

AVANTIX

Avancement Simulateur IQ

25/10/2023

© Avantix – for internal use

Sommaire

- ▶ Simulateur
- ▶ FO cellulaire
- ▶ FO RADAR
- ▶ FO Satellite

AVANTIX

Simulateur

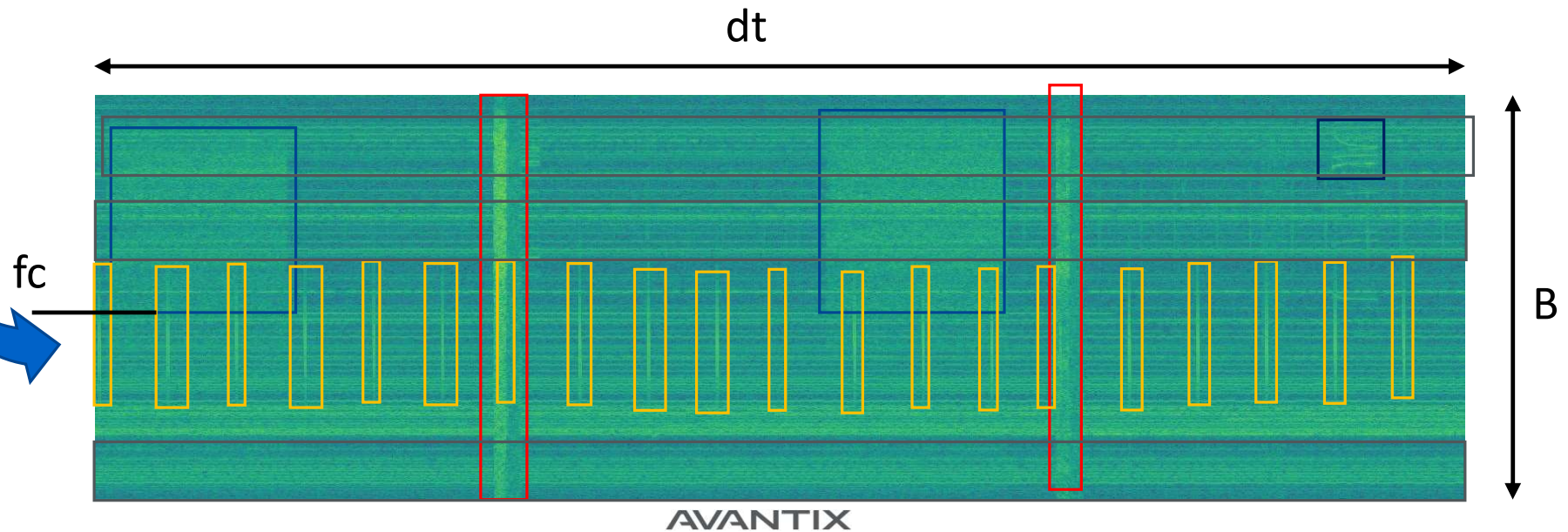
25/10/2023

© Avantix – for internal use

Besoin

Avoir un programme qui génère en sortie :

- un enregistrement IQ
 - d'une certaine durée dt ,
 - centrée autour d'une fréquence centrale fc ,
 - sur une largeur de bande B
- Et un fichier de label par enregistrement ayant une ligne par signal : $tmin$, $tmax$, $fmin$, $fmax$, label



