

AVANTIX

Avancement Simulateur IQ

15/11/2023

© Avantix – for internal use

Sommaire

- ▶ Simulateur
- ▶ RADAR

AVANTIX

Simulateur

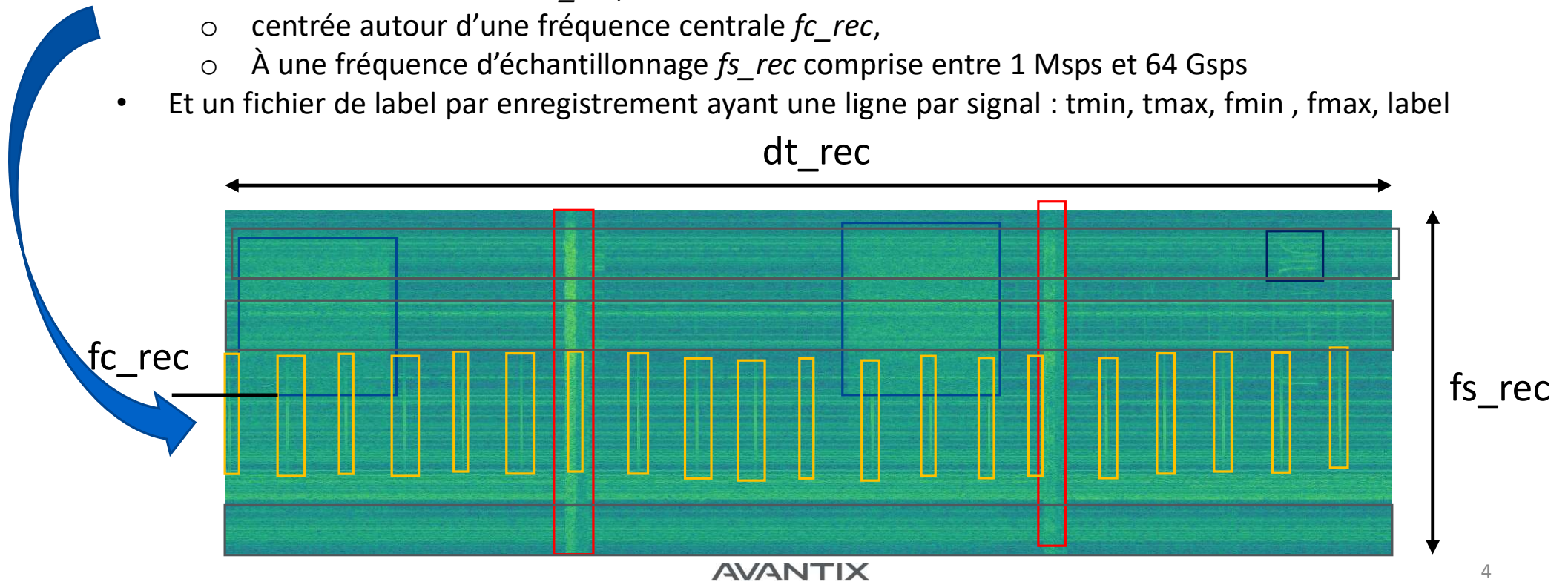
15/11/2023

© 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 Msps et 64 Gsps
- Et un fichier de label par enregistrement ayant une ligne par signal : $tmin$, $tmax$, $fmin$, $fmax$, label



