des sous blocs
- La longueur du bloc détermine la résolution du waterfall

# Le Waterfall (chute d'eau)

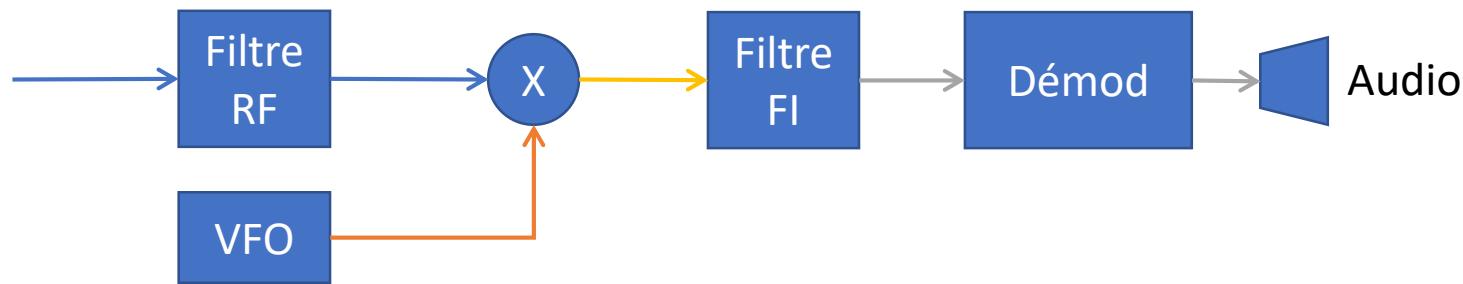
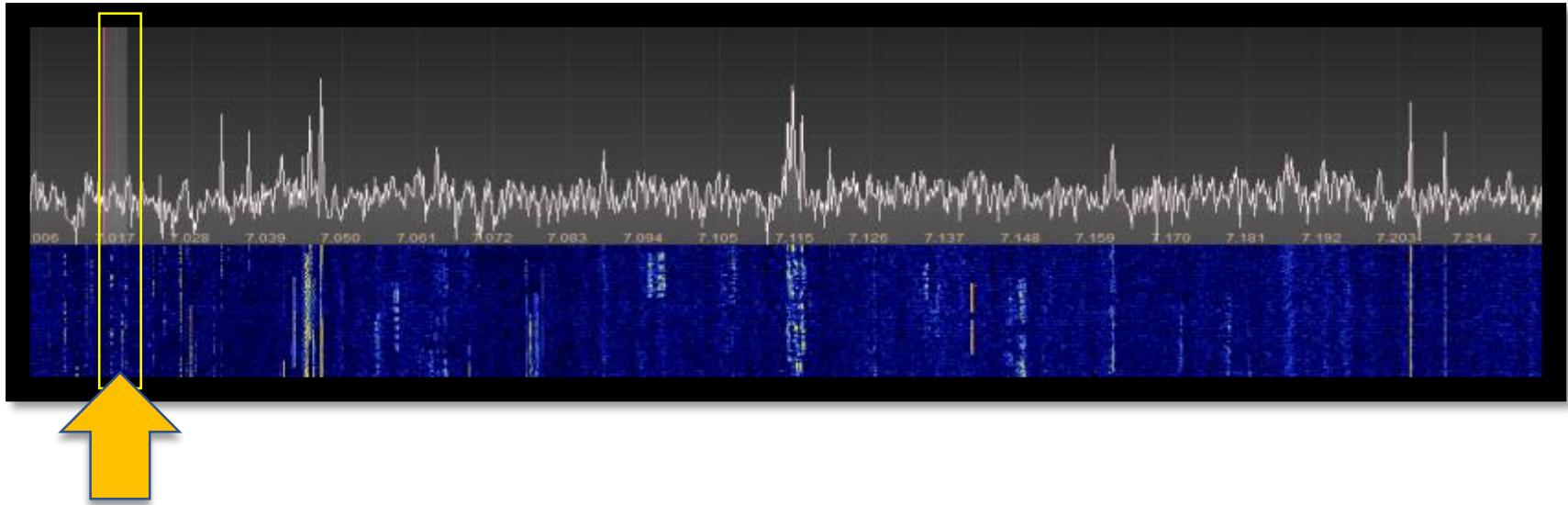
- Juste un empilage de spectre...
- On calcule des FFT en continu en prenant des « morceaux » du signal qui arrive du récepteur
- On peut aussi moyenner plusieurs FFT pour faire « sortir » des signaux faibles



