Radio logicielle

J.-M Friedt

FEMTO-ST/département temps-fréquence, Besançon

jmfriedt@femto-st.fr

transparents à `jmfriedt.free.fr`

12 mars 2022

Qui suis-je ?

- Études de physique et informatique, thèse en sciences pour l'ingénieur (2000)
- Ingénieur chez SENSeOR (2006–2014) (RADARs à courte portée pour mesure de capteurs passifs interrogeables sans-fil)
- Enseignant-chercheur à l'Université de Franche-Comté à Besançon (France) depuis 2014
- Trois mois de visite en professeur invité à Sendai (Japon) à l'Univ. de Tohoku (RADAR de sol et RADAR passif)
- Co-organisateur de la conférence European GNU Radio Days [1] depuis 2018

1. https://www.youtube.com/channel/UCFzddPoztcHLuwFWRPJTNrQ

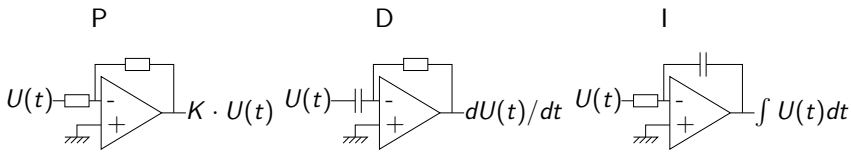
Manuscrits disponibles à `http://jmfriedt.free.fr`

1. J.-M Friedt, G. Goavec-Merou, *La réception radiofréquence définie par logiciel (Software Defined Radio – SDR)*, GNU/Linux Magazine France **153** (Octobre 2012), pp.4-33
2. J.-M Friedt, *La réception de signaux venus de l'espace par récepteur de télévision numérique terrestre* OpenSilicium **13** (Dec 2014/Jan-Fev 2015)
3. J.-M Friedt, G. Cabodevila, *Exploitation de signaux des satellites GPS reçus par récepteur de télévision numérique terrestre DVB-T*, OpenSilicium **15**, Juillet-Sept. 2015
4. **J.-M Friedt, C. Eustache, É. Carry, E. Rubiola, Software defined radio decoding of DCF77 : time and frequency dissemination with a sound card, Radio Science 53 (1), pp.48-61 (Jan 2018)**
5. J.-M Friedt, *Radio Data System (RDS) – analyse du canal numérique transmis par les stations radio FM commerciales, introduction aux codes correcteurs d'erreur*, GNU/Linux Magazine France **204** (Mai 2017)
6. J.-M Friedt, *RADAR passif par intercorrélation de signaux acquis par deux récepteurs de télévision numérique terrestre*, GNU/Linux Magazine France **212** pp.36- (Fév. 2018)
7. J.-M Friedt, *Décodage d'images numériques issues de satellites météorologiques en orbite basse : le protocole LRPT de Meteor-M2 (en 3 parties)*, **226–228** (Mai–Aout 2019)
8. G. Goavec-Merou, J.-M Friedt, F. Meyer, *Leurrage du GPS par radio logicielle*, MISC HS (Fév. 2019)
9. J.-M Friedt, *Quelques éléments théoriques pour aborder la radio logicielle*, GNU/Linux Magazine France **224** (Mars 2019)
10. J.-M Friedt, *Matériel pour la radio logicielle*, GNU/Linux Magazine France **224** (Mars 2019)
11. J.-M Friedt, W. Feng, *Mesure fine de déplacement par RADAR interférométrique à synthèse d'ouverture (InSAR) par radio logicielle (3/3)*, GNU/Linux Magazine France **244** (Jan. 2021)
12. J.-M Friedt, W. Feng, *Analyse et réalisation d'un RADAR à synthèse d'ouverture (SAR) par radio logicielle (2/3)*, GNU/Linux Magazine France **242** (Nov. 2020)
13. J.-M Friedt, W. Feng, *Analyse et réalisation d'un RADAR à bruit par radio logicielle (1/3)*, GNU/Linux Magazine France **240** (Sept. 2020)
14. **W. Feng, J.-M Friedt, G. Goavec-Merou, F. Meyer, Software Defined Radio Implemented GPS Spoofing and Its Computationally Efficient Detection and Suppression IEEE Aerospace and Electronic Systems Magazine 36 (3), Mars 2021**
15. J.-M Friedt, *Décodage par radio logicielle du VOR pour le positionnement sans GPS*, Hackable **36** (Jan.-Fév.-Mar. 2021)
16. **W. Feng, J.-M Friedt, SDR Implemented Ground-based Interferometric Radar for Displacement Measurement, IEEE Trans. on Instrumentation and Measurement 70, 8502218 (2021), doi : 10.1109/TIM.2021.3069805**
17. **W. Feng, J.-M. Friedt, P. Wan, SDR-implemented passive bistatic SAR system using Sentinel-1 signal and its experiment results, soumis MDPI Remote Sensing (2021)**

Matériel v.s logiciel

Passage de l'implémentation **matérielle** à *logicielle* de centrales inertielles (1950s) [2]

- **Stabilité** : un algorithme numériques ne dérive pas dans le temps ou ne présente pas de variation avec l'environnement
- **Flexibilité** : une même plateforme matérielle supporte une multitude d'algorithmes
- **Reconfigurabilité** : capacité à modifier les paramètres de fonctionement



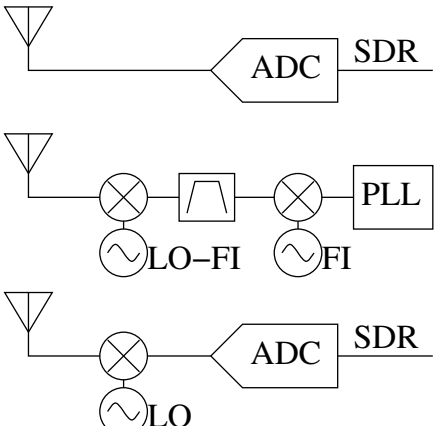
$$s_n = K \cdot \varepsilon_n + K_d \frac{\varepsilon_n - \varepsilon_{n-1}}{T_s} + \sum \varepsilon_n \cdot T_s$$

$$\Leftrightarrow s_n = s_{n-1} + K \cdot (\varepsilon_n - \varepsilon_{n-1}) + K_d \frac{\varepsilon_n - 2\varepsilon_{n-1} + \varepsilon_{n-2}}{T_s} + T_s \varepsilon_n$$

- Excellente **stabilité** du numérique à long terme (pas de dérive), mais **bande passante limitée** (f_s à x00 MHz mais coefficients de FIR)

2. D.A. Mindell, *Digital Apollo : human and machine in spaceflight*, MIT Press (2011)

Récepteur radiofréquence matériel v.s logiciel



- Ce qu'on veut : solution tout logiciel, avec une antenne et un convertisseur analogique numérique.
  Inconvénients : dynamique limitée et l'antenne se comporte comme filtre passe-bande
- Solution **matérielle** : une seule fonction, spécifiée à la conception du circuit (e.g. réception FM)
- Solution **pratique** : une première transposition de fréquence analogique avec amplificateur à gain variable, puis numérisation en bande de base avant de démoduler numériquement le signal.

"In addition to easing the scheduling and configuration burden, an autonomous radio also will **gracefully handle unpredictable or anomalous events**. For example, during entry, descent, and landing (EDL), a spacecraft can undergo large Doppler swings caused by rocket firings, parachute openings, backshell ejection, and a bouncing landing on the surface. Even when all scheduled events occur successfully, there may be Doppler uncertainty due to unpredictable properties of the atmosphere. Ideally, the communication link should operate whether or not each of the EDL events is successful, but the uncertainties involved typically lead to liberal link margins—for example, the **Mars Exploration Rovers** observed link margins that sometimes exceeded 10 dB. An autonomous radio could substantially reduce this margin because it would handle any Doppler swing nearly optimally.

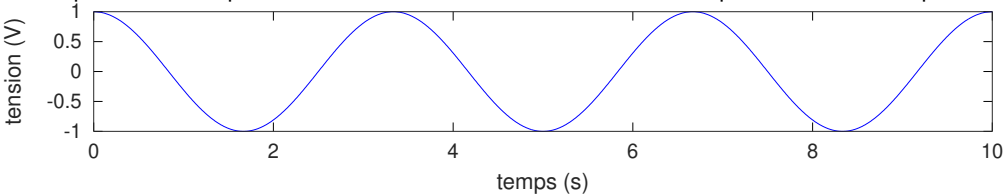
Unfortunately, such flexible technology is not available on NASA's currently flying missions. In perhaps the most glaring example of this, NASA engineers discovered in 2000 that a receiver aboard Cassini, launched in 1997, would fail during the Huygens probe descent onto Titan because it did **not properly account for the Doppler profile of the probe**. Increasing the loop bandwidth of the synchronization loops would have easily fixed the problem, but, unfortunately, these loop bandwidths were hard-wired to fixed values on the spacecraft. With superior engineering and enormous dedication, NASA and the European Space Agency were still able to save the mission by slightly altering the original trajectory, but this solution required forming a large and expensive international recovery team to find the appropriate recommendations on how to overcome the radio's severe limitations."

