

AVANTIX

Avancement Simulateur IQ

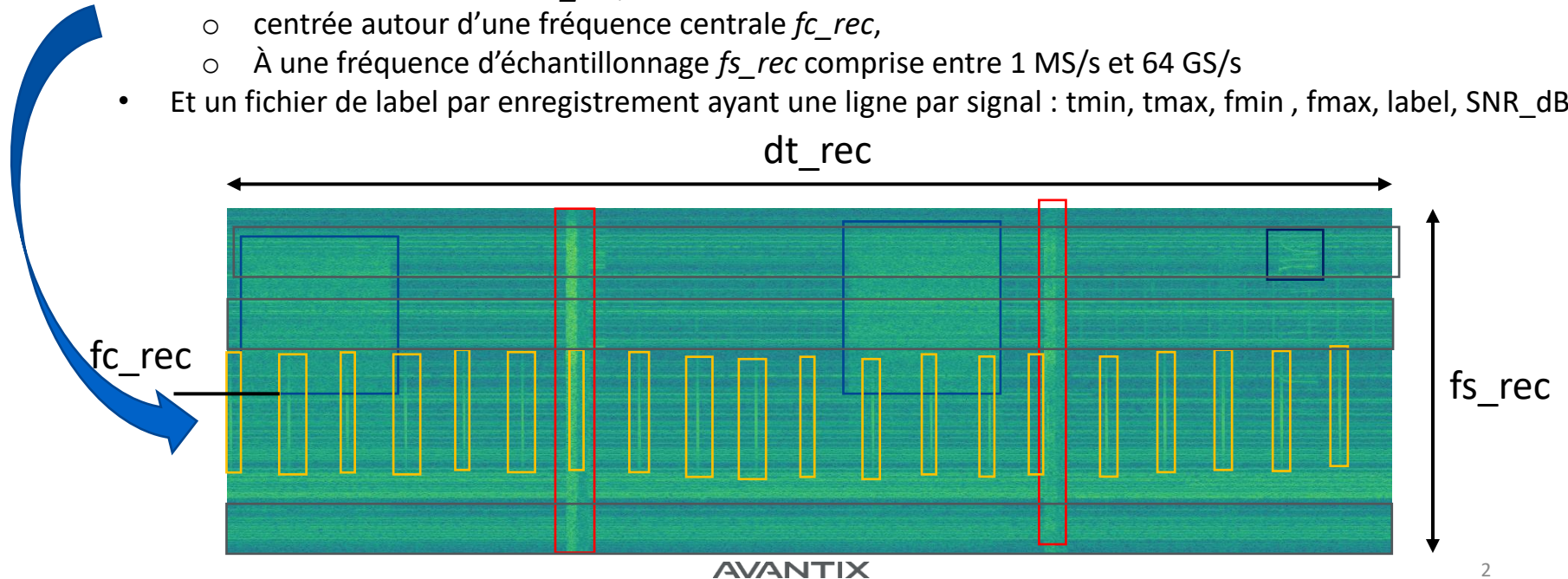
10/01/2024

© Avantix – for internal use

Besoin

Avoir un programme qui génère en sortie :

- un enregistrement IQ
 - d'une certaine durée dt_rec ,
 - centrée autour d'une fréquence centrale fc_rec ,
 - À une fréquence d'échantillonnage fs_rec comprise entre 1 MS/s et 64 GS/s
- Et un fichier de label par enregistrement ayant une ligne par signal : $tmin$, $tmax$, $fmin$, $fmax$, $label$, SNR_dB



Simulateur

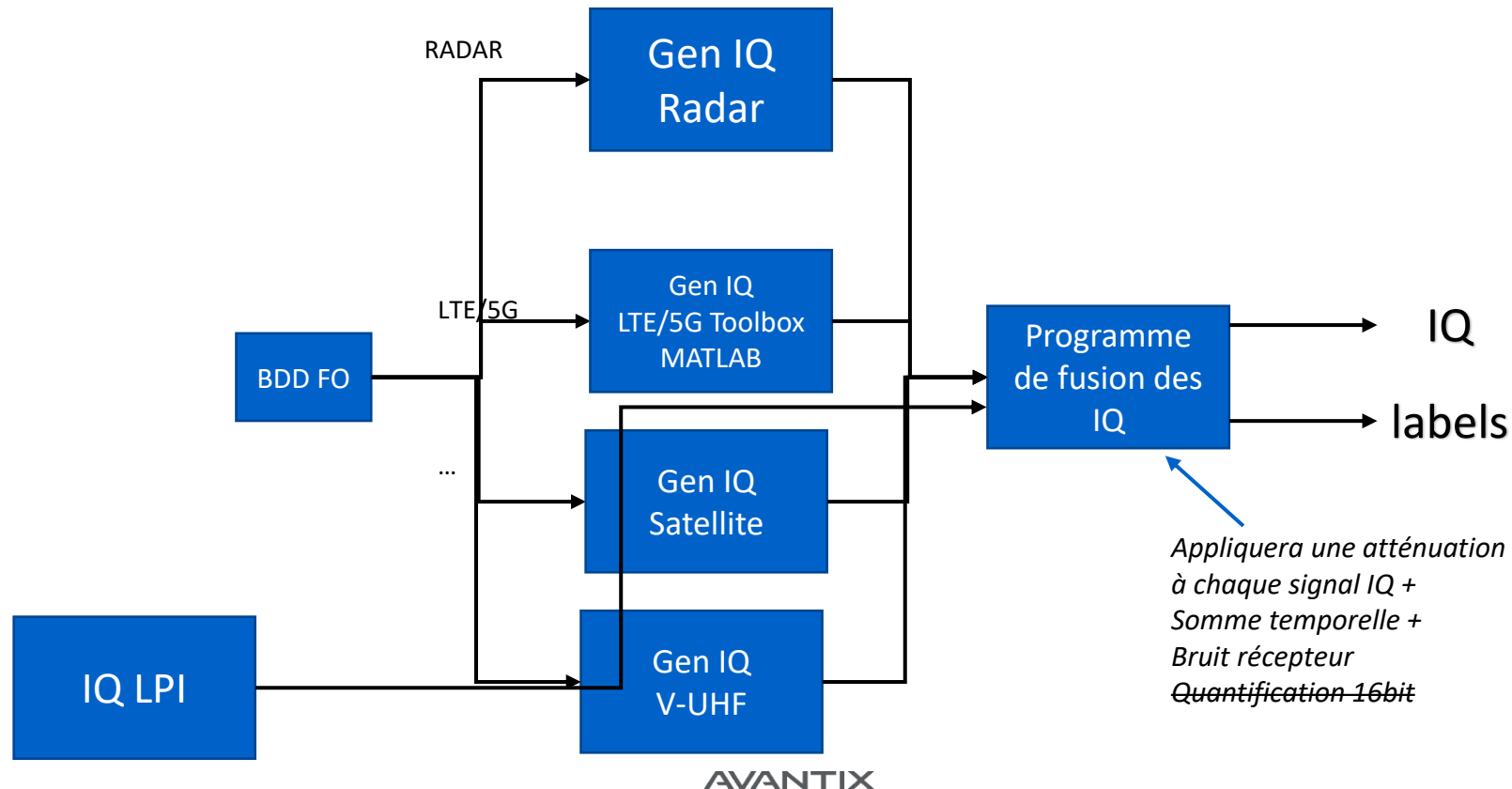
Ce que le programme fera :