# En résumé jusqu'à présent

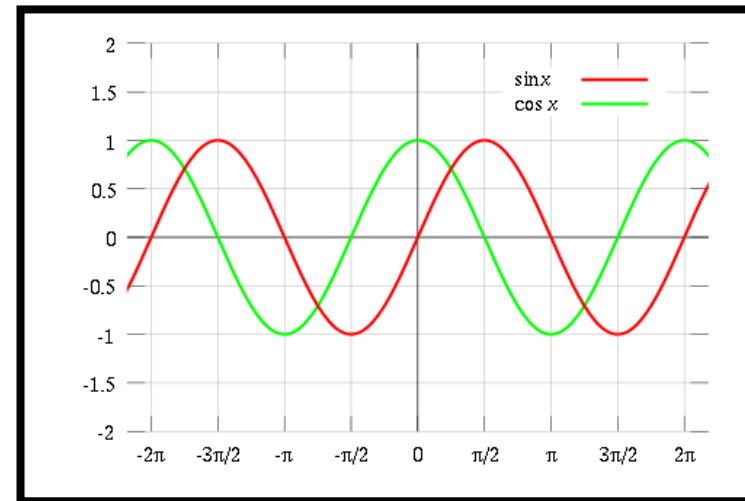


# Sélection de la bande utile

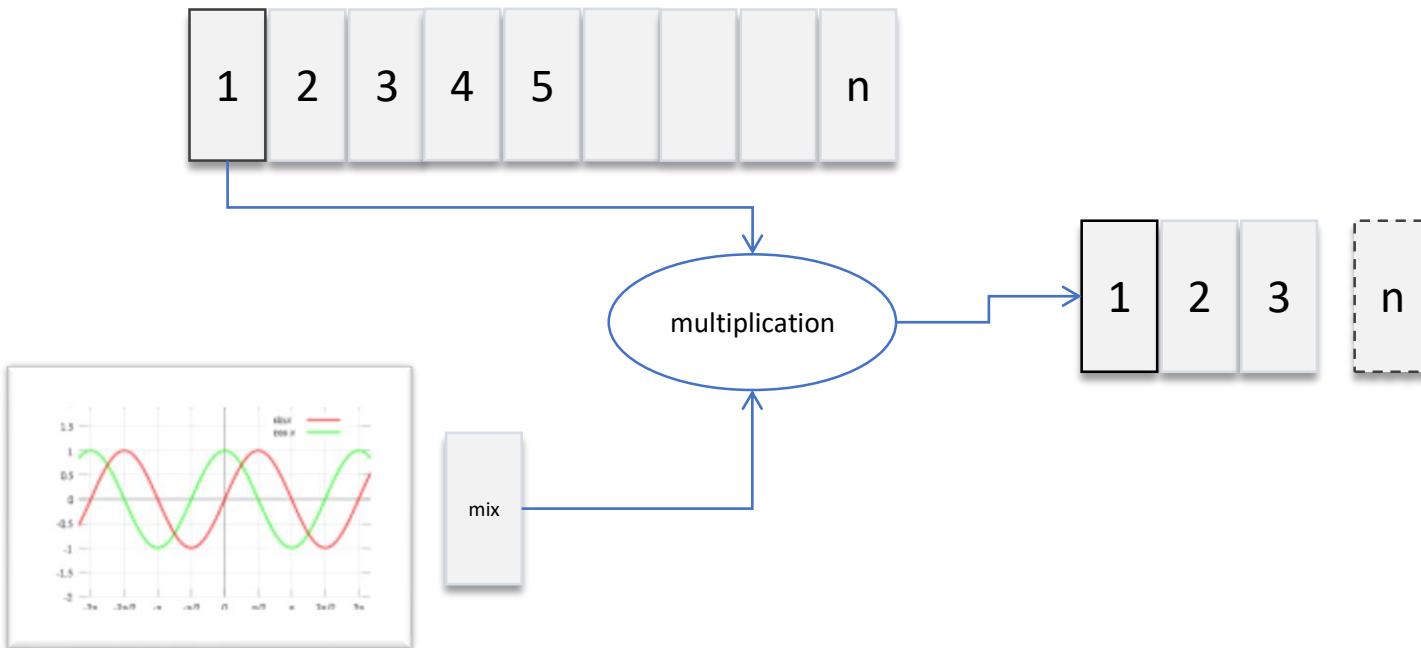


# Le VFO

- 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  $90^\circ$