Échantillonnage en temps et niveaux discrets

- échantillonnage en **niveaux** discrets
  - $valeur = \frac{V}{V_{ref}} \times (2^N - 1)$
  - dynamique limitée par la plus petite variation de tension détectable $\Delta V = V_{ref}/(2^N - 1)$
  - en radiofréquence : puissance $20 \log_{10}(\Delta V)$ vaut 48 dB sur 8 bits, 60 dB sur 10 bits, 72 dB sur 12 bits et 96 dB sur 16 bits
- échantillonnage en **temps** discret
  - fréquence d'échantillonnage f_s (période d'échantillonnage $1/f_s$)
  - hypothèse de périodicité du spectre
  - extension du spectre de $-f_s/2$ à $+f_s/2$
  - f_s connu $\Rightarrow$ on décide de nommer $f_s/2$ (fréquence de Nyquist) à 1 et d'exprimer toutes les fréquences f comme fréquences normalisées

$$f/(f_s/2) = 2 \times f/f_s \in [-1 : 1]$$

  - acquisition sur N points et transformée de Fourier discrète sur N points : résolution spectrale f_s/N

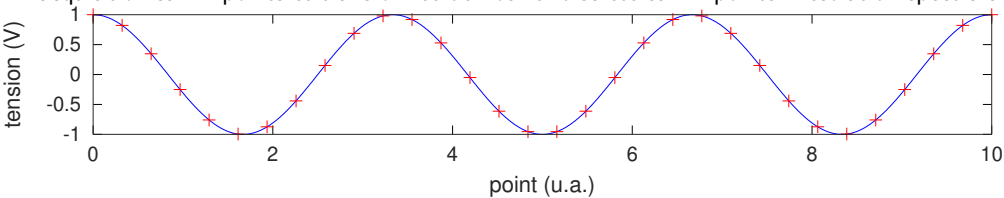


Échantillonnage en temps et niveaux discrets

- échantillonnage en **niveaux** discrets
  - $valeur = \frac{V}{V_{ref}} \times (2^N - 1)$
  - dynamique limitée par la plus petite variation de tension détectable $\Delta V = V_{ref}/(2^N - 1)$
  - en radiofréquence : puissance $20 \log_{10}(\Delta V)$ vaut 48 dB sur 8 bits, 60 dB sur 10 bits, 72 dB sur 12 bits et 96 dB sur 16 bits
- échantillonnage en **temps** discret
  - fréquence d'échantillonnage f_s (période d'échantillonnage $1/f_s$)
  - hypothèse de périodicité du spectre
  - extension du spectre de $-f_s/2$ à $+f_s/2$
  - f_s connu $\Rightarrow$ on décide de nommer $f_s/2$ (fréquence de Nyquist) à 1 et d'exprimer toutes les fréquences f comme fréquences normalisées

$$f/(f_s/2) = 2 \times f/f_s \in [-1 : 1]$$

  - acquisition sur N points et transformée de Fourier discrète sur N points : résolution spectrale f_s/N

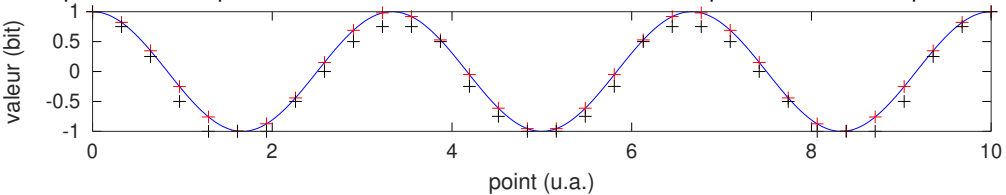


Échantillonnage en temps et niveaux discrets

- échantillonnage en **niveaux** discrets
  - $valeur = \frac{V}{V_{ref}} \times (2^N - 1)$
  - dynamique limitée par la plus petite variation de tension détectable $\Delta V = V_{ref}/(2^N - 1)$
  - en radiofréquence : puissance $20 \log_{10}(\Delta V)$ vaut 48 dB sur 8 bits, 60 dB sur 10 bits, 72 dB sur 12 bits et 96 dB sur 16 bits
- échantillonnage en **temps** discret
  - fréquence d'échantillonnage f_s (période d'échantillonnage $1/f_s$)
  - hypothèse de périodicité du spectre
  - extension du spectre de $-f_s/2$ à $+f_s/2$
  - f_s connu $\Rightarrow$ on décide de nommer $f_s/2$ (fréquence de Nyquist) à 1 et d'exprimer toutes les fréquences f comme fréquences normalisées

$$f/(f_s/2) = 2 \times f/f_s \in [-1 : 1]$$

  - acquisition sur N points et transformée de Fourier discrète sur N points : résolution spectrale f_s/N

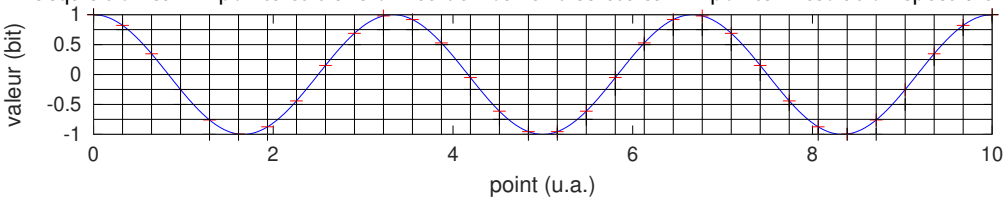


Échantillonnage en temps et niveaux discrets

- échantillonnage en **niveaux** discrets
  - $valeur = \frac{V}{V_{ref}} \times (2^N - 1)$
  - dynamique limitée par la plus petite variation de tension détectable $\Delta V = V_{ref}/(2^N - 1)$
  - en radiofréquence : puissance $20 \log_{10}(\Delta V)$ vaut 48 dB sur 8 bits, 60 dB sur 10 bits, 72 dB sur 12 bits et 96 dB sur 16 bits
- échantillonnage en **temps** discret
  - fréquence d'échantillonnage f_s (période d'échantillonnage $1/f_s$)
  - hypothèse de périodicité du spectre
  - extension du spectre de $-f_s/2$ à $+f_s/2$
  - f_s connu $\Rightarrow$ on décide de nommer $f_s/2$ (fréquence de Nyquist) à 1 et d'exprimer toutes les fréquences f comme fréquences normalisées

$$f/(f_s/2) = 2 \times f/f_s \in [-1 : 1]$$

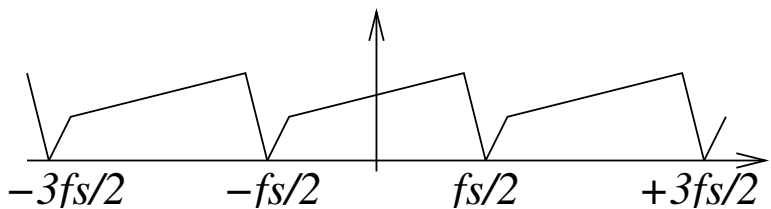
  - acquisition sur N points et transformée de Fourier discrète sur N points : résolution spectrale f_s/N



Transformée de Fourier discrète

Axe des abscisses et repliement spectral

- Le signal continu est échantillonné périodiquement au rythme de f_s points/s
- théorème d'échantillonnage : la plus haute composante spectrale est $f_s/2$
- signal complexe : partie positive et négative du spectre ne sont pas symétriques (signal réel : $X(-f) = X^*(f) \Leftrightarrow |X(-f)| = |X(f)|$)
- Hypothèse de périodicité du spectre
- $f_s/2$ est la **fréquence de Nyquist**, bornes de la **bande de base** centrée sur 0 Hz

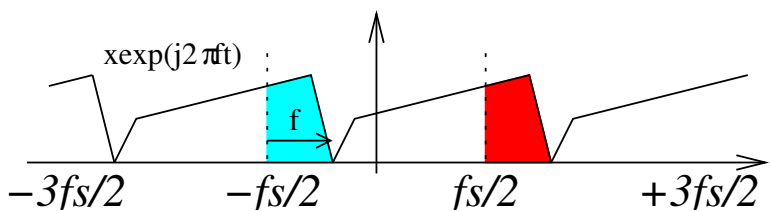


Rappel : $\exp(j\omega_1 t) \cdot \exp(j\omega_2 t) = \exp(j(\omega_1 + \omega_2)t)$

Transformée de Fourier discrète

Axe des abscisses et repliement spectral

- Le signal continu est échantillonné périodiquement au rythme de f_s points/s
- théorème d'échantillonnage : la plus haute composante spectrale est $f_s/2$
- signal complexe : partie positive et négative du spectre ne sont pas symétriques (signal réel : $X(-f) = X^*(f) \Leftrightarrow |X(-f)| = |X(f)|$)
- Hypothèse de périodicité du spectre
- $f_s/2$ est la **fréquence de Nyquist**, bornes de la **bande de base** centrée sur 0 Hz



Rappel : $\exp(j\omega_1 t) \cdot \exp(j\omega_2 t) = \exp(j(\omega_1 + \omega_2)t)$

Mélangeur I, Q

- mélanger pour transposer une fréquence :

$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \cos(\omega_2 t) \propto A_1 \cdot A_2 \cdot \sin((\omega_1 + \omega_2)t + \varphi) + \sin((\omega_1 - \omega_2)t + \varphi)$$

$$\propto A_1 \cdot A_2 \cdot \sin(\varphi) \text{ si } \omega_1 = \omega_2$$

- coefficients **I**(dentité) et **Q**(uadrature) : $A \simeq \sqrt{I^2 + Q^2}$, $\varphi \simeq \arctan(Q/I)$
- un signal réel (porteuse modulée) est devenu complexe en sortie du démodulateur

Mélangeur I, Q

- mélanger pour transposer une fréquence :