- Capable de générer un signal IQ des FO suivantes :
 - **RADAR**
 - **LTE/5G**
 - V-UHF
 - LoRA bande ISM
 - Satellite :
 - THURAYA
 - INMARSAT
 - IRIDUM
 - OneWeb
 - Starlink

Ce que le programme ne fera pas :

- Gérer les cas spécifiques à chaque protocole de communication
- Préfiltrer des FO pour des raisons d'occupation spectrale irréaliste
- Pondérer de manière réaliste les puissances de réception

Spécifications fonctionnelles



Spécifications fonctionnelles

```
import pickle
from iq_sigint_simulator import iq_sigint_simulator

sig_types=['cellular', 'radar']
iq_rec, labels, detailed_labels = iq_sigint_simulator(fc_rec, fs_rec, dt_rec,
sigma_noise, sig_types)

pickle.dump((record, labels), open('record.pkl', 'wb'))
```

- `fc_rec` : Fréquence centrale d'écoute
- `fs_rec` : Fréquence d'échantillonnage de la bande d'écoute
- `dt_rec` : Durée en secondes de l'écoute
- `sigma_noise` : Ecart-type de l'amplitude crête du bruit en Volt
- `sig_types` : Familles des formes d'ondes à simuler

Spécification en cours de rédaction :

\\srv-filer01.global.ad\AFFAIRES TELECOM\1-10-R&D-2200\TECHNIQUE\DOCUMENTS\2_Spécification\DP078293STB004_00-Specification Simulateur R&D ELINT 2023.docx

Spécifications fonctionnelles

L'utilisateur renseigne les familles de formes d'ondes (cellulaire ou RADAR) qu'il souhaite générer sur une fenêtre temps-fréquence.

La fonction *iq_sigint_generator* va ensuite piocher aléatoirement parmi les FO éligibles et générer la fusion de tous les IQ.

Contraintes d'éligibilités :

- RADAR: Génère entre 200k et 1M d'impulsions par seconde pour une bande d'écoute de 500 MHz
- Cellulaire : Un seuil d'occupation spectrale est renseigné par l'utilisateur en ratio

SNR : Cas taille fft optimal

Pour chaque signal, un SNR est choisi aléatoirement entre 3dB et 40dB en apprentissage. On peut envisager sur la base de test entre -10dB et 40dB.

A partir de la puissance de bruit et du SNR, l'amplitude crête du signal est calculé.

$$SNR_{dB} = DSP_{x_{dB}} - DSP_{N_{dB}}$$

$$\Rightarrow DSP_{x_{dB}} = SNR_{dB} + DSP_{N_{dB}}$$

$$\Rightarrow DSP_x = 10^{\frac{SNR_{dB}}{10}} * \frac{\sigma_N^2}{f_s}$$

$$\Rightarrow \frac{A^2}{2} * k_{type} = 10^{\frac{SNR_{dB}}{10}} * \frac{\sigma_N^2}{f_s}$$

$$\Rightarrow A = \sqrt{\frac{2}{f_s * k_{type}} 10^{\frac{SNR_{dB}}{10}} * \sigma_N}$$

$$\text{Avec } k_{type} = \begin{cases} \sqrt{\frac{pw}{\Delta f}} & \text{si chirp radar} \\ pw_{moment} & \text{si pmop radar} \\ pw & \text{si radar no mod ou cw} \\ \frac{1}{\Delta f} & \text{sinon} \end{cases}$$

SNR effectif

Dans la fenêtre de la bounding box, on récupère la valeur max en dB du spectrogramme (S_{xx}) que l'on soustrait à la DSP du bruit.

$$SNR_{eff_{dB}} = \max(S_{xx_{dB}}[x_{min}:x_{max}; y_{min}:y_{max}]) - DSP_{N_{dB}}$$

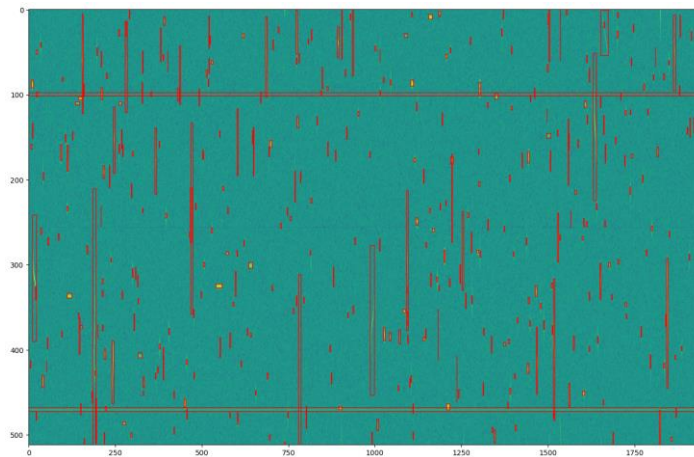
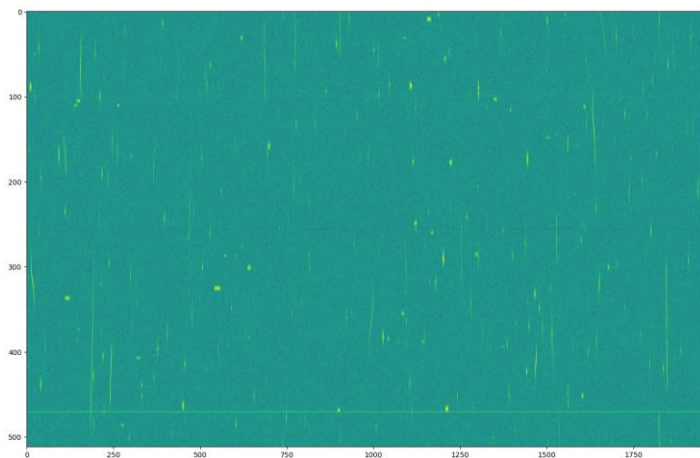
On ne conserve que les SNR supérieur à 10dB.

A améliorer :

Labelizer

Implémentation d'une fonction permettant de générer des bounding box quelques soit les paramètres du spectrogramme.