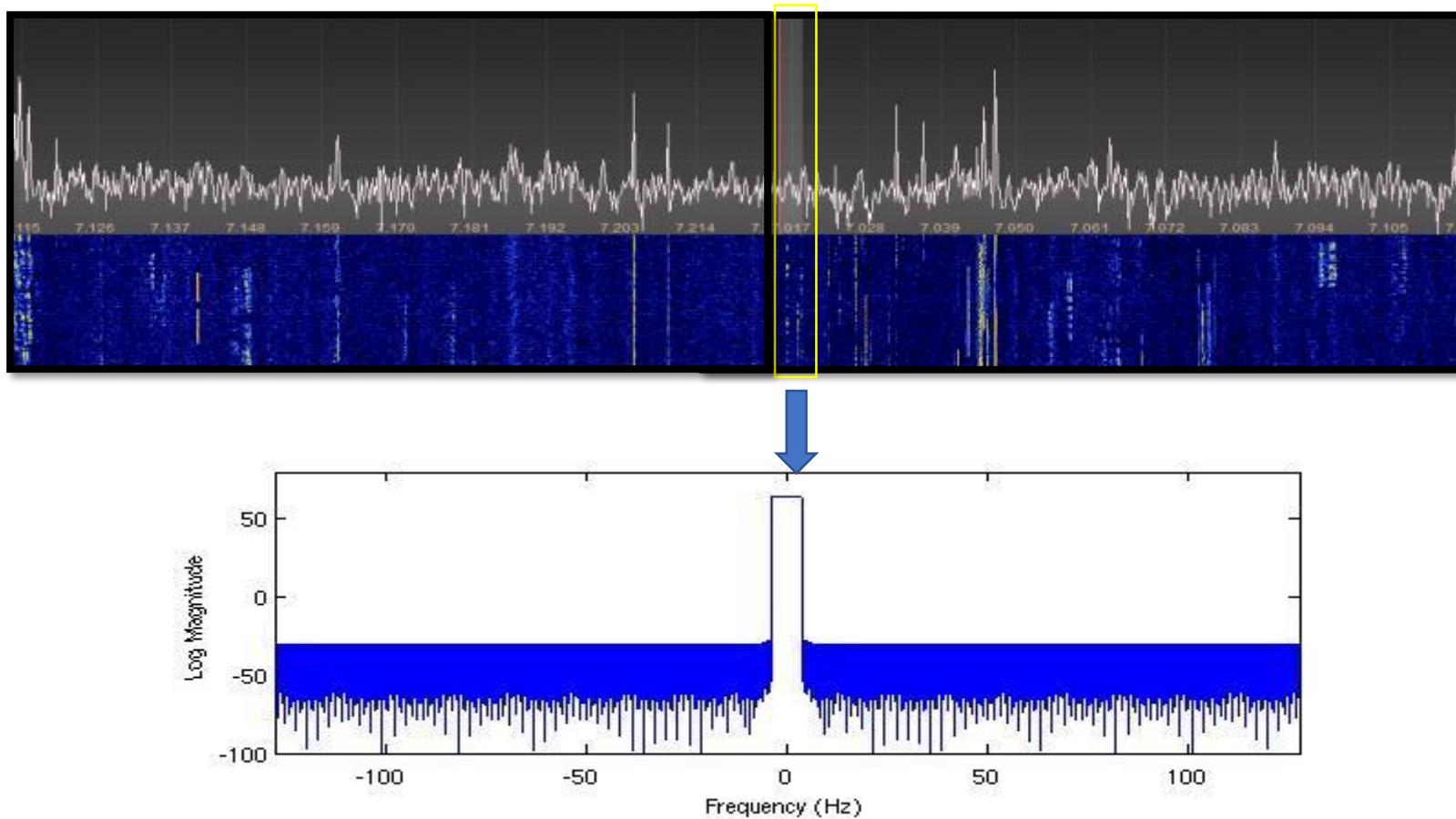


# Le mixage



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

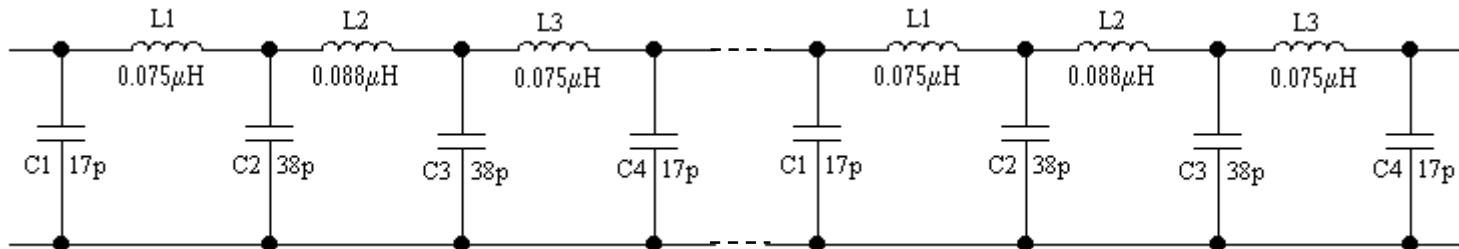
# Extraction de la bande « utile »