Simulateur

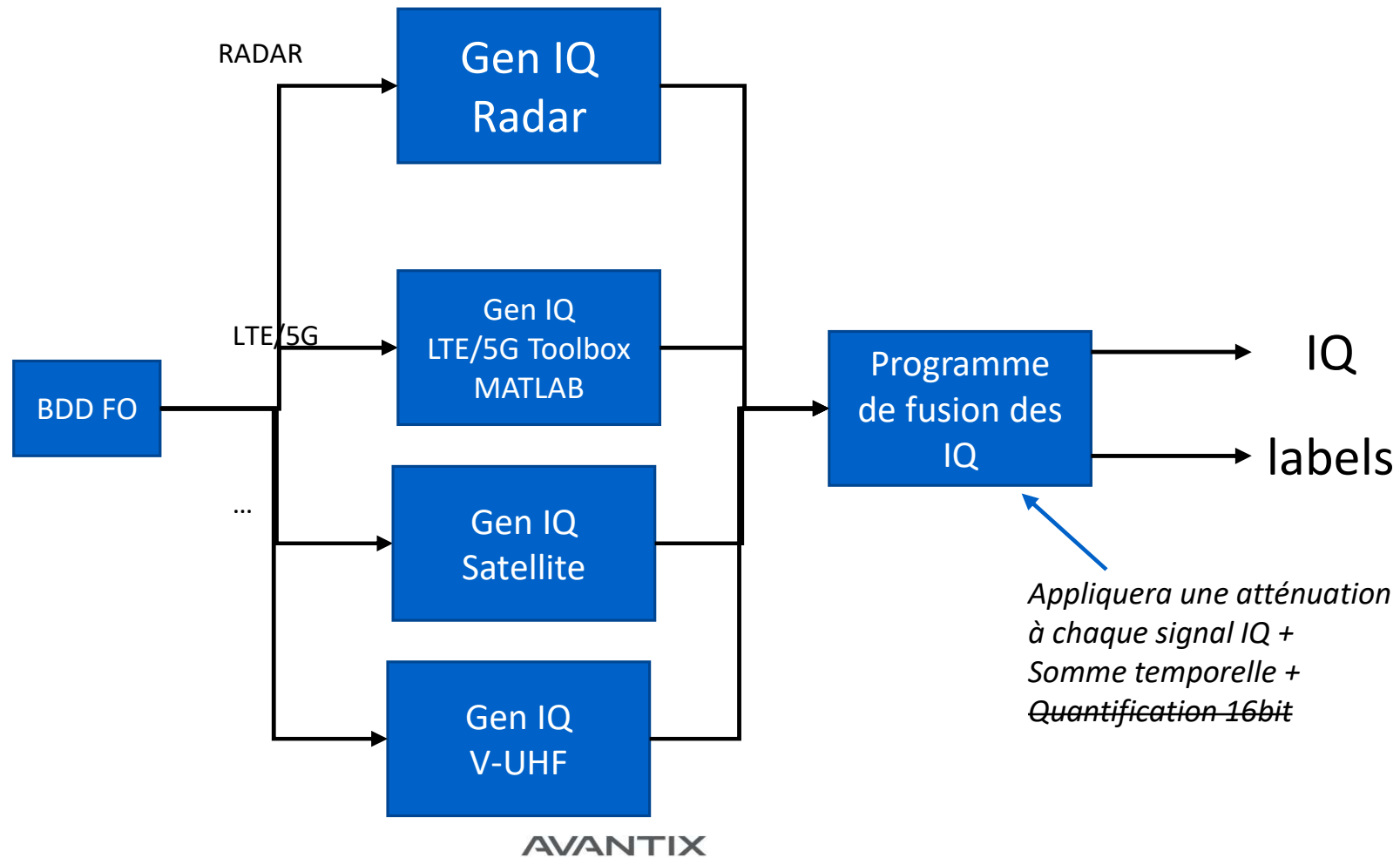
Ce que le programme fera :

- Capable de générer un signal IQ des FO suivantes :
 - **RADAR**
 - **LTE/5G**
 - Satellite défilant
 - Satellite géostationnaire
 - V-UHF
 - LoRA bande ISM

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

python

```
from pkg import Simulator

sim = Simulator(dt, fc, bw,
waveforms=None)

simulation = sim.simulate()

simulation.write('data.pkl')
```

- `dt` définit la durée
- `fc` définit la fréquence centrale
- `bw` définit la largeur de bande.
- `waveforms` est un paramètre optionnel. Par défaut, les formes d'ondes seront sélectionnées de manière aléatoire parmi celles présentes dans la bande B.

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

AVANTIX

FO Cellulaire

25/10/2023

© Avantix – for internal use

Description

- Utilisation des toolbox LTE, 5G et Waveform generator.
- Possibilité de créer un script à partir d'une simulation via le générateur matlab
- Création de 2 fonctions MATLAB pour la génération des IQ
- Utilisation du module matlab.engine afin d'appeler ces fonctions sous python
- Repo : http://10.106.45.116/elint-dev/iq_cellular_generator