Amélioration de la hauteur de bbox : Prise en compte des lobes secondaires

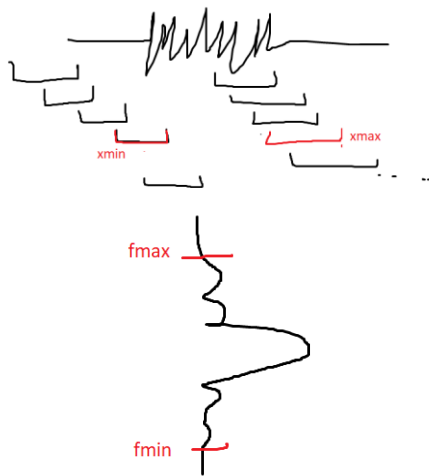


Exemple : $nfft = 512$, $Overlap = 50\%$, $SNR_{eff_min} = 1dB$

Labelizer

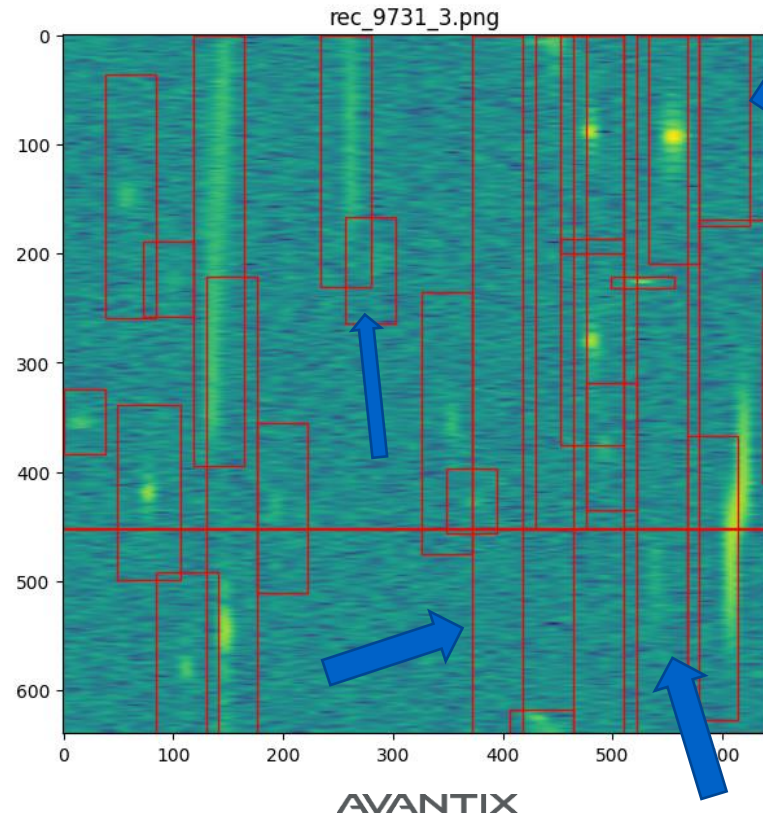
Détails d'implémentation :

1. Recherche xmin et xmax : On cherche la première fenêtre qui rencontre l'impulsion (xmin) et la dernière pour xmax
2. Recherche ymin et ymax : Via fmin et fmax et recherche du dernier lobes secondaires visibles
3. Filtre des signaux ayant un SNR effectif en dessous de 10 dB : On récupère la valeur max de la bbox pour calculer le SNR effectif

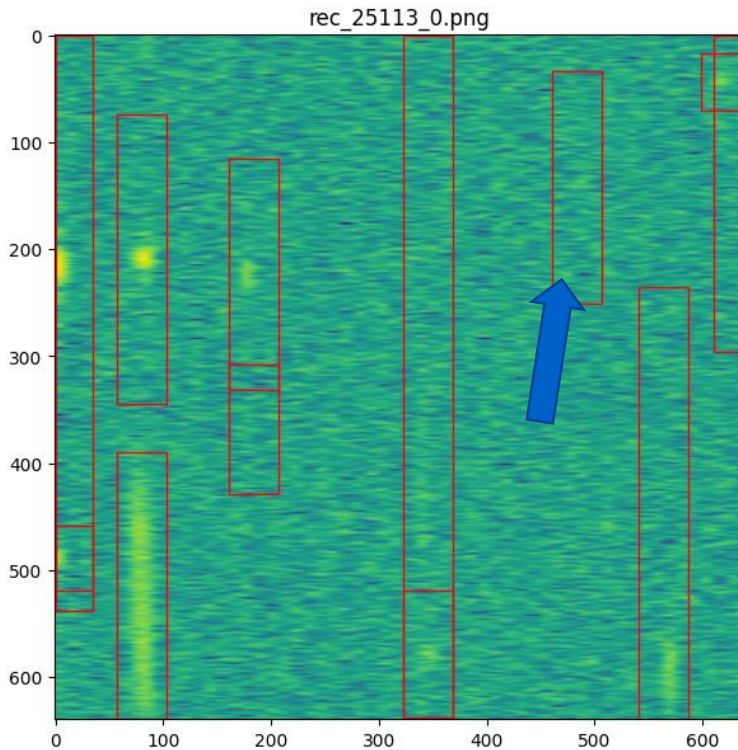


$$SNR_{eff\,dB} = \max(Sxx_{dB}[x_{min}:x_{max};y_{min}:y_{max}]) - DSP_{N\,dB}$$