Simulateur

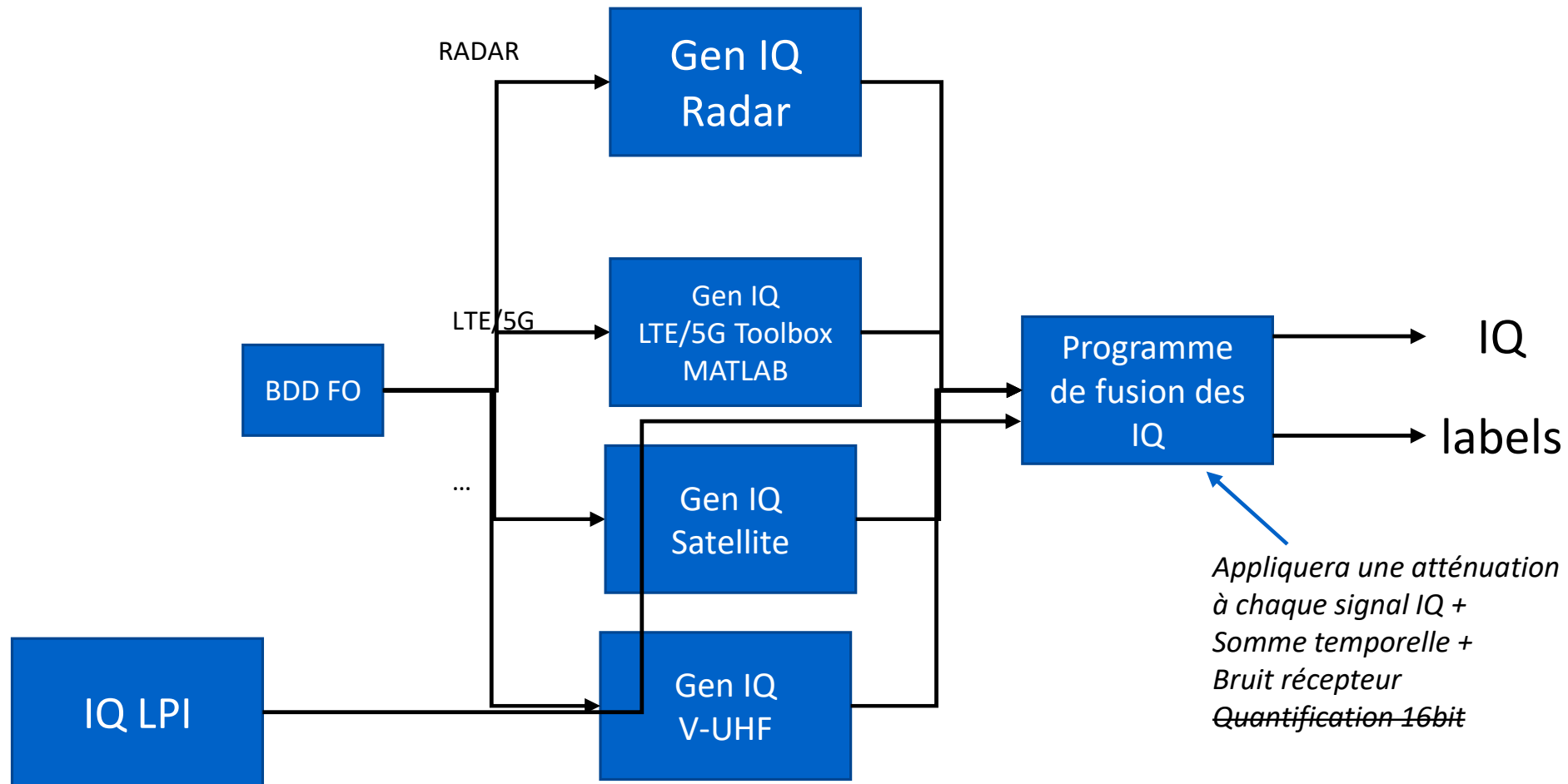
Ce que le programme fera :