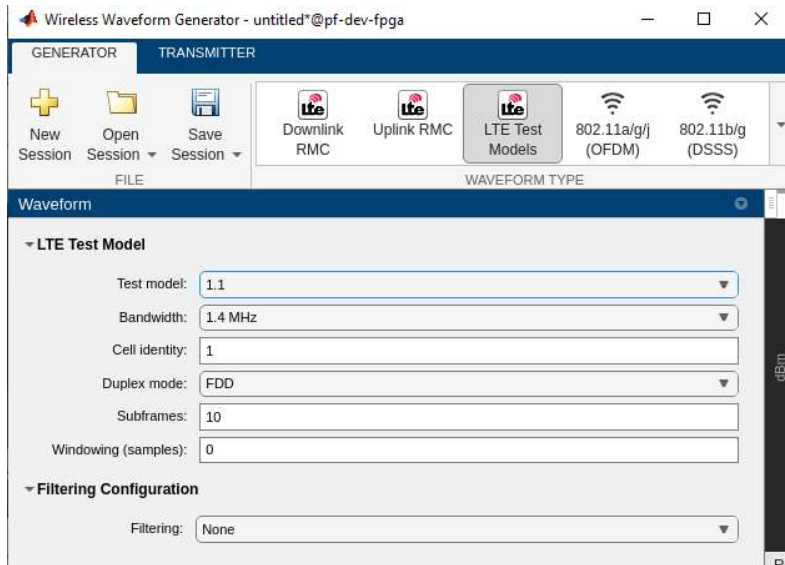
La génération des IQ se base sur des « tests models » qui sont des sortes de scénarios de DL. Voir

<https://fr.mathworks.com/help/lte/ug/lte-downlink-test-model-e-tm-waveform-generation.html>

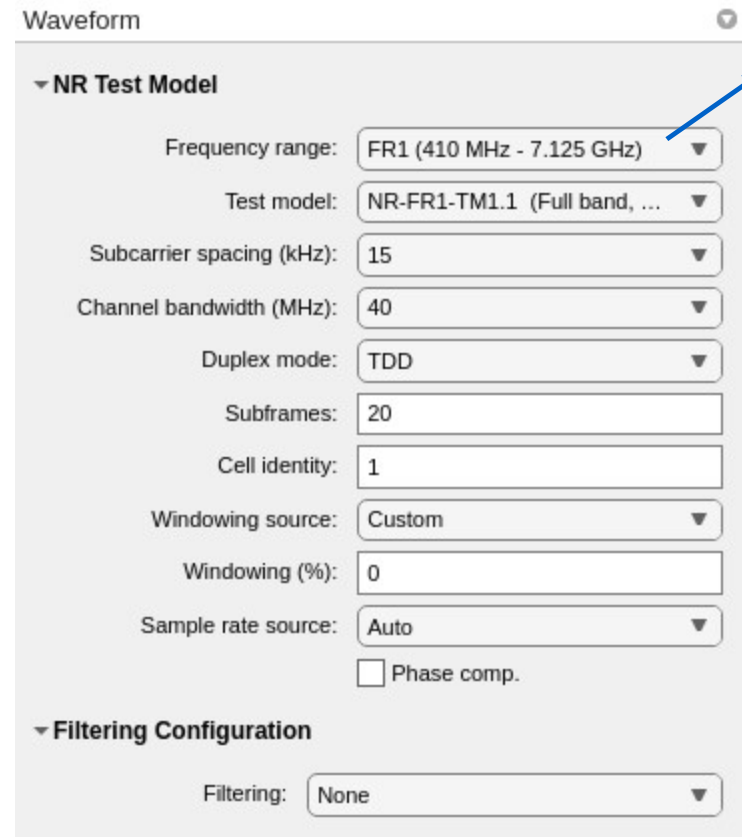
<https://fr.mathworks.com/help/5g/ug/5g-nr-tm-waveform-generation.html>

Test models

LTE



5G



Fixé en FR1

Formes d'ondes LTE

Nom des bandes	Duplex mode	Fmin UL (MHz)	Fmax UL (MHz)	Fmin DL (MHz)	Fmax DL (MHz)	BW (MHz)	1,4	3	5	10	15	20
B1	FDD	1920	1980	2110	2170	60	N/A	N/A	5	10	15	20
B3	FDD	1710	1785	1805	1880	35	1,4	3	5	10	15	20
B7	FDD	2500	2570	2620	2690	30	N/A	N/A	5	10	15	20
B8	FDD	880	915	925	960	35	1,4	3	5	10	N/A	N/A
B20	FDD	832	862	791	821	30	N/A	N/A	5	10	15	20
B28	FDD	703	748	758	803	45	N/A	3	5	10	15	20
B38	TDD	2570	2670	2570	2620	30	N/A	N/A	5	10	15	20
B40	TDD	2300	2400	2300	2400	100	N/A	N/A	5	10	15	20
B41	TDD	2496	2690	2496	2690	100	N/A	N/A	5	10	15	20