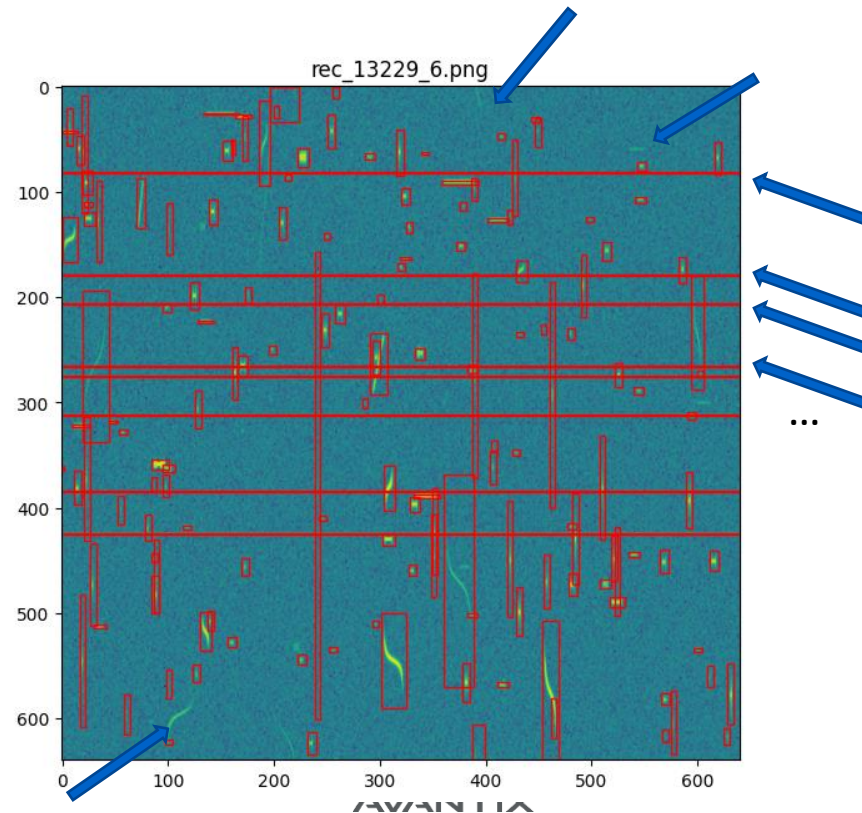
Labelizer : exemples



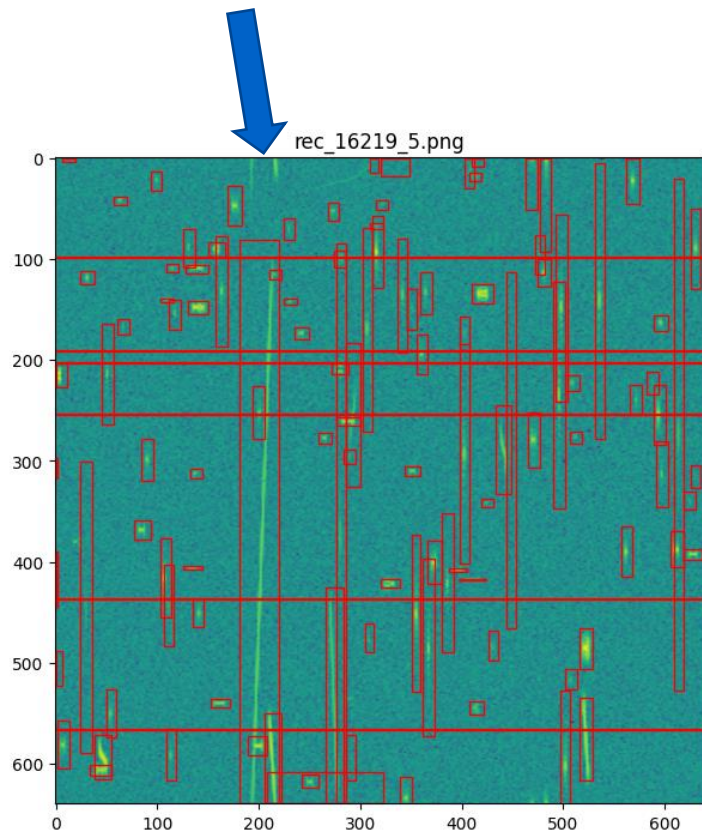
Labelizer : exemples



Labelizer : exemples



Labelizer : exemples



Labelizer

Amélioration à intégrer :

1. Meilleur calcul des ymin et ymax

2. Labélisation aliasing

Génération BT 0.1.0

Paramètres :

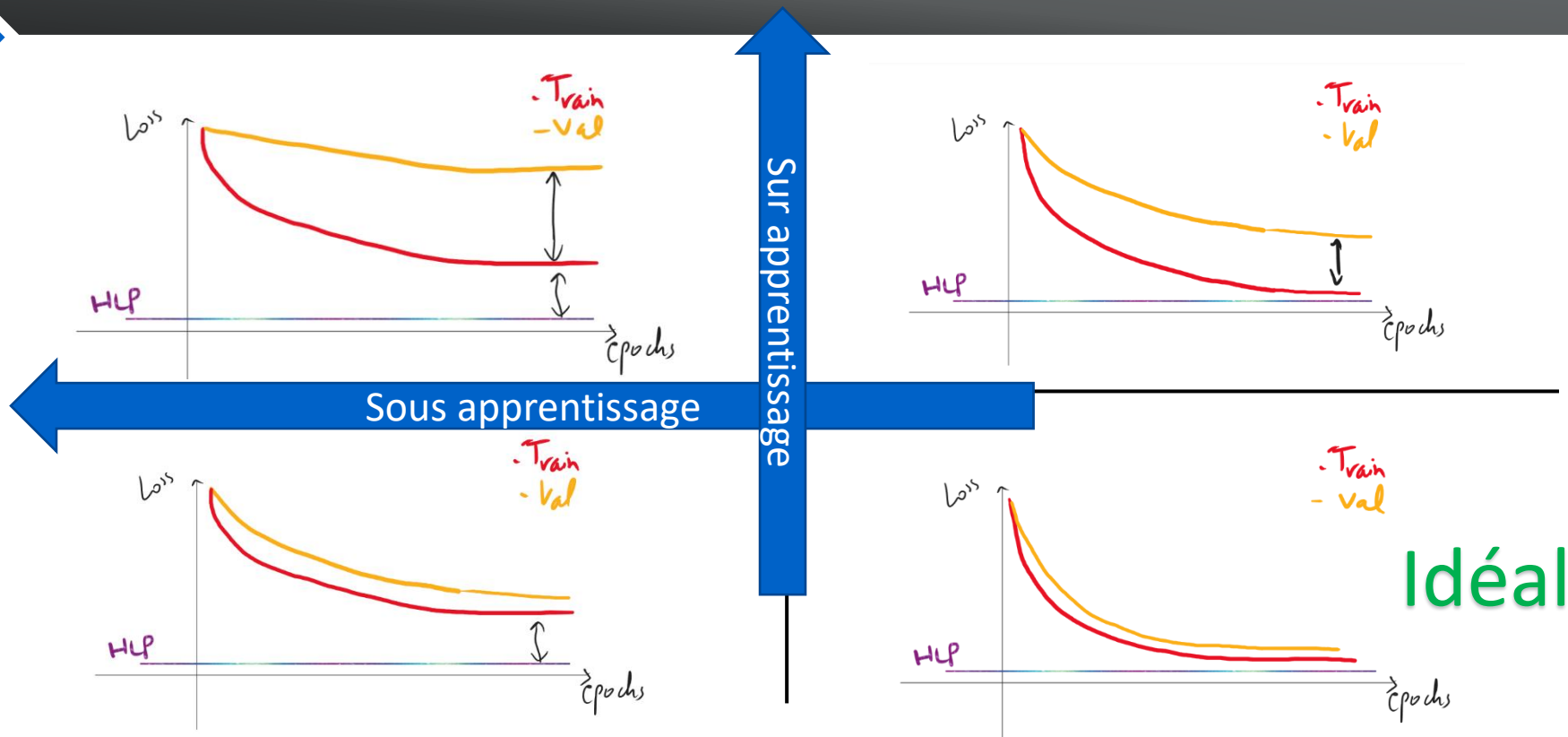
```
fc_rec = 3 GHz  
fs_rec = [1, 5, 25, 100, 500, 1e3, 2e3, 4e3] MS/s  
dt_rec = 1ms  
dt_total = ~250 ms → 15k enregistrements  
sigma_noise = 1mV  
sig_types = 'radar'  
snr_db = 3 à 40dB
```

BT YOLO :

```
fs_rec = [100, 500, 1e3] MS/s  
dt_rec = 1ms / 7 → Split en 7 images  
Definition = [640 pix * 640 pix]. Upsampling only  
Train | Val | Test : 28280 imgs | 5964 imgs | 6076 imgs
```

```
sig_types = 'no_mod_p_mod', 'f_mod'  
snr_db_eff > 8 dB
```