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

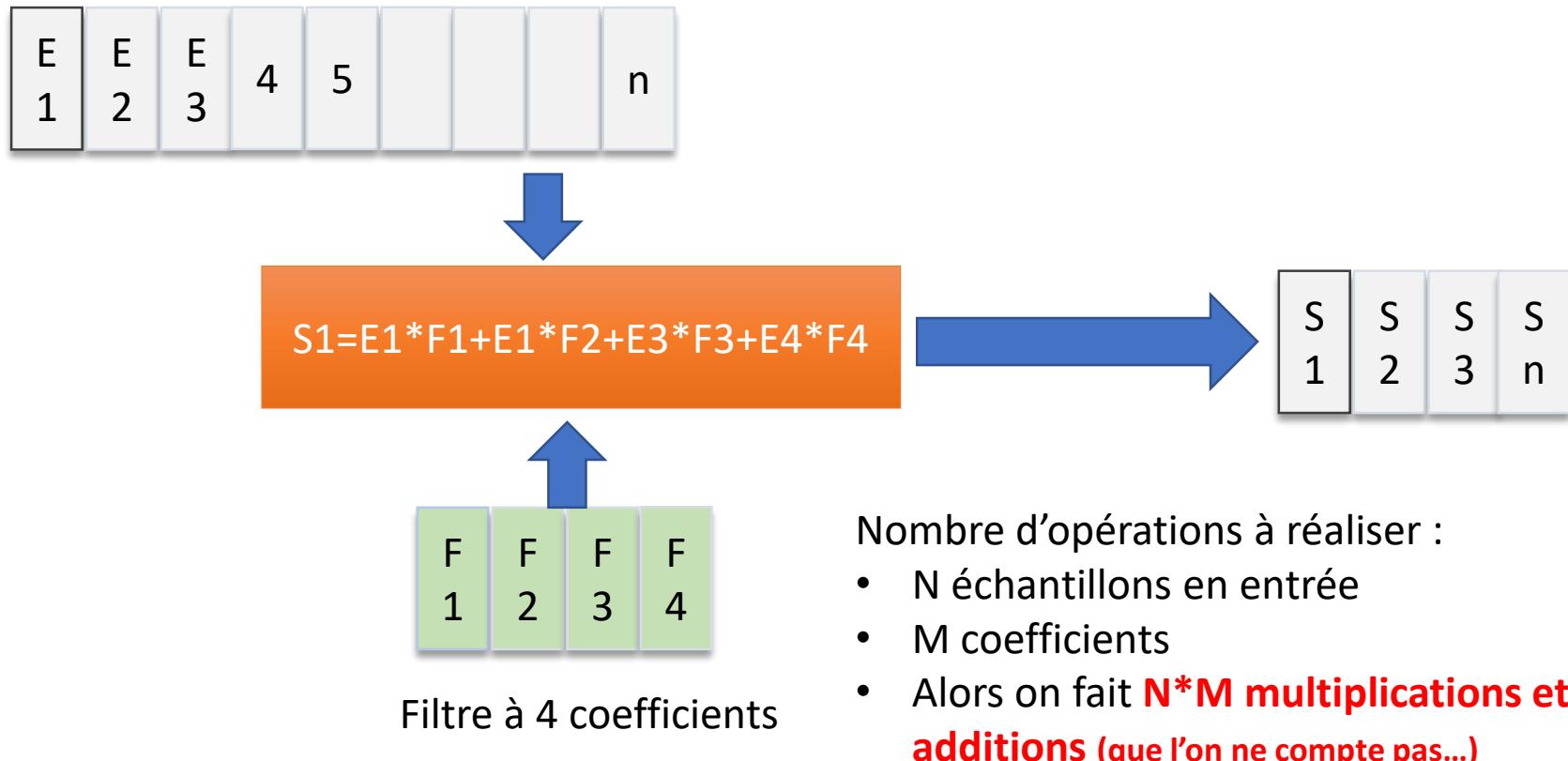
# Introduction au filtre FIR

- 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)



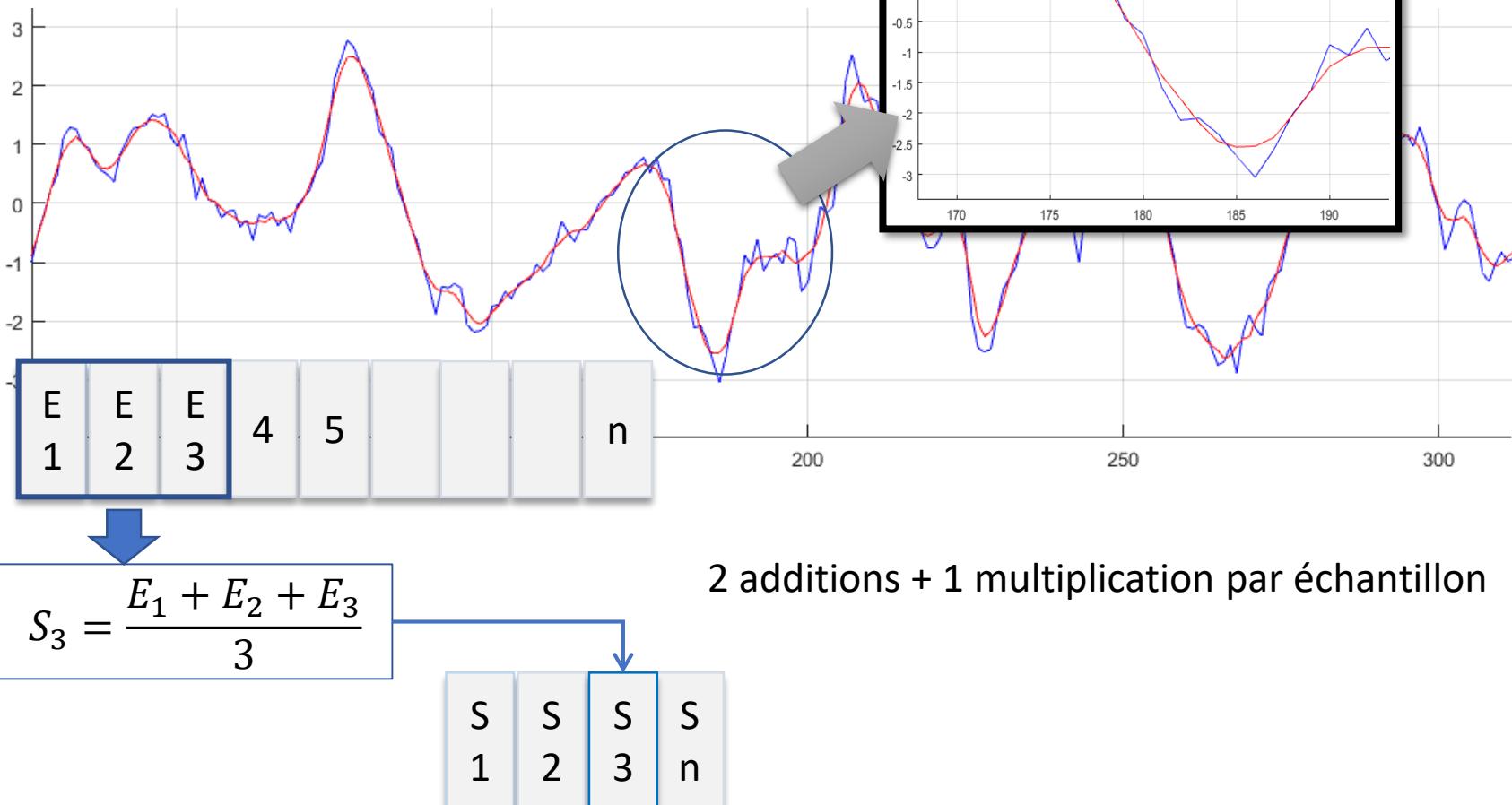
# Le filtrage « FIR »