Formes d'ondes 5G

Nom des bandes	Duplex mode	Fmin UL (MHz)	Fmax UL (MHz)	Fmin DL (MHz)	Fmax DL (MHz)	W (MHz)	SCS (kHz)	5	10	15	20	25	30	35	40	45	50	60	70	80	90	100
n1	FDD	1920	1900	2110	2170	60	15	5	10	15	20	25	30	N/A	40	45	50	N/A	N/A	N/A	N/A	N/A
n1	FDD	1920	1900	2110	2170	60	30	N/A	10	15	20	25	30	N/A	40	45	50	N/A	N/A	N/A	N/A	N/A
n1	FDD	1920	1900	2110	2170	60	60	N/A	10	15	20	25	30	N/A	40	45	50	N/A	N/A	N/A	N/A	N/A
n3	FDD	1710	1755	1805	1880	75	15	5	10	15	20	25	30	35	40	45	50	N/A	N/A	N/A	N/A	N/A
n3	FDD	1710	1755	1805	1880	75	30	N/A	10	15	20	25	30	35	40	45	50	N/A	N/A	N/A	N/A	N/A
n3	FDD	1710	1755	1805	1880	75	60	N/A	10	15	20	25	30	35	40	45	50	N/A	N/A	N/A	N/A	N/A
n7	FDD	2500	2500	2620	2690	70	15	5	10	15	20	25	30	35	40	N/A	50	N/A	N/A	N/A	N/A	N/A
n7	FDD	2500	2500	2620	2690	70	30	N/A	10	15	20	25	30	35	40	N/A	50	N/A	N/A	N/A	N/A	N/A
n7	FDD	2500	2500	2620	2690	70	60	N/A	10	15	20	25	30	35	40	N/A	50	N/A	N/A	N/A	N/A	N/A
n20	FDD	832	832	791	821	30	15	5	10	15	20	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
n20	FDD	832	832	791	821	30	30	N/A	10	15	20	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
n20	FDD	832	832	791	821	30	60	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
n38	TDD	2570	2600	2570	2620	50	15	5	10	15	20	25	30	N/A	40	N/A	N/A	N/A	N/A	N/A	N/A	N/A
n38	TDD	2570	2600	2570	2620	50	30	N/A	10	15	20	25	30	N/A	40	N/A	N/A	N/A	N/A	N/A	N/A	N/A
n38	TDD	2570	2600	2570	2620	50	60	N/A	10	15	20	25	30	N/A	40	N/A	N/A	N/A	N/A	N/A	N/A	N/A
n40	TDD	2300	2400	2300	2400	100	15	5	10	15	20	25	30	N/A	40	N/A	50	N/A	N/A	N/A	N/A	N/A
n40	TDD	2300	2400	2300	2400	100	30	N/A	10	15	20	25	30	N/A	40	N/A	50	60	70	80	90	100
n40	TDD	2300	2400	2300	2400	100	60	N/A	10	15	20	25	30	N/A	40	N/A	50	60	70	80	90	100
n41	TDD	2496	2600	2496	2690	194	15	5	10	15	20	25	30	35	40	45	50	N/A	N/A	N/A	N/A	N/A
n41	TDD	2496	2600	2496	2690	194	30	N/A	10	15	20	25	30	35	40	45	50	60	70	80	90	100
n41	TDD	2496	2600	2496	2690	194	60	N/A	10	15	20	25	30	35	40	45	50	60	70	80	90	100
n77	TDD	3300	4200	3300	4200	900	15	N/A	10	15	20	25	30	N/A	40	N/A	50	N/A	N/A	N/A	N/A	N/A
n77	TDD	3300	4200	3300	4200	900	30	N/A	10	15	20	25	30	N/A	40	N/A	50	60	70	80	90	100
n77	TDD	3300	4200	3300	4200	900	60	N/A	10	15	20	25	30	N/A	40	N/A	50	60	70	80	90	100
n78	TDD	3300	3800	3300	3800	500	15	N/A	10	15	20	25	30	N/A	40	N/A	50	N/A	N/A	N/A	N/A	N/A
n78	TDD	3300	3800	3300	3800	500	30	N/A	10	15	20	25	30	N/A	40	N/A	50	60	70	80	90	100
n78	TDD	3300	3800	3300	3800	500	60	N/A	10	15	20	25	30	N/A	40	N/A	50	60	70	80	90	100