$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \cos(\omega_2 t) \propto A_1 \cdot A_2 \cdot \sin((\omega_1 + \omega_2)t + \varphi) + \sin((\omega_1 - \omega_2)t + \varphi)$$

$$\propto A_1 \cdot A_2 \cdot \sin(\varphi) \text{ si } \omega_1 = \omega_2$$

- le terme “somme de fréquences” est éliminé par filtre passe-bas

- coefficients **I**(dentité) et **Q**(uadrature) : $A \simeq \sqrt{I^2 + Q^2}$, $\varphi \simeq \arctan(Q/I)$
- un signal réel (porteuse modulée) est devenu complexe en sortie du démodulateur

Mélangeur I, Q

- mélanger pour transposer une fréquence :

$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \cos(\omega_2 t) \propto A_1 \cdot A_2 \cdot \sin((\omega_1 + \omega_2)t + \varphi) + \sin((\omega_1 - \omega_2)t + \varphi)$$

$$\propto A_1 \cdot A_2 \cdot \sin(\varphi) \text{ si } \omega_1 = \omega_2$$

- le terme "somme de fréquences" est éliminé par filtre passe-bas
- si $\varphi = 0$, la sortie est toujours nulle quelquesoient A_1, A_2
- utiliser la version transposée de 90^o du signal de référence : cos $\rightarrow$ sin

$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \sin(\omega_2 t) \propto A_1 \cdot A_2 \cdot \cos((\omega_1 + \omega_2)t + \varphi) + \cos((\omega_1 - \omega_2)t + \varphi)$$

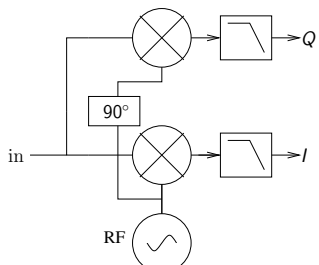
$$\propto A_1 \cdot A_2 \cdot \cos(\varphi) \text{ si } \omega_1 = \omega_2$$

- coefficients **I**(dentité) et **Q**(uadrature) : $A \simeq \sqrt{I^2 + Q^2}$, $\varphi \simeq \arctan(Q/I)$
- un signal réel (porteuse modulée) est devenu complexe en sortie du démodulateur

Mélangeur I, Q

- mélanger pour transposer une fréquence :

$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \cos(\omega_2 t) \propto A_1 \cdot A_2 \cdot \sin(\varphi) \text{ si } \omega_1 = \omega_2$$



$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \sin(\omega_2 t) \propto A_1 \cdot A_2 \cdot \cos(\varphi) \text{ si } \omega_1 = \omega_2$$

- coefficients **I**(dentité) et **Q**(uadrature) : $A \simeq \sqrt{I^2 + Q^2}$, $\varphi \simeq \arctan(Q/I)$
- un signal réel (porteuse modulée) est devenu complexe en sortie du démodulateur

Principe du récepteur : coefficients I, Q

Pourquoi traitons nous des nombres complexes ?

- un signal périodique est caractérisé par 3 grandeurs :

$$s(t) = \underbrace{A(t)}_{AM} \cdot \cos\left(\underbrace{\omega(t)}_{FM} \cdot t + \underbrace{\varphi(t)}_{PM}\right)$$

- puissance et {fréquence,phase} ne sauraient être extraites d'une unique mesure
- transposition par mélangeur de l'oscillateur local et sa copie en quadrature ($\cos(x + 90°) = \sin$)
- Problème : la quadrature n'est pas idéale et les gains sont différents

$$\begin{cases} I_{expe} = A \cdot \cos(\omega t + \varphi) \\ Q_{expe} = (A + \varepsilon) \cdot \sin(\omega t) \end{cases} \quad v.s. \quad \begin{cases} I_{ideal} = A \cdot \cos(\omega t) \\ Q_{ideal} = A \cdot \sin(\omega t) \end{cases}$$

- $I_{expe} = A\cos(\omega t)\cos(\varphi) - A\sin(\omega t)\sin(\varphi) = I_{ideal}\cos(\varphi) - Q_{ideal}\sin(\varphi)$
- $Q_{expe} = (1 + \varepsilon/A)Q_{ideal}$: combinaisons linéaires des I et Q

Analogique v.s numérique : I, Q [3]

Sentinel-1 Level 1 Detailed Algorithm Definition

4.1 Raw Data Analysis

Raw data analysis is required in order to perform corrections of the I and Q channels of the raw signal data. The classical raw data correction (applied for instance in the case of ENVISAT-ASAR and RADARSAT-2) involves (see also Section 9.2):

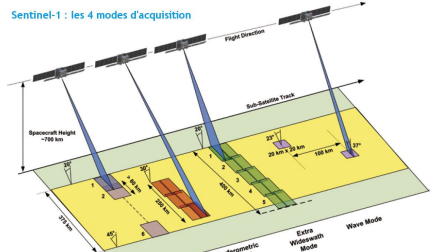
- I/Q bias removal
- I/Q gain imbalance correction
- I/Q non-orthogonality correction

For Sentinel-1 however, the instrument's receive module performs the demodulation in the digital domain, therefore the I/Q gain imbalance and I/Q non-orthogonality corrections are no longer necessary.

The raw data analysis necessary for the raw data correction of ASAR data is defined in [R-6]. Since the IPF also supports the processing of ASAR data, for completeness, the ASAR raw data analysis scheme is reproduced in this section.

Even though for Sentinel-1 the I/Q gain imbalance and the I/Q non-orthogonality corrections are not necessary, they will be made available optionally, using configuration input parameters. Irrespective to the correction flag though, the Raw

Sentinel-1 : les 4 modes d'acquisition



Analogique v.s numérique : I, Q [3]

Sentinel-1 Level 1 Detailed Algorithm Definition

4.1 Raw Data Analysis

Raw data analysis is required in order to perform corrections of the I and Q channels of the raw signal data. The classical raw data correction (applied for instance in the case of ENVISAT-ASAR and RADARSAT-2) involves (see also Section 9.2):

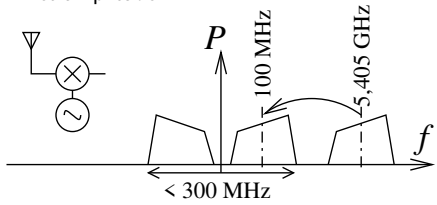
- I/Q bias removal
- I/Q gain imbalance correction
- I/Q non-orthogonality correction

For Sentinel-1 however, the instrument's receive module performs the demodulation in the digital domain, therefore the I/Q gain imbalance and I/Q non-orthogonality corrections are no longer necessary.

The raw data analysis necessary for the raw data correction of ASAR data is defined in [R-6]. Since the IPF also supports the processing of ASAR data, for completeness, the ASAR raw data analysis scheme is reproduced in this section.

Even though for Sentinel-1 the I/Q gain imbalance and the I/Q non-orthogonality corrections are not necessary, they will be made available optionally, using configuration input parameters. Irrespective to the correction flag though, the Raw

- Première transposition radiofréquence : 5,405 GHz → 100 MHz par mélange avec oscillateur local à 5,305 MHz (mélangeur unique matériel, échantillonnage du signal réel)
- Numérisation à 300 MS/s : spectre ±150 MHz
- Seconde transposition numérique "parfaite" pour ramener signal en bande de base : `t=[0:longueur-1]/fs; lo=exp(j*2*pi*freq*t); res=signal.*lo;`
- `lo` numérique est "parfait" en quadrature et amplitude



Analogique v.s numérique : I, Q [3]

Sentinel-1 Level 1 Detailed Algorithm Definition

4.1 Raw Data Analysis

Raw data analysis is required in order to perform corrections of the I and Q channels of the raw signal data. The classical raw data correction (applied for instance in the case of ENVISAT-ASAR and RADARSAT-2) involves (see also Section 9.2):

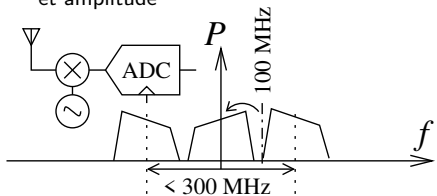
- I/Q bias removal
- I/Q gain imbalance correction
- I/Q non-orthogonality correction

For Sentinel-1 however, the instrument's receive module performs the demodulation in the digital domain, therefore the I/Q gain imbalance and I/Q non-orthogonality corrections are no longer necessary.

The raw data analysis necessary for the raw data correction of ASAR data is defined in [R-6]. Since the IPF also supports the processing of ASAR data, for completeness, the ASAR raw data analysis scheme is reproduced in this section.