Courbes d'apprentissage Deep Learning

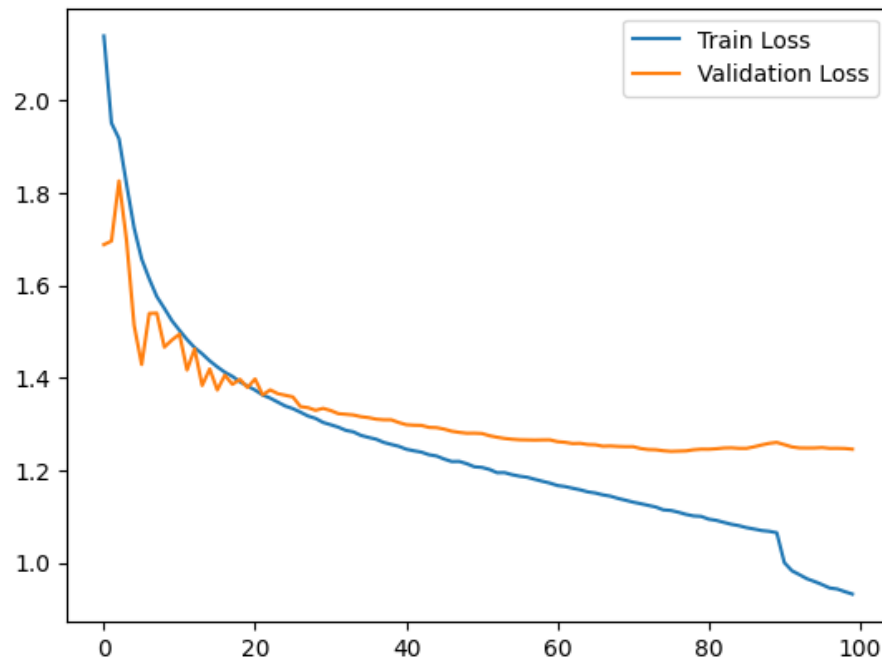


Idéal

YoloV8 : Courbes d'apprentissage

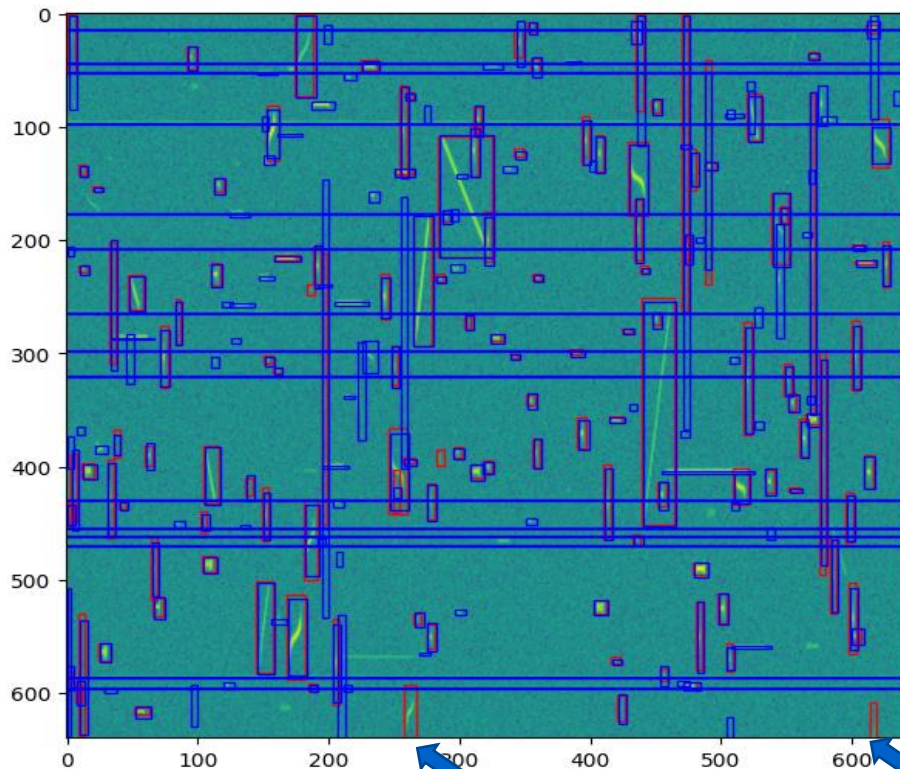
Apprentissage sur BT0.1.0 sans CW :

- Erreur sur train/val surestimée
- Difficulté du modèle de converger car étiquetage inconsistent