- Opération répétée pour chaque échantillon



# Exemple de FIR

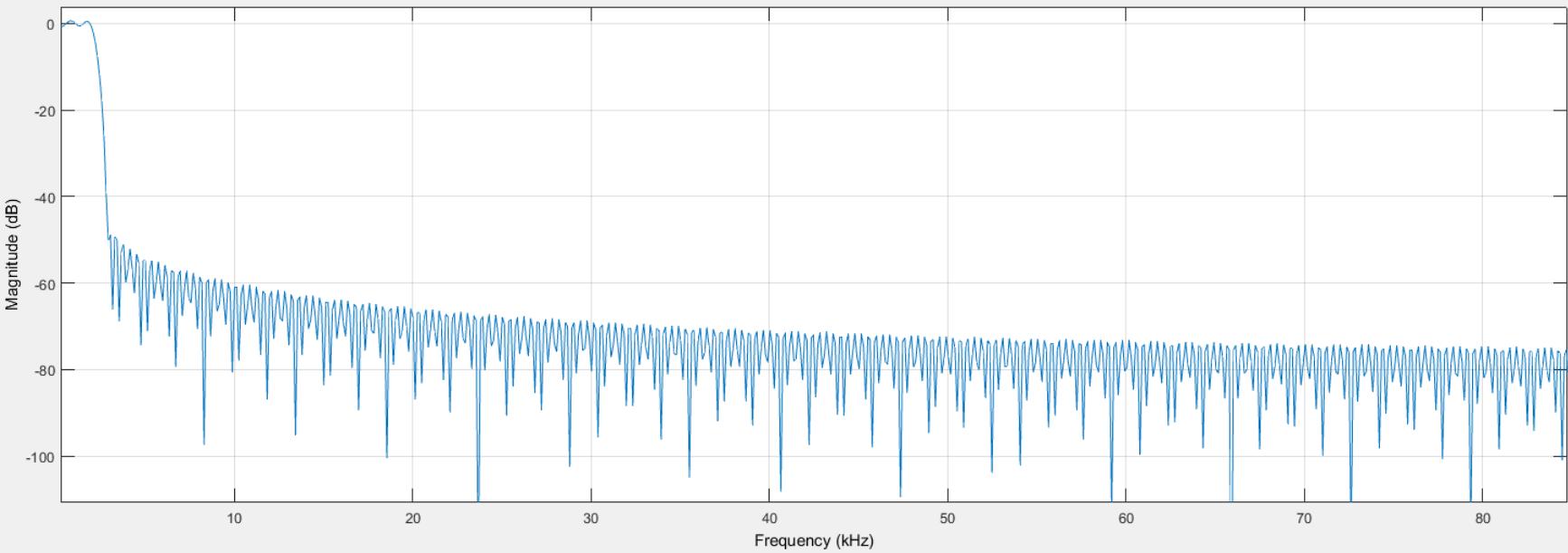
- La moyenne glissante :