Even though for Sentinel-1 the I/Q gain imbalance and the I/Q non-orthogonality corrections are not necessary, they will be made available optionally, using configuration input parameters. Irrespective to the correction flag though, the Raw

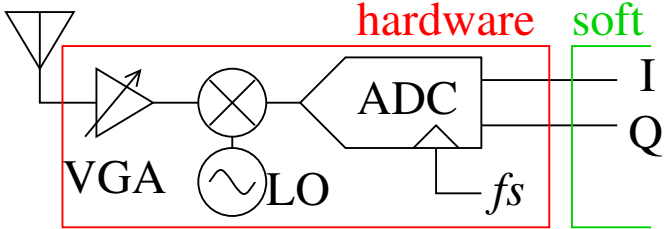
- Première transposition radiofréquence : 5,405 GHz $\rightarrow$ 100 MHz par mélange avec oscillateur local à 5,305 MHz (mélangeur unique matériel, échantillonnage du signal réel)
- Numérisation à 300 MS/s : spectre $\pm$150 MHz
- Seconde transposition numérique "parfaite" pour ramener signal en bande de base : `t=[0:longueur-1]/fs; lo=exp(j*2*pi*freq*t); res=signal.*lo;`
- `lo` numérique est "parfait" en quadrature et amplitude



Architecture d'un récepteur à conversion directe

Récepteur homodyne ou *zero-IF*

- LO transpose signal RF en bande de base : élimine la porteuse
- VGA : gain de réception pour détecter le signal sans le saturer
- f_s détermine la **bande passante** de mesure : la transformée de Fourier s'étend de $-f_s/2$ à $+f_s/2$
- mélange avec LO et sa copie en quadrature génère le complexe I+jQ

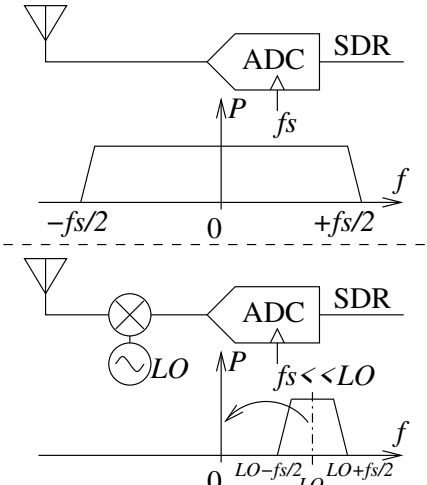


Débit de données :

- 4 octets/conversion (représentation flottante)
- 2 conversions/échantillon (I, Q)
- f_s conversions/seconde

Exemple : f_s=2 Méchantillons/s $\Rightarrow$ 16 MB/s ou 1 GB/minute

Architecture d'un récepteur à conversion directe

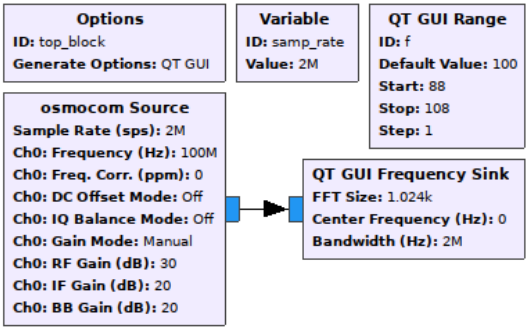


Transposition de fréquence par LO (*Local Oscillator*) avant échantillonnage : perte partielle de flexibilité par rapport à conversion directe mais

- bénéfice en terme de **débit** et taille de stockage (MB/s au lieu de 100-1000 MB/s)
- bénéfice en terme de **réjection** de sources puissantes interférant avec réception d'un signal faible (dynamique ADC : $20\log_{10}(\text{bit})=6\times\text{bits}=48$ dB pour 8 bits)
- ADC lents ont une meilleure **résolution** que ADC radiofréquences

Aspects logiciels : GNURadio

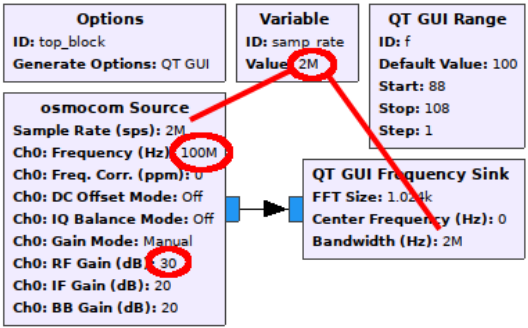
- Aspect logiciel : GNURadio pour le traitement numérique du signal
- Logiciel libre disponible sous GNU/Linux, MacOS & MS-Windows
- Ensemble de bibliothèques C++ connectées par Python ou C++
- Générateur de code python : `gnuradio-companion`
- Opensource : **ajouter ses blocs**




```
from PyQt5 import Qt
from gnuradio import gr
from gnuradio import qtgui
from gnuradio.filter import firdes
from gnuradio.qtgui import Range, RangeWidget
import osmosdr

class top_block(gr.top_block, Qt.QWidget):
    def __init__(self):
        gr.top_block.__init__(self, "Top Block")
        Qt.QWidget.__init__(self)
[...]
        # Variables
        self.samp_rate = samp_rate = 2e6
        self.f = f = 100
        # Blocks
        self._f_range = Range(88, 108, 1, 100, 200)
        self._f_win = RangeWidget(self._f_range, self.set_f, "f", "→
            ↪counter_slider", float)
        self.top_layout.addWidget(self._f_win)
        self.qtgui_freq_sink_x_0=qtgui.freq_sink_c(
          1024, #size
          firdes.WIN_BLACKMAN_hARRIS, #wintype
          0, samp_rate, "", 1 #fc, bw, name, inputs
        )
[... configuration du freq sink]
        self.osmosdr_src_0 = osmosdr.source( args="numchan=" + str(1) + →
            ↪" " + '' )
        self.osmosdr_src_0.set_sample_rate(samp_rate)
        self.osmosdr_src_0.set_center_freq(f*1e6, 0)
[...]
        self.connect((self.osmosdr_src_0, 0), (self.qtgui_freq_sink_x_0 ,→
            ↪ 0))
[...]
def main(top_block_cls=top_block, options=None):
[...]
```

Aspects logiciels : GNURadio

- Aspect logiciel : GNURadio pour le traitement numérique du signal
- Logiciel libre disponible sous GNU/Linux, MacOS & MS-Windows
- Ensemble de bibliothèques C++ connectées par Python ou C++
- Générateur de code python : `gnuradio-companion`
- Opensource : **ajouter ses blocs**



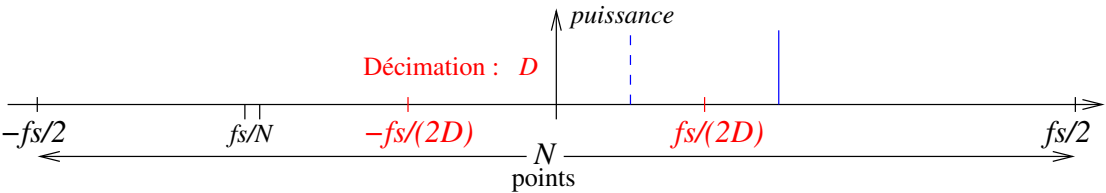

```
from PyQt5 import Qt
from gnuradio import gr
from gnuradio import qtgui
from gnuradio.filter import firdes
from gnuradio.qtgui import Range, RangeWidget
import osmosdr

class top_block(gr.top_block, Qt.QWidget):
    def __init__(self):
        gr.top_block.__init__(self, "Top Block")
        Qt.QWidget.__init__(self)
[...]
        # Variables
        self.samp_rate = samp_rate = 2e6
        self.f = f = 100
        # Blocks
        self._f_range = Range(88, 108, 1, 100, 200)
        self._f_win = RangeWidget(self._f_range, self.set_f, "f", "→
            ↪counter_slider", float)
        self.top_layout.addWidget(self._f_win)
        self.qtgui_freq_sink_x_0=qtgui.freq_sink_c(
          1024, #size
          firdes.WIN_BLACKMAN_hARRIS, #wintype
          0, samp_rate, "", 1 #fc, bw, name, inputs
        )
[... configuration du freq sink]
        self.osmosdr_src_0 = osmosdr.source( args="numchan=" + str(1) + →
            ↪" " + '' )
        self.osmosdr_src_0.set_sample_rate(samp_rate)
        self.osmosdr_src_0.set_center_freq(f*1e6, 0)
[...]
        self.connect((self.osmosdr_src_0, 0), (self.qtgui_freq_sink_x_0,→
            ↪ 0))
[...]
def main(top_block_cls=top_block, options=None):
[...]
```

Implémentation logicielle de quelques blocs de traitement

Analogique	Numérique
Redresseur (démod. AM)	valeur absolue
Filtre (passe bas)	FIR : $y_n = \sum b_k x_{n-k}$
Mélangeur	$s_n \leftarrow s_n \cdot sin(n/f_s),\ n \in \mathbb{N}$

Mais aussi des mises en œuvre d'algorithmes uniques au numérique (démodulateur FM)

Réduction du débit de données : la **décimation**. Décimer de D permet de faire une homothétie sur l'axe des abscisses

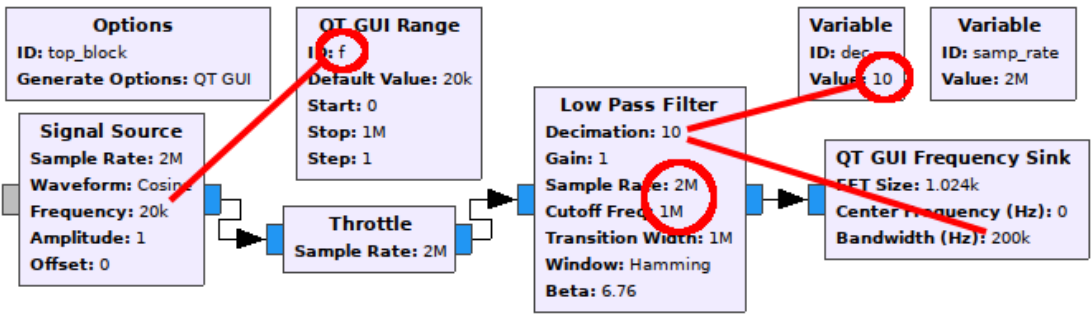


Tout traitement spectral **nécessite de maîtriser la fréquence d'échantillonnage** f_s/D : le spectre s'étend de $-f_s/(2D)$ à $+f_s/(2D)$

GNURadio comme outil de prototypage rapide

Mise en œuvre simple des concepts de traitement numérique du signal en temps discret.