IQ_CELLULAR_GENERATOR = MATLAB + PYTHON

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

from iq_cellular_generator import iq_cellular_generator
```

✓ 1.0s

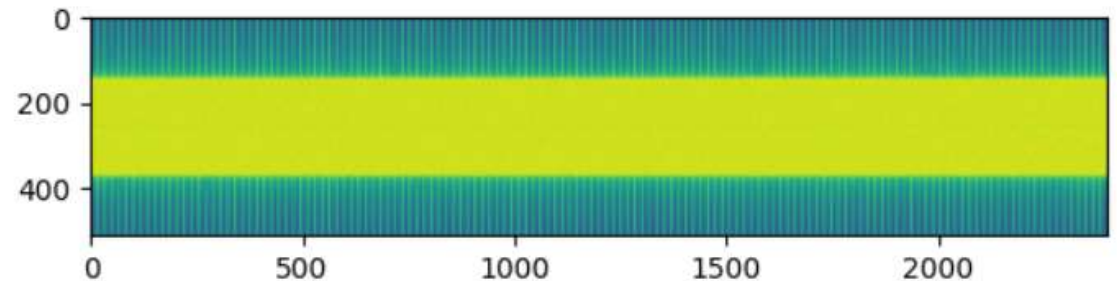
```
fc = 2.5*1e6 # Hz
B = 4e9 # Hz
dt = 1e-3 # s
type_cellular = 'LTE' # '5G'
```

✓ 0.0s

```
iq, fs, configs = iq_cellular_generator(fc, B, dt, type_cellular)
```

✓ 4.9s

```
nfft = 512
f, t, Sxx = scipy.signal.spectrogram(iq, fs, nfft=nfft, nperseg=nfft, noverlap=int(0.75*nfft))
Sxx=10*np.log10(Sxx)
plt.imshow(scipy.fft.fftshift(Sxx[:, :4000], axes=0))
```



Solution 1 : matlab.engine → Solution choisie

+ Facile d'utilisation

- Nécessite MATLAB et les toolbox d'être installés

Solution 2 : [MATLAB Compiler Toolbox](#)

+ Ne nécessite ni MATLAB ni les toolbox

- Onéreuse

AVANTIX

AVANTIX

FO RADAR

25/10/2023

© Avantix – for internal use

F0 Radar

DI (ns)	Sans Modulation	Modulations de Fréquence (MHz/μs)		Modulations de Phase Phase / Nombre de moments
10	Oui	-	-	-
100	Oui	-	-	-
1000	Oui	20	100	180° 7, 8
10000	Oui	2	10	180° 7, 8, 13, 25, 33, 39 60° 25, 30 45° 8, 35, 70
100000	Oui	0,2	1	180° 7, 8, 13, 25, 33, 39 60° 25, 30 45° 8, 35, 70
1000000	Non	0,02	0,1	-
10000000	Non	0,002	0,01	-
100000000	Non	0,0002	0,001	-
CW	Oui	-	-	-

TODO ZAB

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

Les paramètres de cette fonction seront :

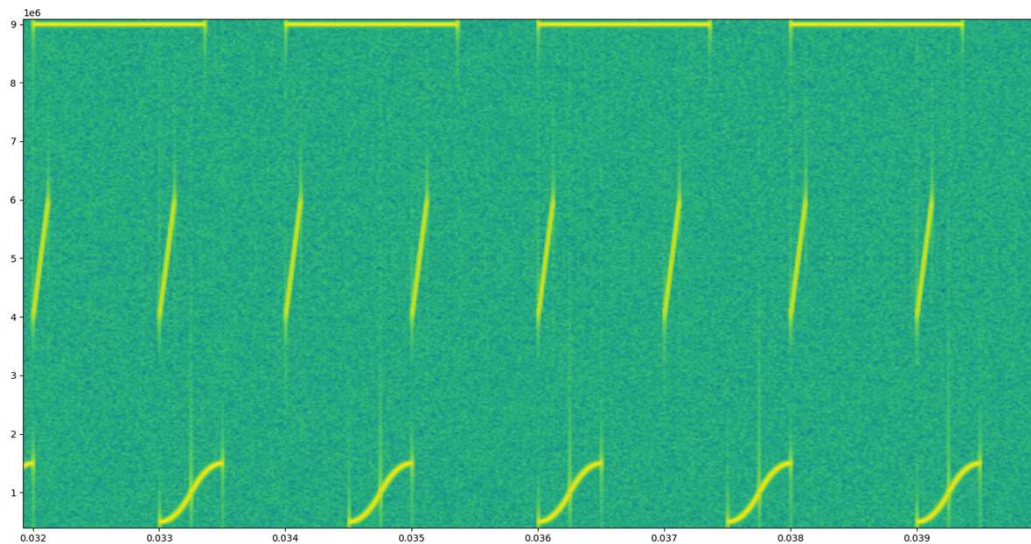
- Fréquence d'échantillonnage (Hz)
- Fréquence centrale de notre impulsion (Hz)
- Durée de l'impulsion (seconde)
- Paramètre de modulation (optionnel)
 - Modulation de fréquence (1 couple de paramètre)
 - L'évasion en fréquence (Hz)
 - L'aspect de la modulation
 - Linéaire = chirp
 - Quadratique = S-law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

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

(voir slide suivant pour exemple)

F0 Radar : Modulation en fréquence

1Mhz - S-law
5Mhz - chirp linéaire
9Mhz – Aucune modulation