2 additions + 1 multiplication par échantillon

# 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**

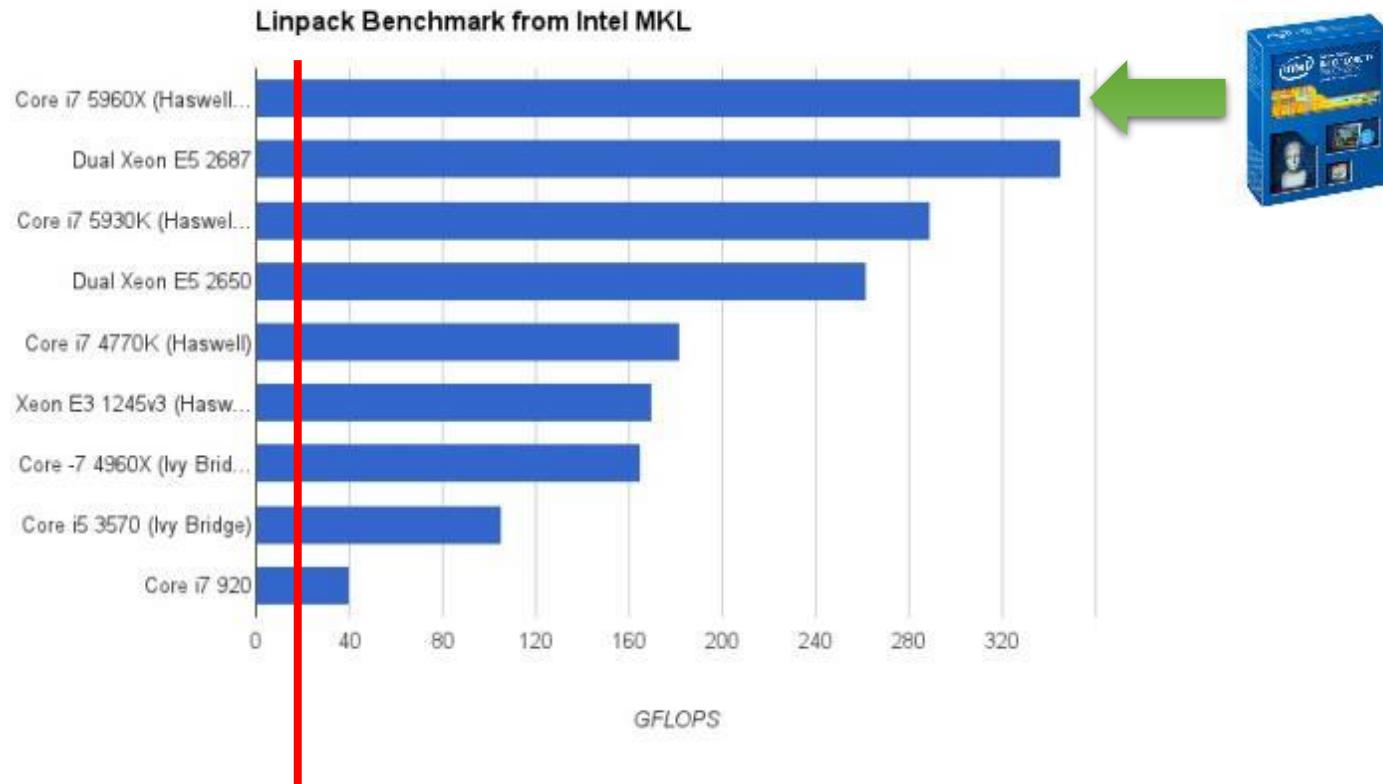
# 5060 coefficients, et alors ?

- 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...