- 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



AVANTIX

Spécifications fonctionnelles

```
import pickle
from iq_sigint_simulator import iq_sigint_simulator

sig_types=['cellular', 'radar']
iq_rec, labels = iq_sigint_simulator(fc_rec, fs_rec, dt_rec, sig_types=
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
- `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.

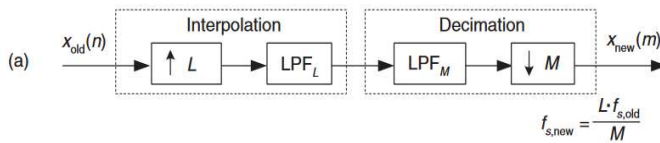
La fonction *iq_sigint_generator* va ensuite piocher aléatoirement parmi les FO éligibles et générer la fusion de tous les IQ.

Contraintes d'éligibilités :

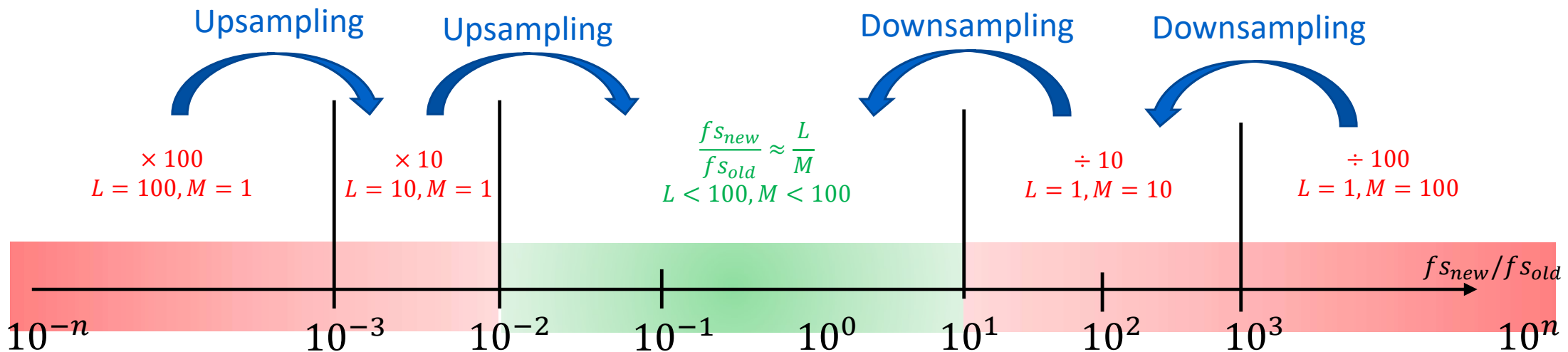
- RADAR: Génère entre 200k et 1M d'impulsions par seconde
- Cellulaire : Un seuil d'occupation spectrale est renseigné par l'utilisateur en ratio

Fusion des IQ : Sample Rate Conversion