```
#Generation sans mod
puls=gen_iq_radars(fs=20e6,fo=9e6,di=0.00260)
for i in range (0,len(s),int(9e4)):
    if (len(s[i:])>len(puls)):
        s[i:i+len(puls)]+=puls
    else:
        s[i:]+=puls[0:len(s[i:])]

#Generation d'un shirp
puls=gen_iq_radars(fs=20e6,fo=5e6,di=0.000120,freq_mod=(1e6,'linear'))
for i in range (0,len(s),int(2e4)):
    if (len(s[i:])>len(puls)):
        s[i:i+len(puls)]+=puls
    else:
        s[i:]+=puls[0:len(s[i:])]

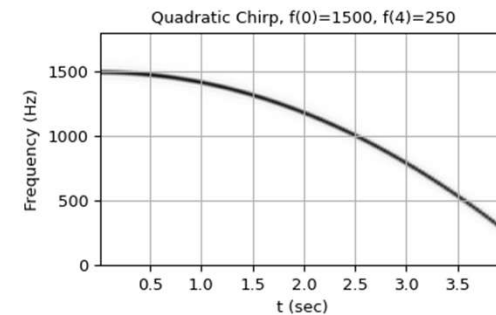
#Generation d'une s_law
puls=gen_iq_radars(fs=20e6,fo=1e6,di=0.0005,freq_mod=(0.5e6,'quadratic'))
for i in range (0,len(s),int(2e4)):
    if (len(s[i:])>len(puls)):
        s[i:i+len(puls)]+=puls
    else:
        s[i:]+=puls[0:len(s[i:])]
```

F0 Radar : Loi en S

```
elif freq_mod:
    #Extraction des paramètre de fréquence
    delta_f=freq_mod[0]
    freq_form=freq_mod[1]
    if freq_form == 'linear':
        puls=scipy.signal.chirp(t, f0=fo-delta_f, f1=fo+delta_f, t1=t[-1], method=freq_form)
    else :
        puls[0:len(puls)//2]=scipy.signal.chirp(t, f0=fo-delta_f, f1=fo, t1=t[len(puls)//2],
                                                method=freq_form,
                                                vertex_zero=True)[0:len(puls)//2]