Exemple : repliement spectral

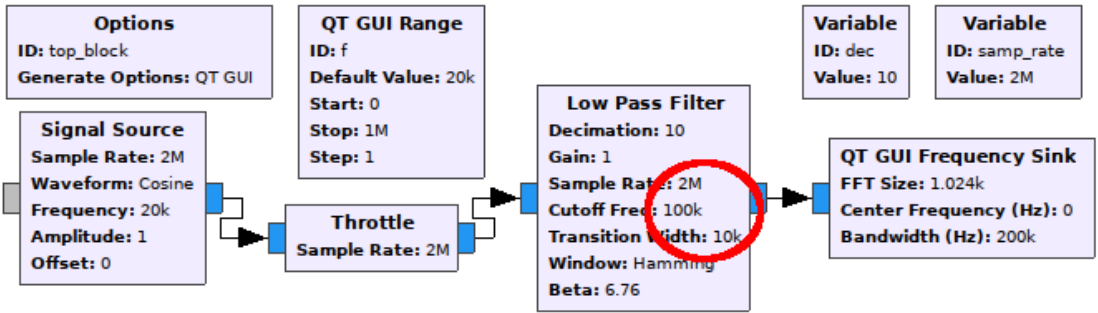


Si nous ne prenons pas soin de filtrer avant décimation, il y aura repliement dans la bande de base.

GNURadio comme outil de prototypage rapide

Mise en œuvre simple des concepts de traitement numérique du signal en temps discret.

Exemple : repliement spectral



Si nous ne prenons pas soin de filtrer avant décimation, il y aura repliement dans la bande de base.

Résolution du filtre

- Transformée de Fourier bijective : N points temporels $\leftrightarrow$ N points fréquentiels
- Une transformée de Fourier sur N points échantillonnés à f_s présente une **résolution spectrale** de f_s/N
- Un filtre FIR sur N points présente la même résolution spectrale
- Demander une bande de transition δf excessivement fine se traduit par un **nombre énorme de coefficients**

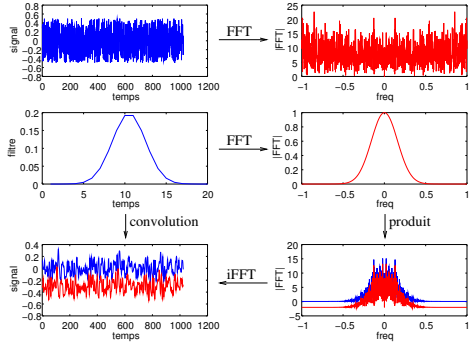
$$\delta f = f_s/N \Leftrightarrow N = f_s/\delta f$$

- $y_n = \sum b_k \cdot x_{n-k}$ nécessite N **multiplications** pour chacun des N coefficients : N^2
- $FFT(y) = FFT(b) \cdot FFT(x) \Leftrightarrow y = iFFT(FFT(b) \cdot FFT(x))$ avec "seulement" $N \log_2(N)$ opérations
- Si un filtre fin est nécessaire, cascader les filtres

Exemple : filtre de 30 Hz de transition sur un échantillonnage à 192 kHz

- $192000/30 = 6400$ coefficients ou $N \log_2(N) \simeq 81000$ opérations
- décimation de 50 (coupure à $f_s/100$ avec une largeur de $f_s/200$) : 200 coefficients pour 1500 opérations
- aprés décimation, $f_s/50 = 3840$ Hz et $3840/30 = 128$ coefficients, pour 900 opérations
- $1500 + 900 \ll 81000$

Tester avec `Filter Design Tool` sous GNU/Linux



Démonstration du théormème de convolution

$$\left.\begin{array}{l} conv(x,y)(\tau) = \int x(t) \cdot y(\tau - t) dt \\ TF(x)(f) = \int x(t) e^{j2\pi ft} dt \end{array}\right\} \Rightarrow$$

$TF(conv(x,y)) = \iint x(t) y(\tau - t) dt e^{j2\pi f\tau} d\tau$

on définit $s = \tau - t \Rightarrow ds = d\tau$ et

$TF(conv(x,y)) = \iint x(t) y(s) dt e^{j2\pi f(t+s)} ds$

$\Leftrightarrow TF(conv(x,y)) =$

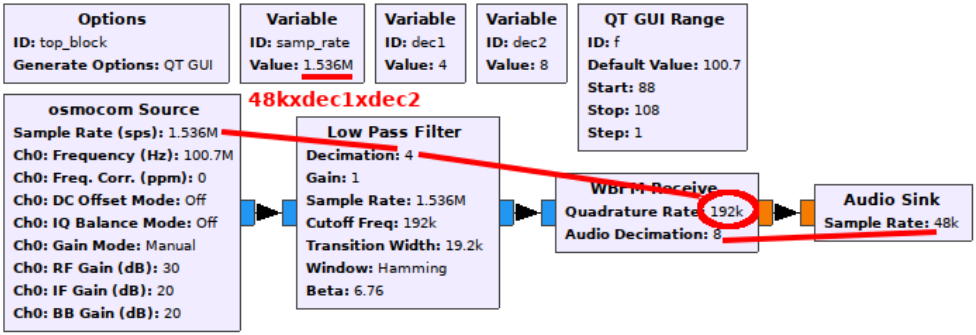
$\int x(t) e^{j2\pi ft} dt \times \int y(s) e^{j2\pi fs} ds$

$\Leftrightarrow TF(conv(x,y)) = TF(x) \cdot TF(y)$

Importance de la gestion du flux de données

- Plus on avance dans la chaîne de traitement, plus la bande passante se réduit
- $\Rightarrow$ décimation pour réduire le débit de données à traiter
- on ne peut pas créer d'information au cours du traitement, uniquement en perdre (porteuse + modulation $\rightarrow$ bande de base $\rightarrow$ contenu informatif)
- exemple : FM commerciale nécessite 250 kHz de bande pour encoder du son à 16 kHz (gauche + droite) + RDS

Réduire le débit de données pour réduire la charge sur le processeur, ou permettre des calculs de plus en plus complexes.

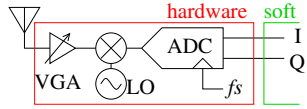


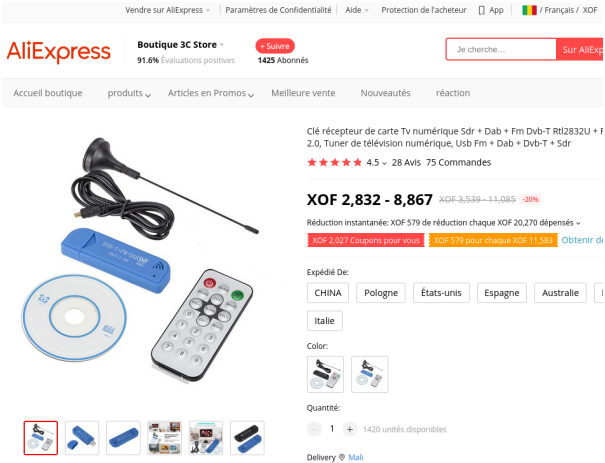
Conclusion sur la gestion du flux de données

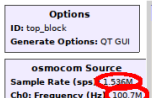
- Mémoriser l'évolution du débit de données au cours du traitement
- Ajuster le débit de données de la source si la cible impose sa fréquence d'échantillonnage à l'arrivée
- Filtre passe bas pour empêcher le repliement spectral
- Penser aux variables pour paramétrer les facteurs de décimation, afin de garantir la cohérence + faciliter le test de diverses valeurs (e.g. bande optimale pour démoduler FM)

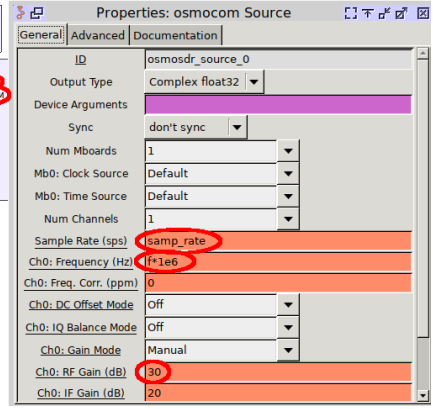
Interface matérielle : récepteur TNT RTL-SDR

- Détournement d'un récepteur de télévision numérique terrestre pour la réception de signaux radiofréquences
- R820T(2) (RF)+RTL2832U (ADC+USB) pour moins de 10 $/pièce
- fréquence d'échantillonnage 1 MHz $\leq f_s \leq$ 2,4 MHz
- porteuse $\in$ [50 : 1600] MHz
- gain variable de l'étage radiofréquence : [0 : 39] dB
- source GNU Radio : `gr-osmosdr` ou `soapysdr-module-osmosdr`









Applications

Reconfiguration du logiciel : une même plateforme matérielle permet d'appréhender une multitude de modes de communication

Mode	modulation	porteuse (MHz)	bande passante (kHz)
WBFM	FM analogique	88–108	120
DAB	OFDM (FM num)	175–250	1537
POCSAG	FSK	466	9
ADS-B	ASK	1090	50
XE1203	FSK	434, 868	dépend du débit
ACARS	AM	131,725	5
Iridium	PSK	1620	31,5
GPS	BPSK	1575,42	1000
NOOA (APT)	analog	137	34
Meteor M2 (LRPT)	QPSK	137,9	120
RADAR passif	bruit/xcorr		max ($\Delta R = c_0/(2B)$)