Réaliser une cascade de upsampling/downsampling afin d'obtenir le rapport de conversion souhaité. La cascade se fait par ratio de 100 ou 10 de sur ou sous-échantillonnage sauf la dernière qui impose un rapport L/M avec L et M inférieur à 100.



Objectif : $\frac{f_{s,new}}{f_{s,old}} \approx \frac{L}{M} \times (10^2)^{n_2} \times (10^1)^{n_1} \times (10^{-1})^{n_{-1}} \times (10^{-2})^{n_{-2}}$
Avec $L < 100$ et $M < 100 \rightarrow$ Garantie une erreur < 1%

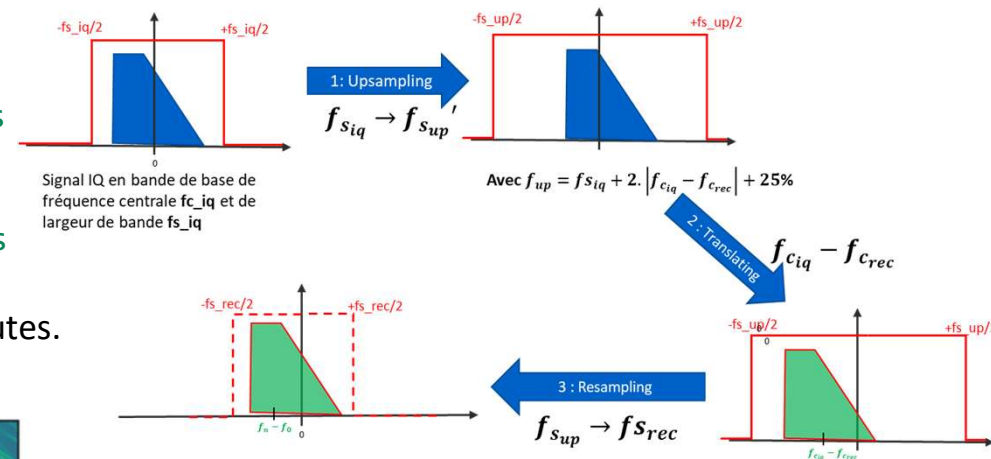
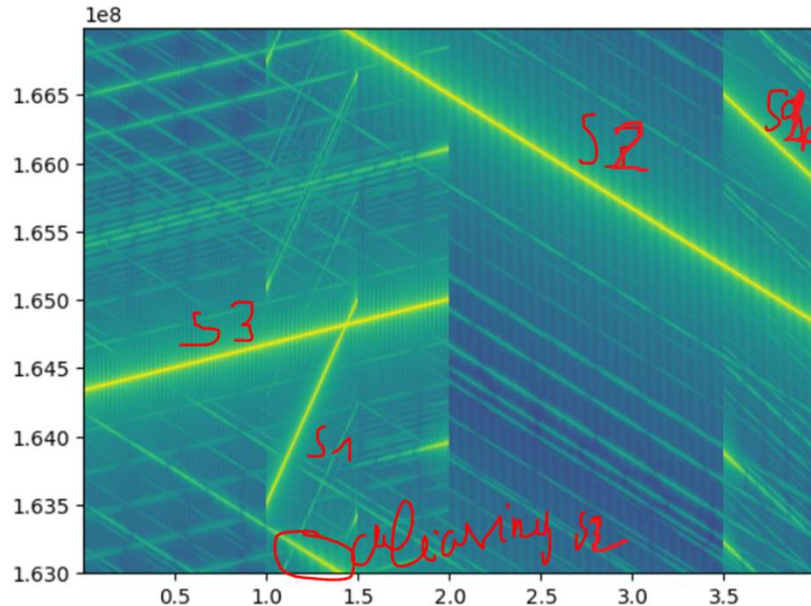


Fusion des IQ

En test, 4 chirps à fusionner :

- s1: tstart = 1s, fstart=163.5MHz, fend=165MHz, duration=0.5s
- S2: tstart = -1s, fstart=169MHz, fend=164MHz, duration=6s
- S3: tstart = -1s, fstart=164MHz, fend=165MHz, duration=3s
- S4: tstart = 3.5s, fstart=166.5MHz, fend=164MHz, duration=2s

Chaque signal a été projeté en fréquence et en temps dans la bande des écoutes. Ensuite les 4 signaux ont été additionnés.



Dev en cours