YoloV8 : Exemple de résultats

Train

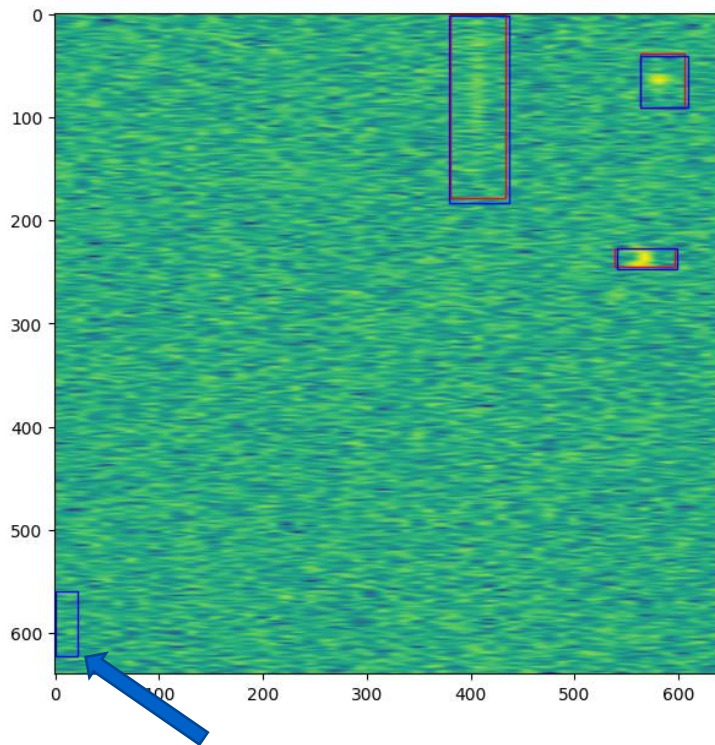


AVANTIX

Vérité terrain
Prédiction

YoloV8 : Exemple de résultats

Train

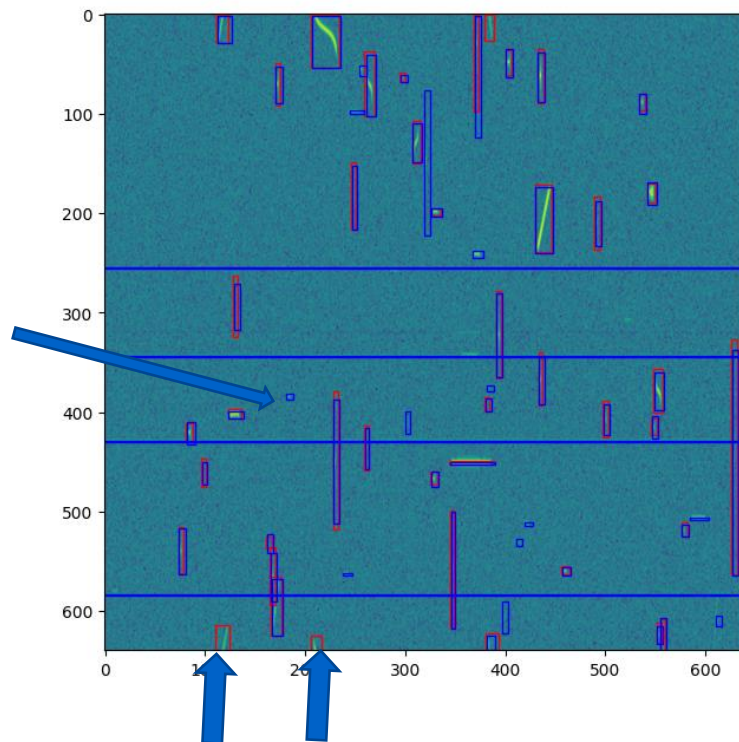


Vérité terrain

Prédiction

YoloV8 : Exemple de résultats

Train



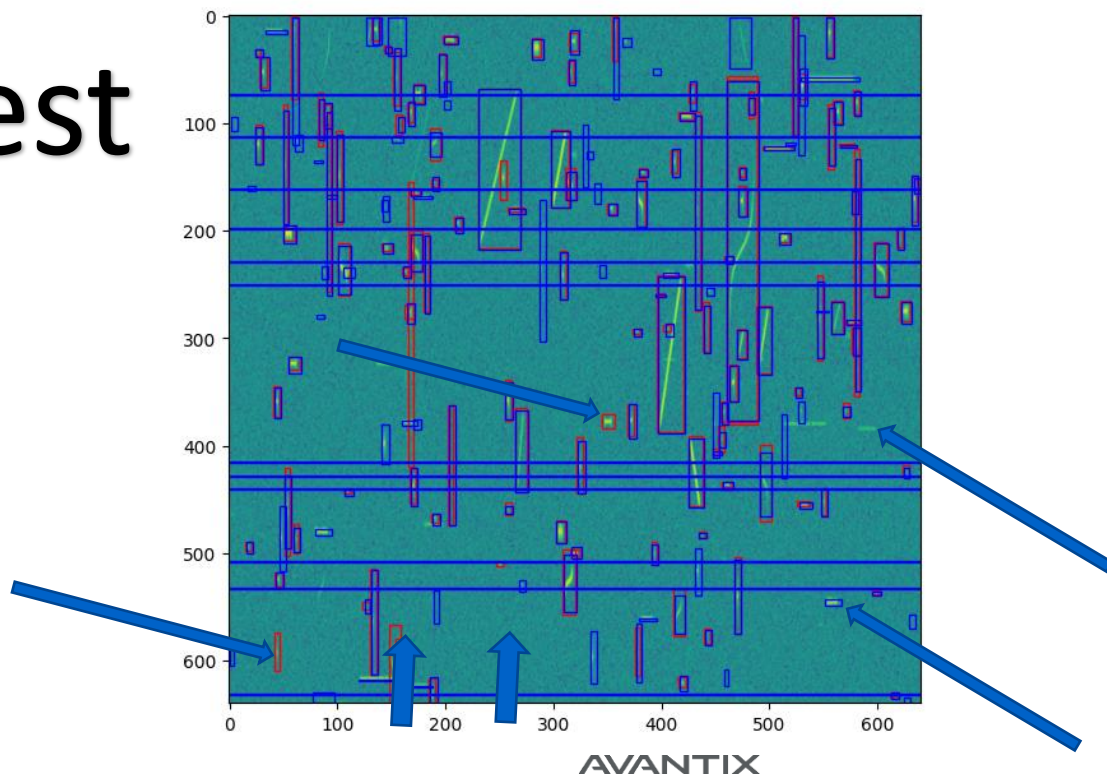
AVANTIX

Vérité terrain

Prédiction

YoloV8 : Exemple de résultats

Test

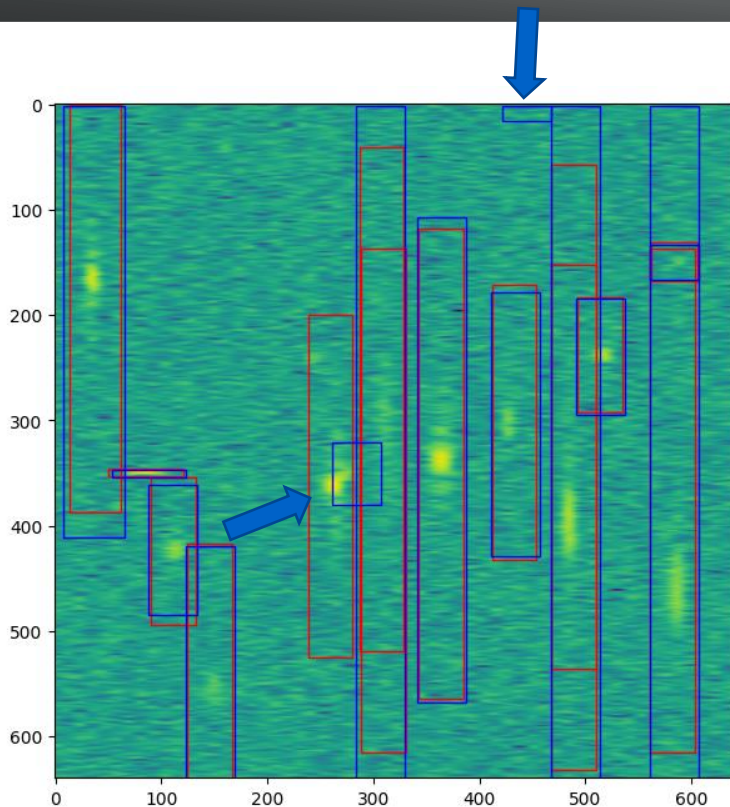


Vérité terrain

Prédiction

YoloV8 : Exemple de résultats

Test

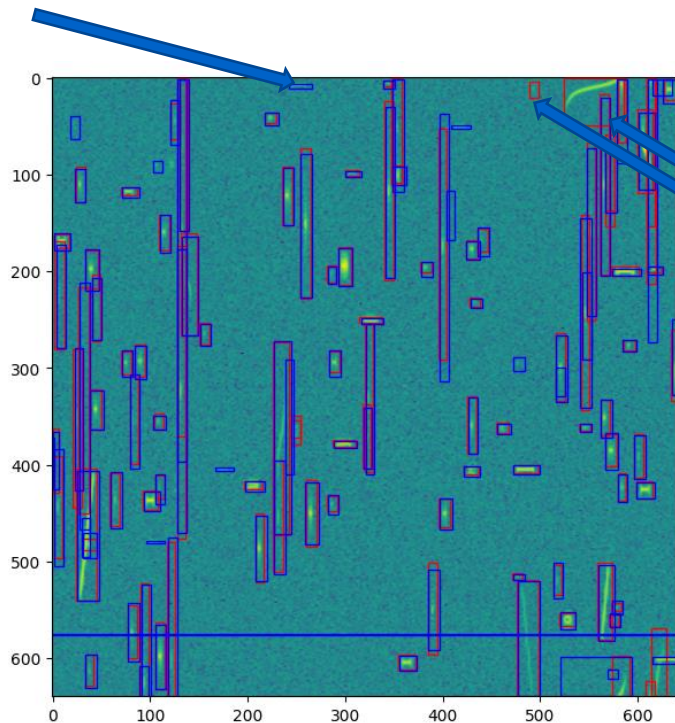


Vérité terrain

Prédiction

YoloV8 : Exemple de résultats

Test



Vérité terrain

Prédiction

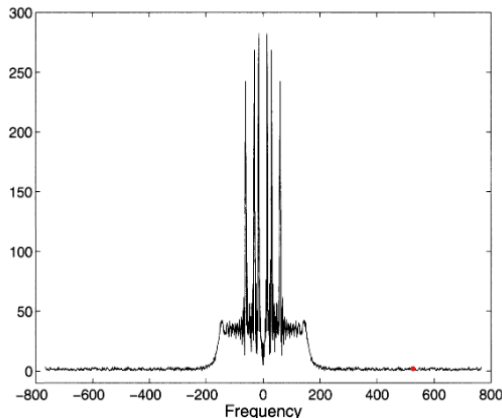
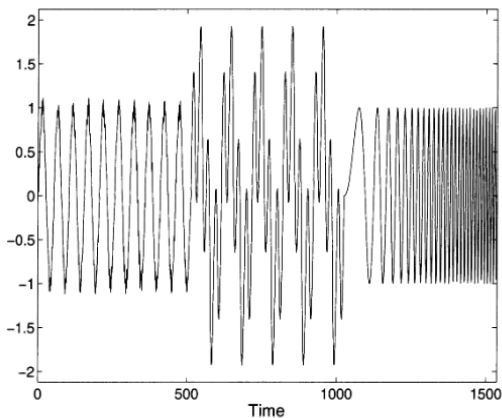
AVANTIX

Analyse temps fréquence