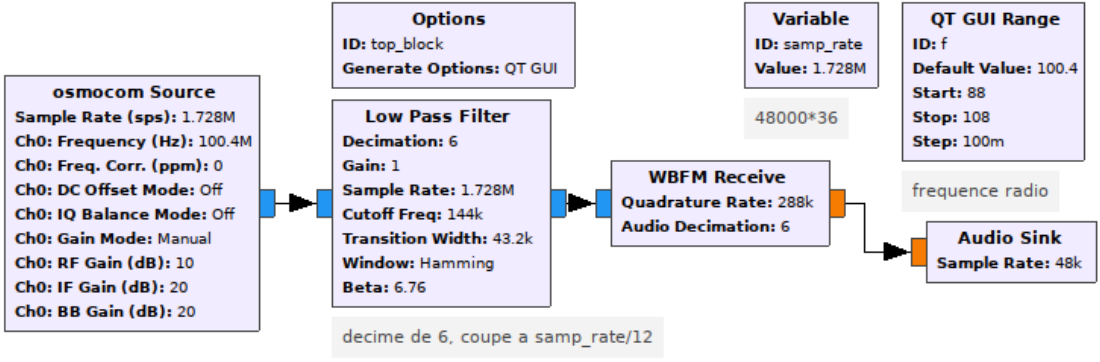
Tous[4] ces modes sont accessibles avec un récepteur échantillonnant à $fs \leq 2,4$ MS/s ($f_s \geq 2BW$)

4. https://www.sigidwiki.com/wiki/Signal_Identification_Guide

Exemple de réception : bande FM commerciale

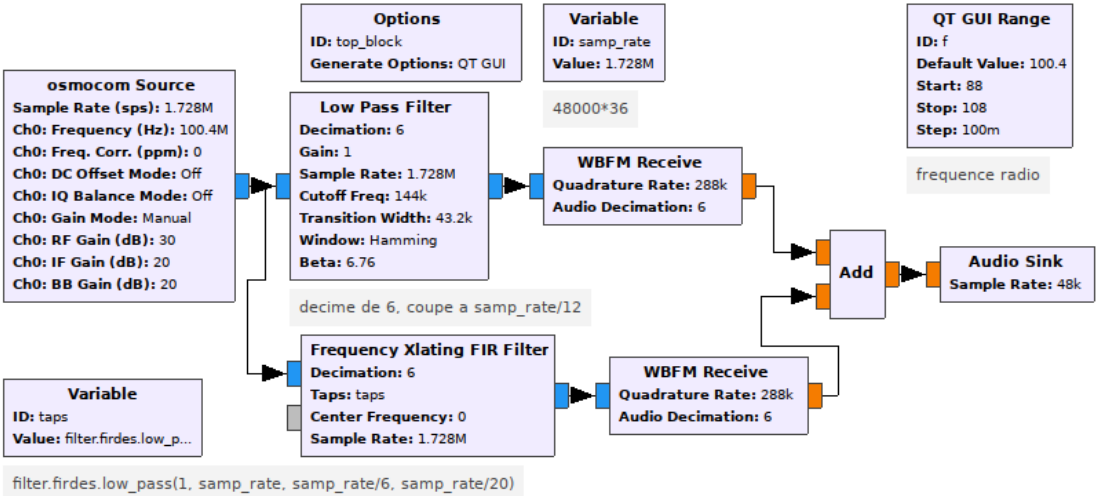
Recevoir une station FM

1. L'objectif est d'émettre du son sur une carte audio, avec $f_{audio} \in [22,05\ ;44,1\ ;48\ ;96]$ kHz
2. décimations par facteurs entiers : choisir un échantillonnage multiple de f_{audio}
3. sélection d'une station radio par filtrage-décimation
4. démodulation FM large bande pour fournir audio : filtrage-décimation



Exemple de réception : bande FM commerciale

Étendre les fonctionnalités : recevoir deux stations FM simultanément

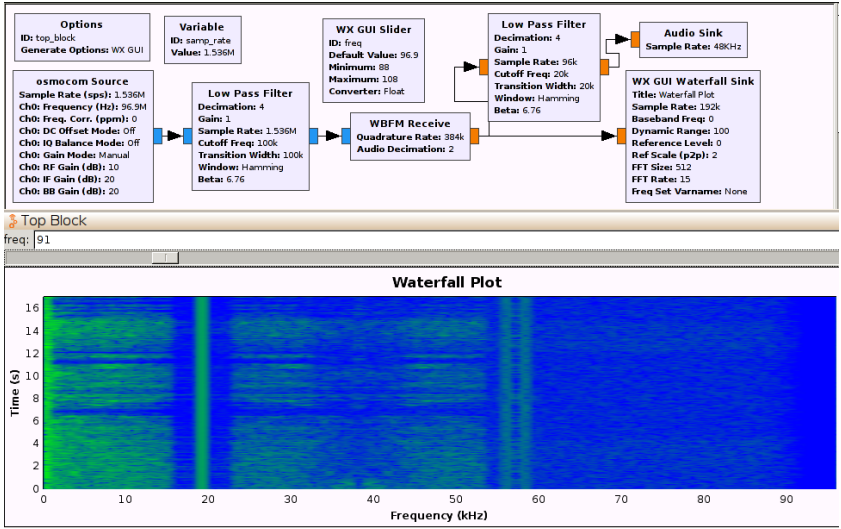


SDR : ajouter une fonction se limite à du logiciel, tant que la bande d'acquisition est suffisante.

Xlating FIR Filter : transposition de fréquence et filtrage

Exemple de réception : bande FM commerciale

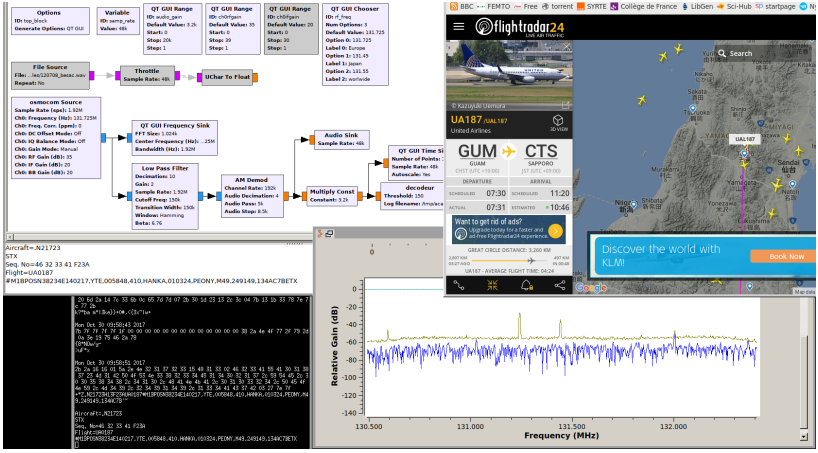
Affichage *waterfall* : diverses informations transmises par une station FM



L+R, pilote 19 kHz (stéréo), L-R, RDS (centré sur 57 kHz)

Exemples de signaux reçus

Aéronautique : ACARS[5][6] (131,725 MHz), ADS-B (1090 MHz)



Les liaisons aéronautiques civiles (108–138 MHz) sont modulées en AM, incluant contrôle aérien

5. `https://sourceforge.net/projects/gr-acars`

6. J.-M Friedt, G. Goavec-Merou, *La réception radiofréqence définie par logiciel (Software Defined Radio – SDR)*, GNU/Linux Magazine France 153 (Octobre 2012), pp.4-33 à `jmfriedt.free.fr/lm_sdr.pdf`

Démodulation FM

Problème : que se passe-t-il si on décalle la porteuse ?

- AFSK : Audio Frequency Shift Keying – la modulation FM porte une modulation audio
- la modulation audio peut porter une information numérique (ACARS)
- la modulation audio peut porter une information analogique (NOAA POES : 137,1 MHz, 137,62 MHz, 137,9125 MHz)

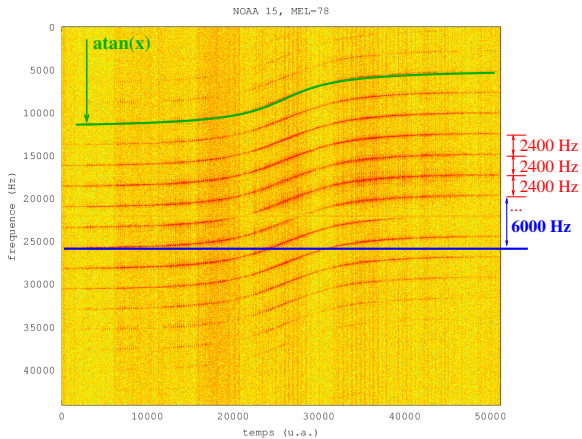
La porteuse sert de référence par rapport à laquelle les raies latérales portent l'information

film NOAA : `~/sdr/noaa/150430_08h49_spits.ogv`

Démodulation FM

- AFSK : Audio Frequency Shift Keying – la modulation FM porte une modulation audio
- la modulation audio peut porter une information numérique (ACARS)
- la modulation audio peut porter une information analogique (NOAA POES : 137,1 MHz, 137,62 MHz, 137,9125 MHz)

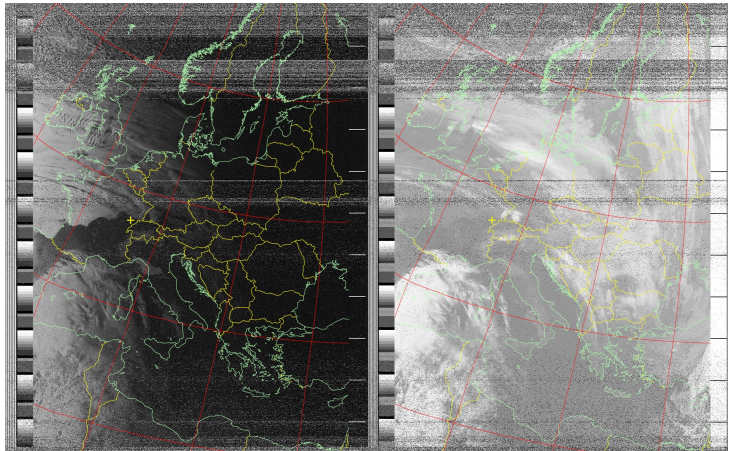
La porteuse sert de référence par rapport à laquelle les raies latérales portent l'information