```
import matplotlib.pyplot as plt
import numpy as np

from iq_sigint_generator import iq_sigint_generator

np.random.seed(1)
sig_types = ['cellular', 'radar']

fs_rec = 4e6 # Hz
dt_rec = 400e-6
fc_rec = np.random.randint(750, 4_300)*1e6 # Hz
iq_rec1, label_list1 = iq_sigint_generator(fc_rec, fs_rec, dt_rec, sig_types)

nfft = 512
_ = plt.specgram(iq_rec1, NFFT=nfft, Fs=fs_rec, Fc=fc_rec, detrend=None, noverlap=int(0.75*nfft))
```

✓ 11.7s

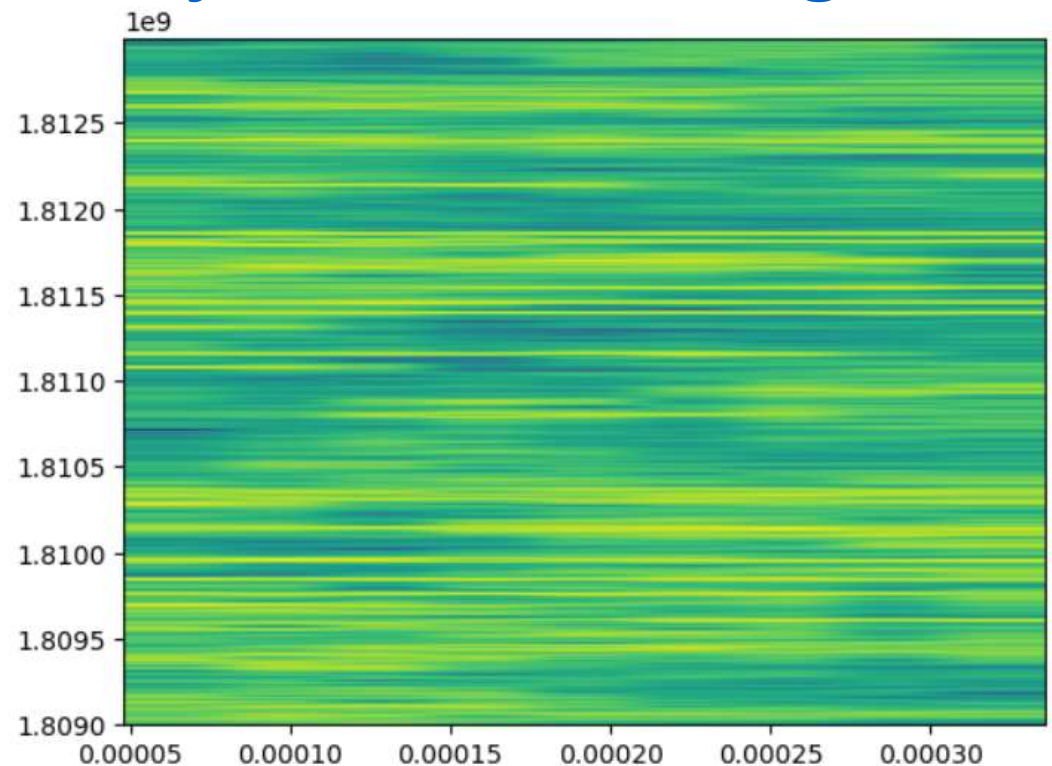
Num imp radar=276
cellular
Bandwidth occupancy: 62.70%
radar
100
200

label_list1

✓ 0.0s