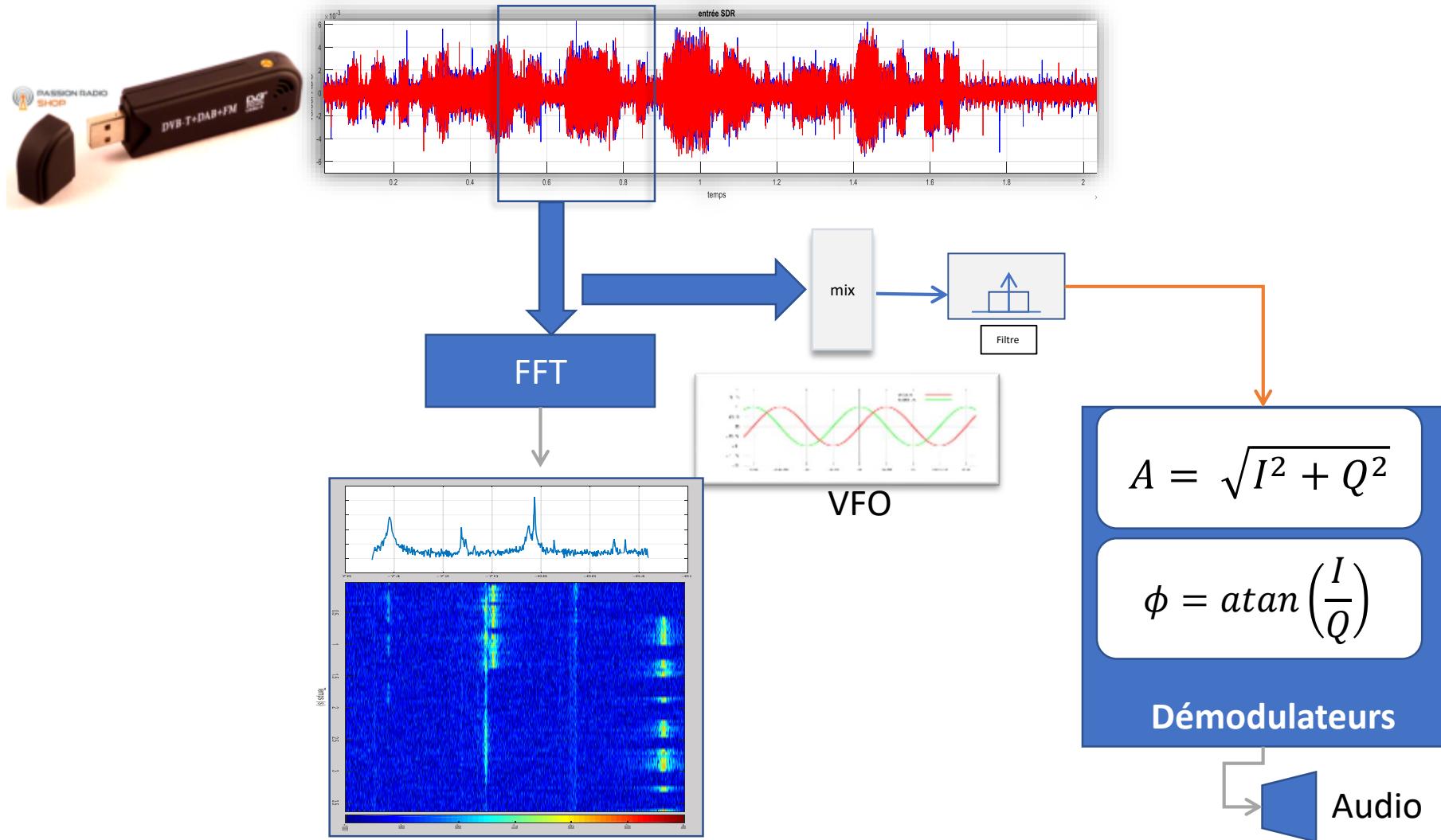
# 20 Giga Flops pour filtrer ?

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



Conclusion... faut pas faire comme ça !

# En résumé jusqu'à présent...



# Points à retenir

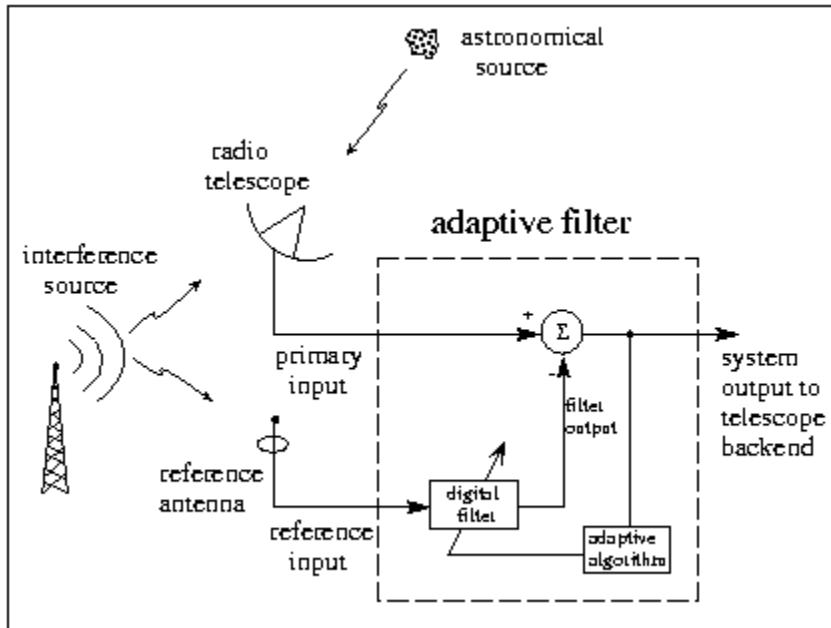
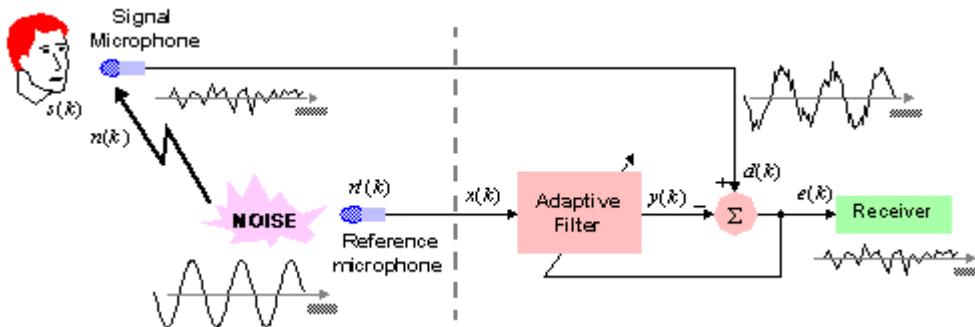
- La puissance de calcul nécessaire est directement proportionnelle à la largeur bande numérisée par le SDR
- Le bus USB est un point limitant: l'USB2 ne permet pas de passer des bandes très larges
- La « haute résolution » du waterfall est très consommatrice de ressources de calcul
- **Ne pas espérer des 10aines de MHz en temps réel sur une Raspberry PI...**

# Quel avenir pour la SDR ?

Ça existe déjà chez les « pro », on l'a quand ?