10/01/2024

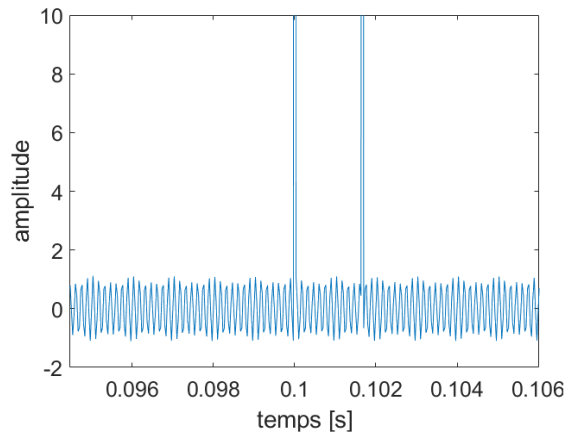
© Avantix – for internal use

Principe de l'ATF



- ATF pensée pour l'analyse de signaux dont les composantes fréquentielles varient
- « retrouver la partition à partir du signal »
- Aussi très utilisé en physique quantique (la fonction d'onde du moment est la TF de la fonction d'onde de la position)

Spectrogramme



← 0s 0,2s →

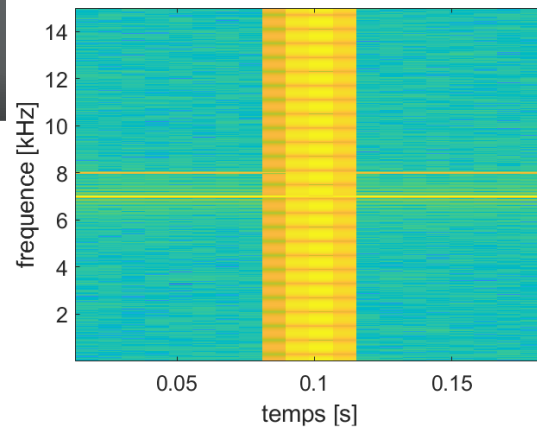
6000 samples

$F_e = 30\text{kHz}$

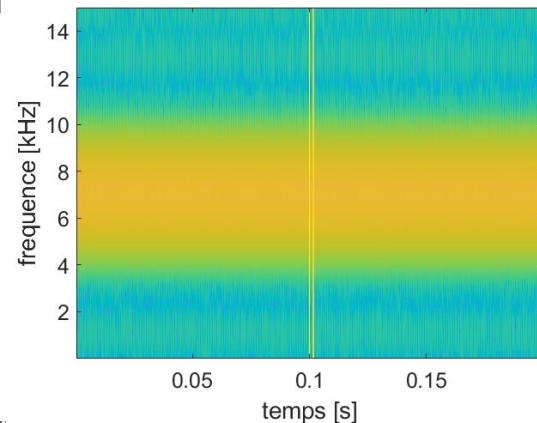
CW à 7 et 8kHz (ratio de 1/10 entre les amplitudes)

Deux Diracs écartés de 50 samples

fft sur une fenêtre
glissante de taille N

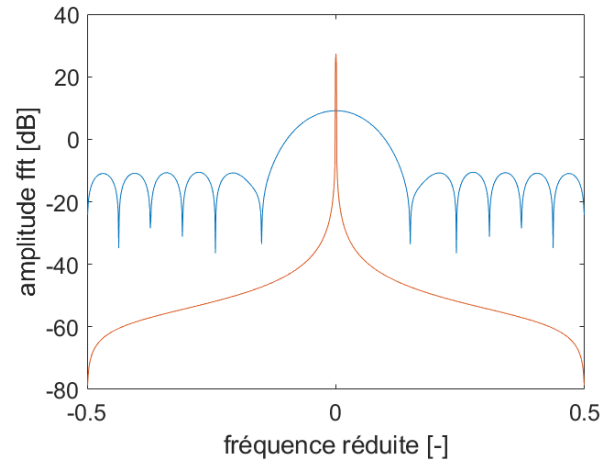
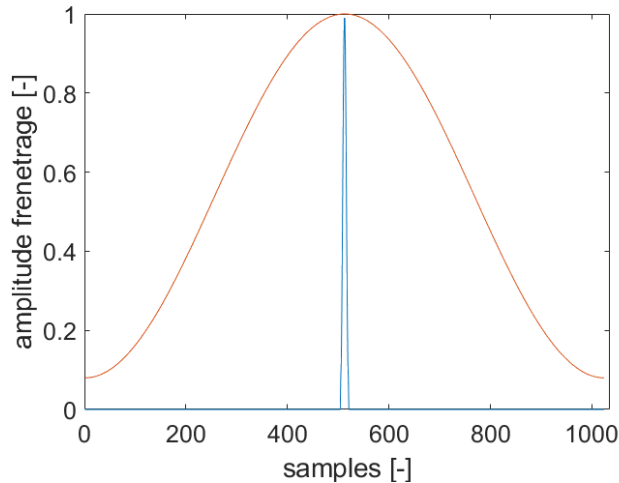