```
[{'fc_iq': 1817994722.8342829,
  'tmn': '2a',
  'bw(MHz)': 15.0,
  'dm': 'FDD',
  'totsubframes': 10,
  'tstart_iq': -0.0072259121998412956},
 {'pw': 1.0525948689799727e-07,
  'fc_iq': 1812512469.7455637,
  'mod': None,
  'tstart_iq': 1.0876092663996968e-05},
 {'pw': 2.2508033397597535e-05,
  'fc_iq': 1856320093.2720733,
  'mod': {'type': 'frequency',
```

Ajouter bounding box



AVANTIX

RADAR

15/11/2023

© Avantix – for internal use

F0 Radar

DI (ns)	Sans Modulation	Modulations de Fréquence (MHz/μs)		Modulations de Phase Phase / Nombre de moments	
10-100	Oui	-	-	-	
100-1_000	Oui	-	-	-	
1_000-10_000	Oui	20	100	180°	7, 8
10_000-100_000	Oui	2	10	180°	7, 8, 13, 25, 33, 39
				60°	25, 30
				45°	8, 35, 70
100_000-1_000_000	Oui	0,2	1	180°	7, 8, 13, 25, 33, 39
				60°	25, 30
				45°	8, 35, 70
1_000_000-10_000_000	Non	0,02	0,1	-	
10_000_000-100_000_000	Non	0,002	0,01	-	
100_000_000-inf	Non	0,0002	0,001	-	
CW	Oui	-	-	-	

Création d'une fonction pour la génération d'une impulsion iq

Les paramètres de cette fonction seront :

- Fréquence d'échantillonnage (Hz)
- Fréquence centrale de notre impulsion (Hz)
- Durée de l'impulsion (seconde)
- Paramètre de modulation (optionnel)
 - Modulation de fréquence (1 couple de paramètre)
 - L'évasion en fréquence (Hz)
 - L'aspect de la modulation
 - Linéaire = chirp
 - Quadratique = S-law (paramètre à fixer)
 - (autres fonctions à créer si besoin)
 - Modulation de phase
 - Angle
 - Séquence (ex: 0,1,0,0,0,1,0,1,1,1)

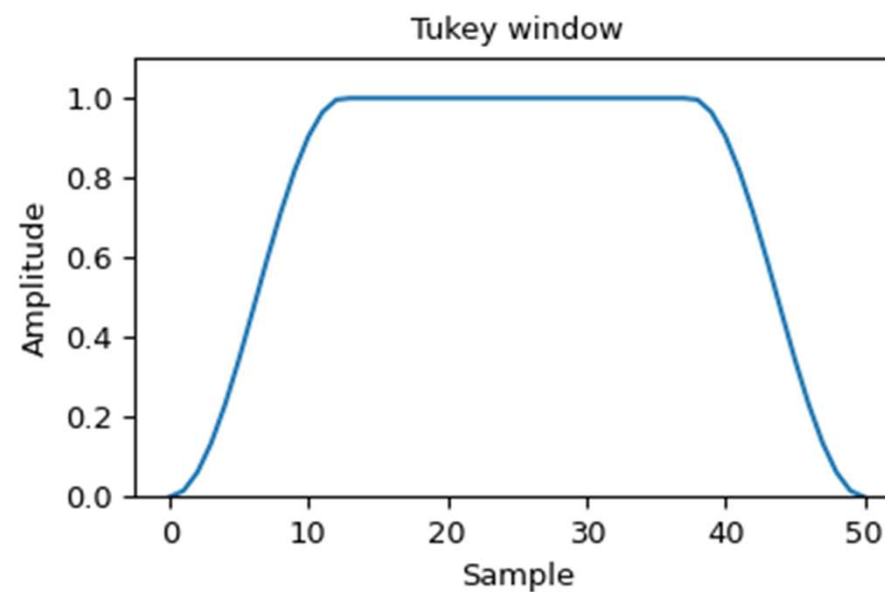
En sortie de cette fonction nous avons une impulsion avec les paramètres renseignés.

(voir exemple)

F0 Radar

DI (ns)	Sans Modulation	Modulations de Fréquence (MHz/μs)		Modulations de Phase Phase / Nombre de moments	
10-100	Oui	-	-	-	
100-1_000	Oui	-	-	-	
1_000-10_000	Oui	20	100	180°	7, 8
10_000-100_000	Oui	2	10	180°	7, 8, 13, 25, 33, 39
				60°	25, 30
				45°	8, 35, 70
100_000-1_000_000	Oui	0,2	1	180°	7, 8, 13, 25, 33, 39
				60°	25, 30
				45°	8, 35, 70
1_000_000-10_000_000	Non	0,02	0,1	-	
10_000_000-100_000_000	Non	0,002	0,01	-	