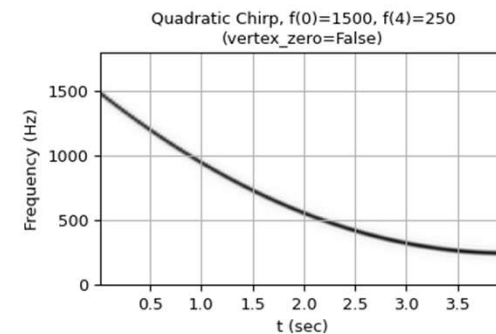
        sheak=len(t)%2

        puls[len(puls)//2:]=scipy.signal.chirp(t, f0=fo, f1=fo+delta_f, t1=t[len(puls)//2],
                                                method=freq_form,
                                                vertex_zero=False)[0:(len(puls)//2)+sheak]
    return hilbert(puls)
```



Quadratic chirp from 1500 Hz to 250 Hz (vertex of the parabolic curve of the frequency is at $t=T$):

```
>>> w = chirp(t, f0=1500, f1=250, t1=T, method='quadratic',
...          vertex_zero=False)
>>> plot_spectrogram(f'Quadratic Chirp, f(0)=1500, f({T})=250\n' +
...                  '(vertex_zero=False)', w, fs)
>>> plt.show()
```



F0 Radar : Loi en S

The theoretic pulse duration time $t_i = 2k$ was limited up to $t_i < 2k$ and the shift in time has changed its value from k up to $t_i/2$ both in order to eliminate the effect mentioned above. It is shown in Fig. 1. After taking into account modifications mentioned above the expressions (9) and (10) are as follows:

$$\theta(t) = \frac{t_i \sqrt{\Delta^2 + 4}}{2\Delta} - \left[\frac{t_i^2 (\Delta^2 + 4)}{4\Delta^2} - \left(t - \frac{t_i}{2} \right)^2 \right]^{\frac{1}{2}}, \quad t \in \langle 0, t_i \rangle, \quad (11)$$

$$f(t) = \frac{t - \frac{t_i}{2}}{\sqrt{\frac{t_i^2 (\Delta^2 + 4)}{4\Delta^2} - \left(t - \frac{t_i}{2} \right)^2}}, \quad t \in \langle 0, t_i \rangle. \quad (12)$$

That is why the NLFM signal model under condition of chosen parameters: A , t_i and Δ is finally described by the formula as follows:

$$s(t) = \text{Arect} \left(\frac{t - t_i/2}{t_i} \right) \exp \left(j 2\pi \left(\frac{t_i \sqrt{\Delta^2 + 4}}{2\Delta} - \left(\frac{t_i^2 (\Delta^2 + 4)}{4\Delta^2} - \left(t - \frac{t_i}{2} \right)^2 \right)^{\frac{1}{2}} \right) \right). \quad (13)$$

The instantaneous frequency change interval Δ deter-

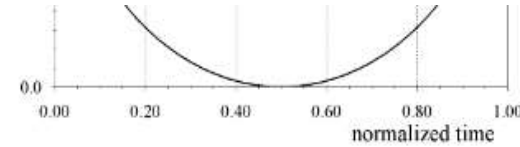


Fig. 2. Phase characteristic of the NLFM signal.

The frequency characteristic of the NLFM signal versus normalized time is presented in Fig. 3.

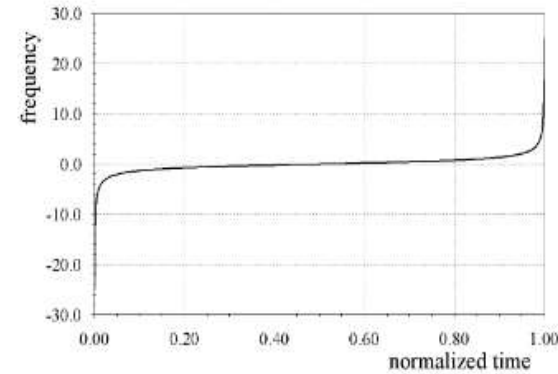


Fig. 3. Frequency characteristic of the NLFM signal.

Source : <http://przyrbwn.icm.edu.pl/APP/PDF/116/a116z327.pdf>

AVANTIX

FO Satellite

25/10/2023

© Avantix – for internal use

F0 satellite

Satellite Toolbox : <https://fr.mathworks.com/products/satellite-communications.html>

The toolbox offers these standards-based waveforms:

- Consultative Committee for Space Data Systems (CCSDS) Telecommand (TC)
- CCSDS Telemetry (TM)
- Digital Video Broadcasting Satellite Second Generation (DVB-S2)
- Digital Video Broadcasting Satellite Second Generation extended (DVB-S2X)
- Global Positioning System (GPS)

AVANTIX

MERCI

AVANTIX.NET