Démodulation FM

- AFSK : Audio Frequency Shift Keying – la modulation FM porte une modulation audio
- la modulation audio peut porter une information numérique (ACARS)
- la modulation audio peut porter une information analogique (NOAA POES : 137,1 MHz, 137,62 MHz, 137,9125 MHz)

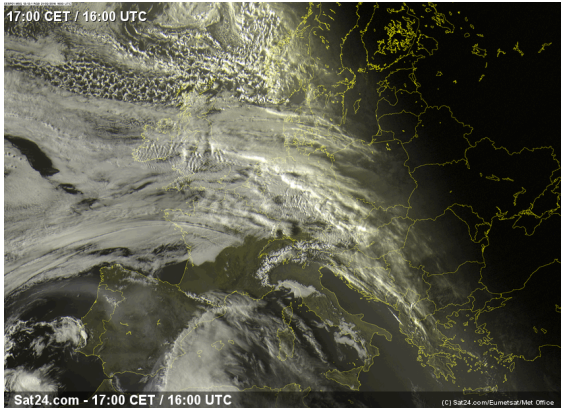
La porteuse sert de référence par rapport à laquelle les raies latérales portent l'information



Démodulation FM

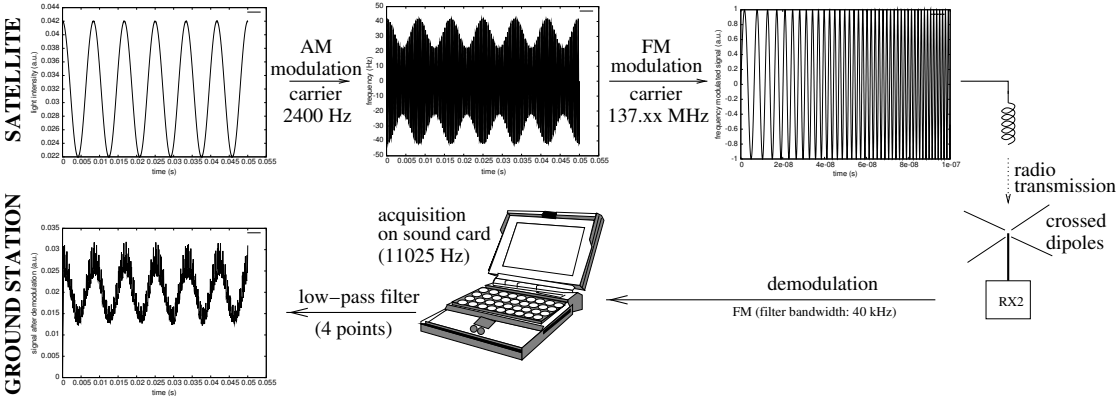
- AFSK : Audio Frequency Shift Keying – la modulation FM porte une modulation audio
- la modulation audio peut porter une information numérique (ACARS)
- la modulation audio peut porter une information analogique (NOAA POES : 137,1 MHz, 137,62 MHz, 137,9125 MHz)

La porteuse sert de référence par rapport à laquelle les raies latérales portent l'information



Modulation des signaux POES [8]

L'intensité lumineuse des pixels module en amplitude une porteuse audiofréquence (2400 Hz) qui module une porteuse radiofréquence ($\simeq$ 137 MHz) en fréquence [7] (APT).



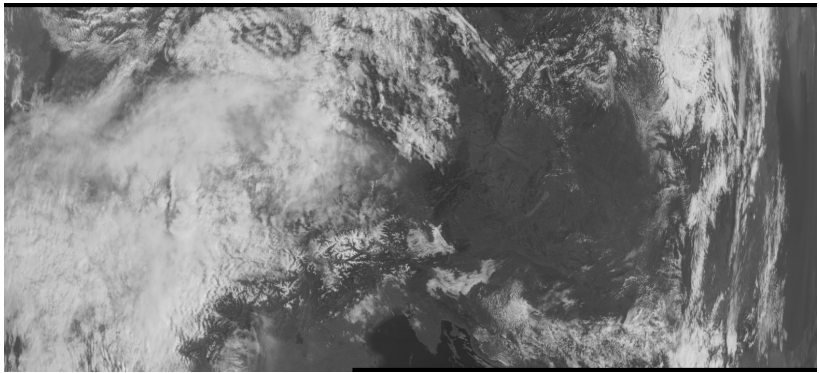
Doppler : $v = \frac{40000 \text{ km}}{1.5 \text{ h}} = 3400\text{m/s}$ et $\Delta f = f \cdot \frac{v}{c} \cdot \frac{R}{R+r}$

7. J.-M Friedt, *Satellite image eavesdropping : a multidisciplinary science education project*, European Journal of Physics **26** (August 2005) pp.969-984

8. Polar Orbiting Environmental Satellites – 800 km

Exemples de signaux reçus

Analogique→ numérique : METEOR M2 (137,9 MHz [9])



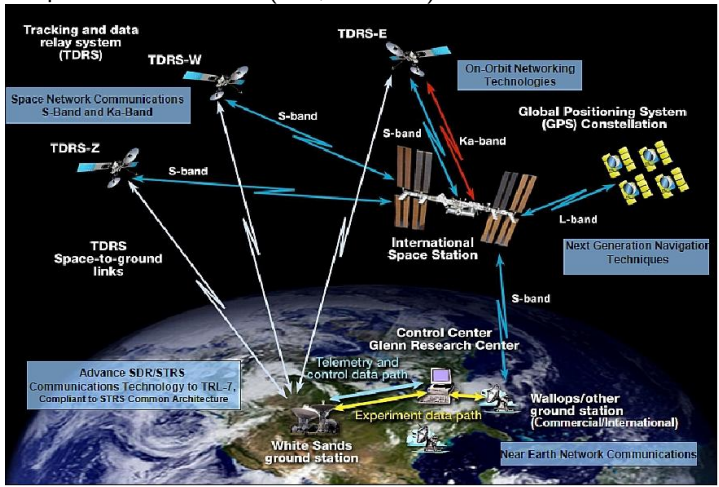
Opportunité de comprendre les couches OSI

- couche physique (modulation QPSK)
- code convolutif pour entrelacer les bits & algorithme de Viterbi
- éventuellement code correcteur d'erreur Reed-Solomon
- images encodées en JPEG (Huffman, DCT)

9. http://phqfh.co.uk/status.htm

Exemples de signaux reçus

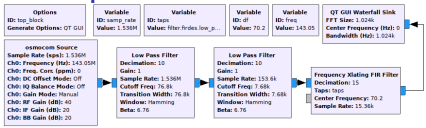
Analogique→ numérique : METEOR M2 (137,9 MHz [9])

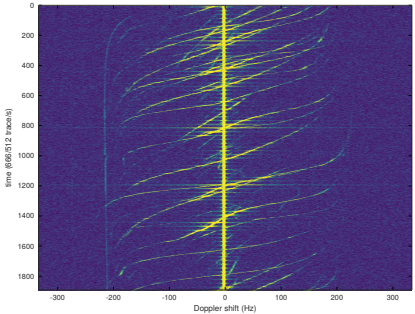


Opportunité de comprendre les couches OSI

- couche physique (modulation QPSK)
- code convolutif pour entrelacer les bits & algorithme de Viterbi

Exemple : RADAR



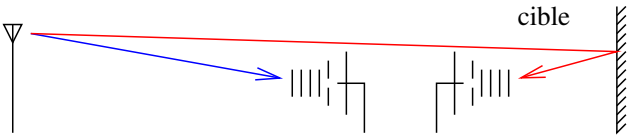


- GRAVES (Grand Réseau Adapté à la VEille Spatiale) : 143,05 MHz, 30 km de Besançon, $\geq$400 kW
- RADAR à onde continue : pas de notion de distance
- Surveillance de l'espace : les lois de Kepler relient la vitesse à l'altitude, inutile de connaître la distance
- décimations successives pour ne par nécessiter un nombre prohibitif de coefficients de filtrage
- RADAR : loi en $1/d^4$ (cible ponctuelle)

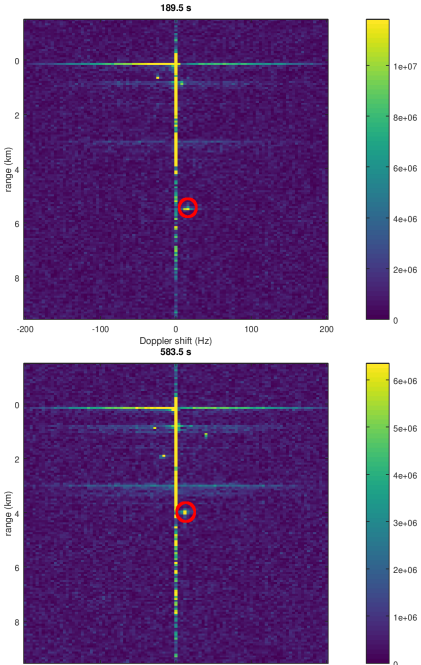
W. Feng, J.-M Friedt, G. Cherniak, M. Sato, *(Yet another) passive RADAR using DVB-T receiver and SDR*, FOSDEM 2018