N=1024



N=16

Incertitude temps-fréquence



$$\left(\int_{-\infty}^{\infty} (x - a)^2 |f(x)|^2 dx \right)^{1/2} \left(\int_{-\infty}^{\infty} (\omega - b)^2 |\hat{f}(\omega)|^2 d\omega \right)^{1/2} \geq \frac{1}{4\pi} \|f\|_2^2.$$

Aucune fenêtre ne donne de bonne résolution en temps et en fréquence

Winger-Ville et classes de Cohen

$$\text{WVD}_x(t, f) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau.$$

- Pas de fenêtrage
- Vient de la physique quantique
- **Mesure l'énergie**
- Problématique à temps discret

$$\text{WVD}_x(n, k) = \sum_{m=-N}^N x(n + m) x^*(n + m) e^{-j4\pi k m/N}$$

$$\text{WVD}_x(n, k) = \sum_{m=-N}^N x(n + m/2) x^*(n - m/2) e^{-j2\pi k m/N}.$$

Les classes de Cohen sont des régularisations de la transformée de Winger-Ville

$$C_x^\sigma(t, f) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{WVD}_x(\tau, \nu) \sigma((t - \tau), (f - \nu)) d\tau d\nu$$

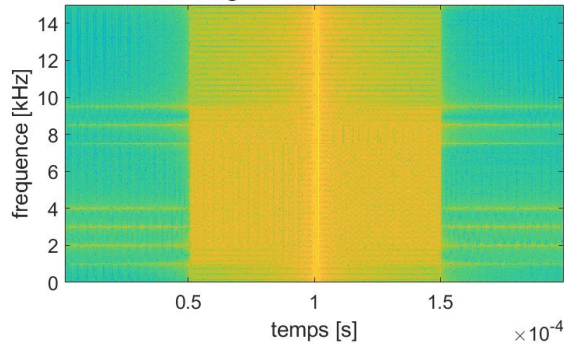
$$\text{SPWVD}_x^{g,H}(t, f) = \int_{-\infty}^{\infty} g(t) H(f) x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau.$$

-> On remet des fenêtres

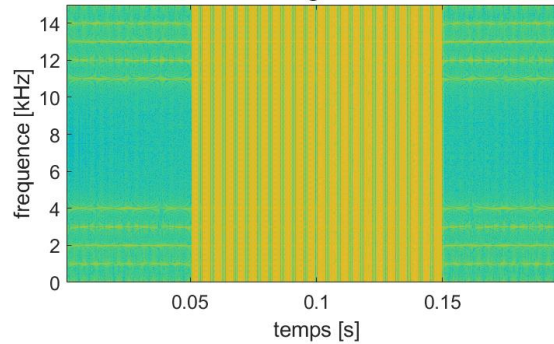
Wigner-Ville et classes de Cohen (2)

Exemple sur un scénario facile, 2 CW pas très proches (2 et 4 kHz, fe à 30kHz) et deux Dirac proches au milieu:

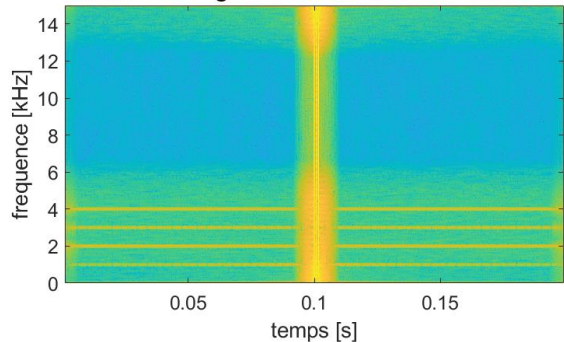
Wigner-Ville de matlab



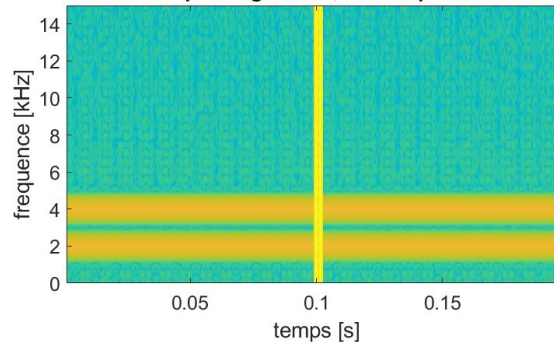
mon Wigner-Ville



Wigner-Ville matlab filtré



spectrogramme, 64 samples



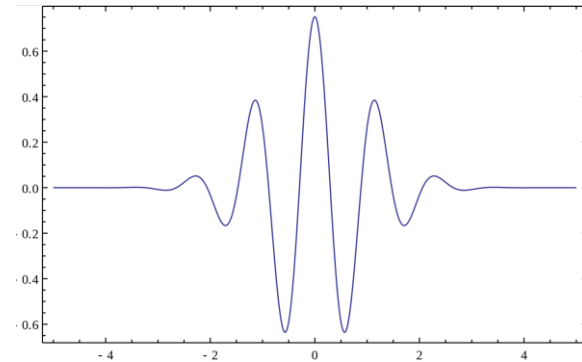
- Gros problèmes de coefficients binomiaux
- Les diracs débordent sans filtrage
- Le spectrogramme est bcp mieux

Ondelettes

On décompose le signal dans la base $\{\psi_{jk} : j, k \in \mathbb{Z}\}$ avec $\psi_{jk}(x) = 2^{\frac{j}{2}} \psi(2^j x - k)$

ψ est l'ondelette mère, elle est dilatée/compressée et décalée, exemple

- Le «socle» dépend de la fréquence
- Adapté si les hautes fréquences sont plus localisées que les basses fréquence (image)
- Pas adapté dans notre cas



Décomposer le signal dans des bases de Fourier locales (spectrogramme) est plus adapté dans notre cas.

Conclusion

- ▶ On veut identifier des signaux malgré le bruit (y compris en bas SNR)
- ▶ Localisés en temps et en fréquence sur des fenêtres très variées
- ▶ Pas de solution miracle pour l'analyse multi-résolution dans la littérature (encore moins avec du bruit)
- ▶ **Le spectrogramme est le meilleur outil**
- ▶ Il faut adapter les fenêtres aux types de signaux que l'on voudra identifier
- ▶ Plus on veut être résistant au bruit, plus il faudra de tailles de fenêtres

AVANTIX

MERCI