# Filtrage adaptatif

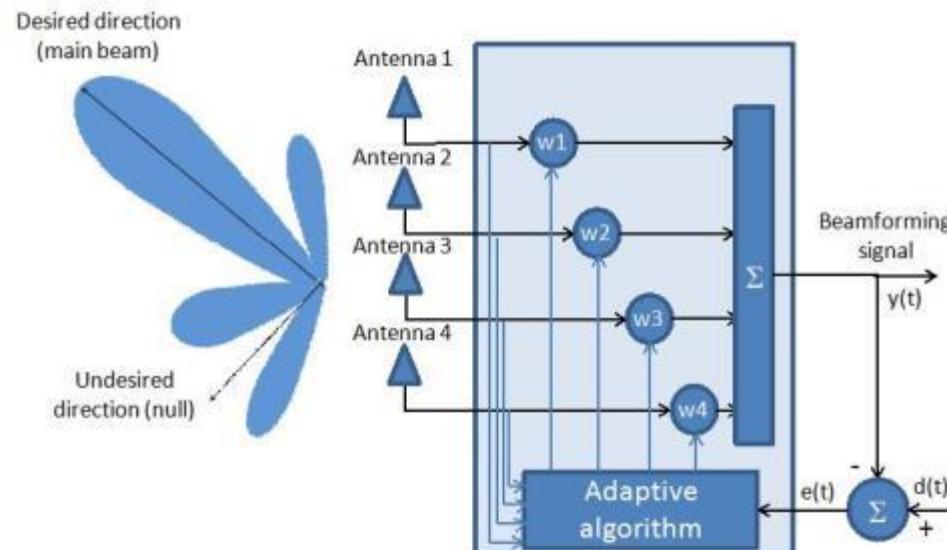
- Exemple en audio : suppression de bruit ou d'écho



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

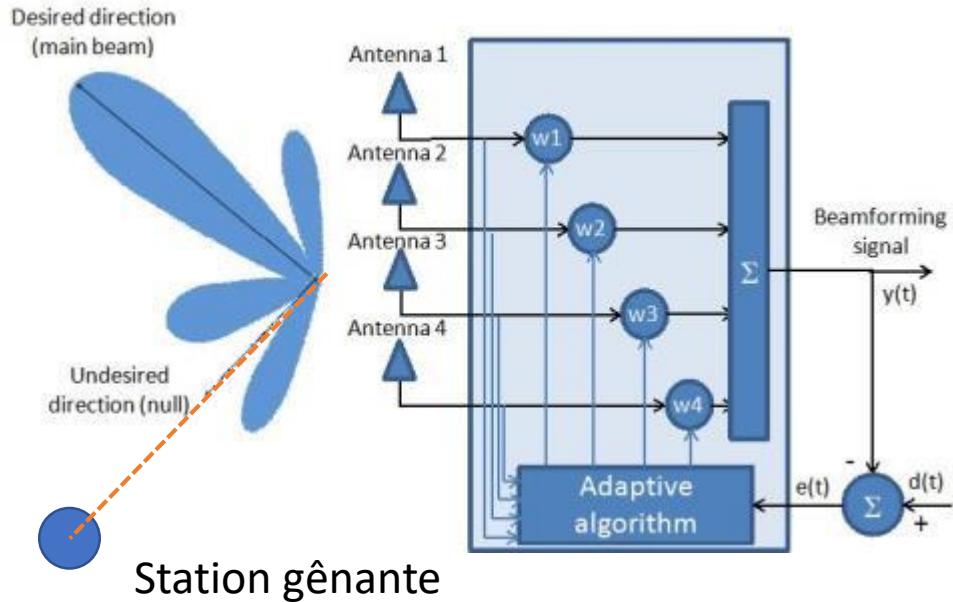
# FFC

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



- On maximise la sensibilité dans la direction souhaitée

# Antibrouillage



- Ici on minimise la sensibilité dans la direction du « brouilleur »
- Avec  $N$  antennes,  $N-1$  « nuls » (2 voies = 1 nul)

# Formation à la SDR

Objectifs : Réaliser son propre SDR (logiciel ET matériel)

Le concept à l'étude :

- 5 jours, pendant les vacances scolaires, dans des hackerspaces
- Jour 1 : rappel maths et python pour ceux qui veulent
- Jours suivants :
  - Matin : présentations / cours
  - Après-midi : labo logiciel et matériel
  - Soirées : labo logiciel et matériel suite... pour ceux qui veulent
- Formation par une équipe de bénévoles
- Supports de cours « open source »
- Participation aux frais des repas – formation gratuite et ouverte à tous
- Hébergement non compris

**Vous voulez participer ? Des commentaires ?**  
→ Contact : **f4gkr@f4gkr.org**