Exemples de signaux reçus

- RADAR passif : exploiter les source radiofréquences existantes pour détecter des cibles
- Exemple des émetteurs de télévision numérique terrestre (3 kW)
- signal émis inconnu ⇒ enregistrer référence et observation
- intercorrélation est filtre optimal pour trouver référence dans observation (après compensation décallage Doppler)



W. Feng, J.-M Friedt, G. Cherniak, M. Sato, *Passive bistatic radar using digital video broadcasting–terrestrial receivers as general-purpose software-defined radio receivers*, Rev. Sci. Instrum. **89**, 104701 (2018)



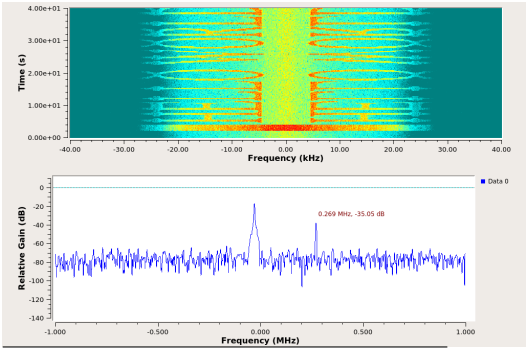
AM v.s FM

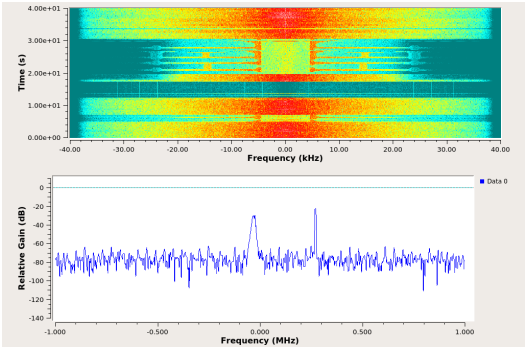
Pourquoi les émissions aéronautiques sont-elles modulées en AM ?

- occupation spectrale plus importante en FM qu'en AM
- inefficacité de la transmission AM qui n'utilise pas la pleine puissance de l'amplificateur

AM v.s FM

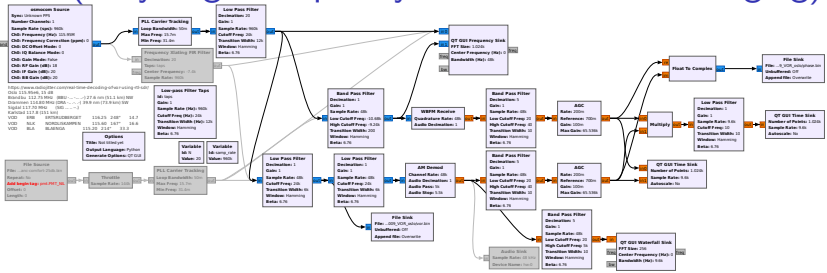
- effet “capture” de la FM : la station la plus puissante est la seule démodulée par la PLL, au détriment de la plus faible
- en AM, une interférence est détectable car les deux stations sont audibles [10]
- en FM, seule la station la plus puissante est entendue, la plus faible est “supprimée” ...
- qui est en faveur [11] de la FM pour les émissions commerciales. [12]



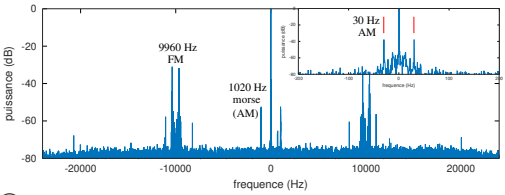


10. `http://www.airliners.net/forum/viewtopic.php?t=1357995`
11. G.J. King, *FM Radio Servicing Handbook, 2nd Ed.*, Newnes-Butterworths (1970)
12. J.- M Friedt,*La peinture sur spectre radiofréquence, et l'effet capture de la modulation en fréquence – ou pourquoi les avions communiquent encore en AM*, GNU/Linux Magazine France 216 (Juin 2018)

VOR (Very High Frequency Omnidirectional Ranging) [13]



Deux signaux portés par une porteuse radiofréquence modulée en amplitude : modulation en amplitude et en phase de deux sinusoïdes dont la phase est l'azimuth



13. J.-M Friedt, *Décodage par radio logicielle du VOR pour le positionnement sans GPS*, Hackable **36** (2021)

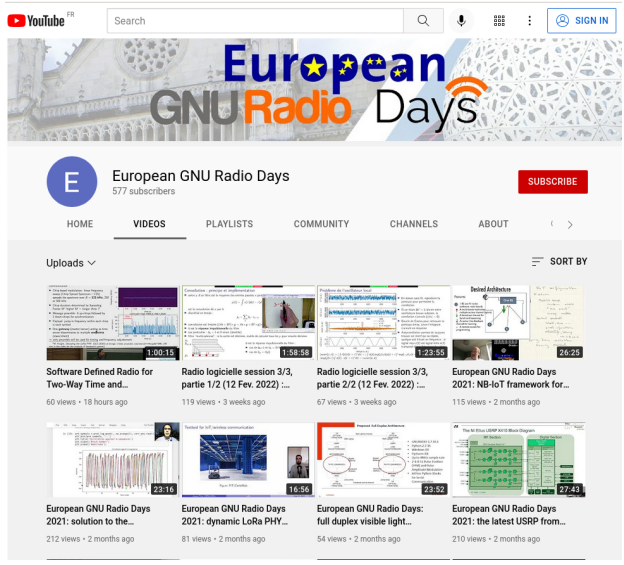
Quelques exemples de récepteurs SDR

Nom	Porteuse [14]	BW [15] (MHz)	Rés. [16]	Émetteur [17]	Coût
RTL-SDR	50-1600	2,4	8 bits	N	9 euros
Pluto (AD)	70-6000 (325–3800)	20 MHz	12 bits	O	89→150 euros
HackRF One	1-6000	20 MHz	8 bits	O	262 euros
LimeSDR	0,1-3800	61,4 MHz	12 bits	O MIMO	299$
BladeRF	300-3800	40 MHz	12 bits	O	420$
Ettus B210	70-6000	61,4 MHz	12 bits	O MIMO	1220 euros
Ettus X310	70-6000	160 MHz	12 bits	O MIMO	5270 euros+frontend

14. Définit la bande de fréquence accessible, e.g. 88–108 MHz FM commerciale, 137 MHz sat. POES, 460 MHz POCSAG, 1574,2 MHz GPS, 1620 MHz Iridium ...
15. Détermine la nature des protocoles décodables
16. Dynamique $20\log_{10}(256) = 48$ dB ; $20\log_{10}(4096) = 72$ dB
17. L'émission est règlementée sur les bandes de fréquence, puissance et rapport cyclique

Conclusion

- La radio logicielle comme une approche **pédagogique** à l'enseignement du **traitement du signal numérique** échantillonné en temps discret
- La radio logicielle comme une approche **flexible** au traitement de signaux radiofréquences
- Disponibilité d'un environnement **libre** de prototypage pour se familiariser avec les concepts
- Domaine abordable pour un coût démarrant à moins de 10 euros
- Perspectives **multiples de développements** (communications numériques ou analogiques, sécurité, caractérisation de canaux de propagation, liaisons spatiales, RADAR ...) – voir `websdr`



Bibliographie 1/2

1. T. Collins *& al.*, *Software-Defined Radio for Engineers*, (2018) at `https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html`
2. S.W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing, 2nd Ed* (1999) at `https://www.dspguide.com/pdfbook.htm`
3. T. McDermott, *Wireless Digital Communications : Design and Theory*, Tucson Amateur Packet Radio Corporation – TAPR (1997)
4. J.G. Proakis, D.K. Manolakis, *Digital Signal Processing*, Prentice Hall (2006)
5. R.G. Lyons, *Understanding Digital Signal Processing*, Prentice Hall (2004)
6. A.V. Oppenheim, R.W. Schafer, *Discrete-Time Signal Processing (3rd Edition)*, Prentice-Hall Signal Processing Series (2009), and videos of his lectures at `ocw.mit.edu/resources/res-6-007-signals-and-systems-spring-2011/video-lectures/lecture-1-introduction/`
7. K. Borre, D.M. Akos, N. Bertelsen, *A Software-Defined GPS and Galileo Receiver : A Single-Frequency Approach*, Birkhäuser (2007)
8. E.D. Kaplan, C. Hegarty, *Understanding GPS : Principles and Applications, 2nd Ed.*, Artech House (2005)
9. C.A. Balanis, *Antenna Theory, Analysis and Design*, Wiley Interscience (2005)

Bibliographie 2/2

Sur le Web :

1. Principles of Digital Communications course at `ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-450-principles-of-digital-communications-i-fall-2006/video-lectures/`
2. Balint videos at `www.youtube.com/playlist?list=PL618122BD66C8B3C4` and `www.youtube.com/watch?v=1bgC3AjCnA4`
3. Tom Rondeau's presentations, for example `gnuradio.org/redmine/projects/gnuradio/wiki/Guided_Tutorial_PSK_Demodulation` and `www.youtube.com/watch?v=_hGNT1w-jig`
4. M. Lichtman, *PySDR : A Guide to SDR and DSP using Python* at `https://pysdr.org` (accessed 02/2021[18])
5. *Learning DSP Illustrated* at `https://dspillustrations.com/pages/index.html` (accessed 02/2021)

18. presented at FOSDEM2021 at `https://fosdem.org/2021/schedule/event/fsr_pysdr_guide_to_sdr_and_dsp_using_python/`