100_000_000-inf	Non	0,0002	0,001	-	
CW	Oui	-	-	-	

Atténuation des bords de l'impulsion



Les phases transitoires sont fixées à une durer de 2μs
Donc 4μs
Soit $p_w/4\mu s$ nous donne le ratio de 0 dans la fonction pour générer la fenêtre

F0 Radar : Loi en S

$$\beta = \tan^{-1}(\alpha), 0 \leq \alpha < \infty \quad (9)$$

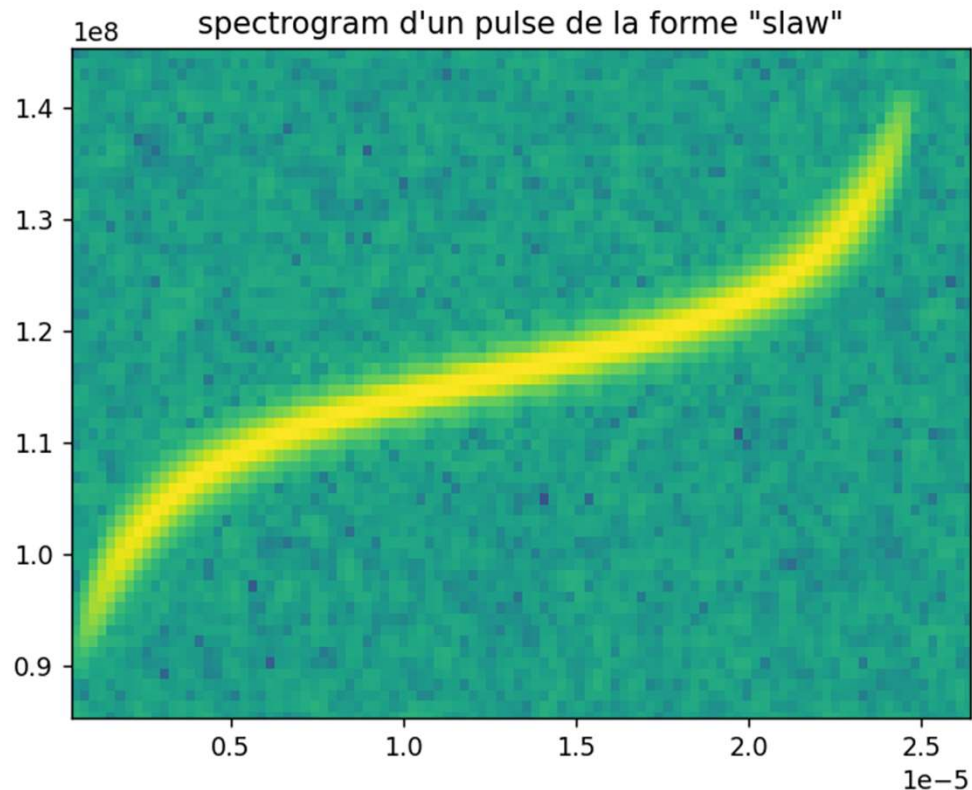
In (9), α is sidelobecontrol factor of NLFM S.

$$f(t)_S = f_c + B \left(\frac{\tan\left(\frac{2\beta t}{\tau}\right)}{2 \tan \beta} \right), -\frac{\tau}{2} \leq t \leq \frac{\tau}{2} \quad (10)$$

In (10), $f(t)_S$ is instantaneous frequency of NLFMS (Hz), τ is pulse duration (s), B is signal bandwidth (Hz), t is time duration (s) and f_c is frequency center (Hz).

$$\phi(t)_S = 2\pi \left(f_c t - \left(\frac{B\tau \log\left\{\cos\left(\frac{2\beta t}{\tau}\right)\right\}}{4\beta \tan(\beta)} \right) \right), -\frac{\tau}{2} \leq t \leq \frac{\tau}{2} \quad (11)$$

$$S(t)_S = \exp(j\phi(t)_S), -\frac{\tau}{2} \leq t \leq \frac{\tau}{2} \quad (12)$$



Exemple :
 $B=1\text{Ghz}$
 $F_c=115,5\text{Mhz}$
 $\Delta f = 61,33\text{Mhz}$
 $P_w=24\mu\text{s}$

Source: <https://pdfs.semanticscholar.org/7962/04c109a91f4fa19535eaa101a563316ddeb1.pdf>

F0 Radar

DI (ns)	Sans Modulation	Modulations de Fréquence (MHz/μs)		Modulations de Phase Phase / Nombre de moments	
10-100	Oui	-	-	-	
100-1_000	Oui	-	-	-	
1_000-10_000	Oui	20	100	180°	7, 8
10_000-100_000	Oui	2	10	180°	7, 8, 13, 25, 33, 39
				60°	25, 30
				45°	8, 35, 70
100_000-1_000_000	Oui	0,2	1	180°	7, 8, 13, 25, 33, 39
				60°	25, 30
				45°	8, 35, 70
1_000_000-10_000_000	Non	0,02	0,1	-	
10_000_000-100_000_000	Non	0,002	0,01	-	
100_000_000-inf	Non	0,0002	0,001	-	
CW	Oui	-	-	-	

Dans iq_radar_generator nous avons un générateur de pdw:

J'objectif est de s'assurer que les impulsions sont comprises entre 500Mhz et 18Ghz

Dans le cas où elle ne respecte pas la condition, nous régénérons une nouvelle pdw jusqu'à ce que la condition soit respecter

Problème :

He360 < 500Mhz → fc_radar compris entre 1 MHz et 18GHz

AVANTIX

MERCI

AVANTIX.NET