

Digital communication

J.-M Friedt

FEMTO-ST/time & frequency department

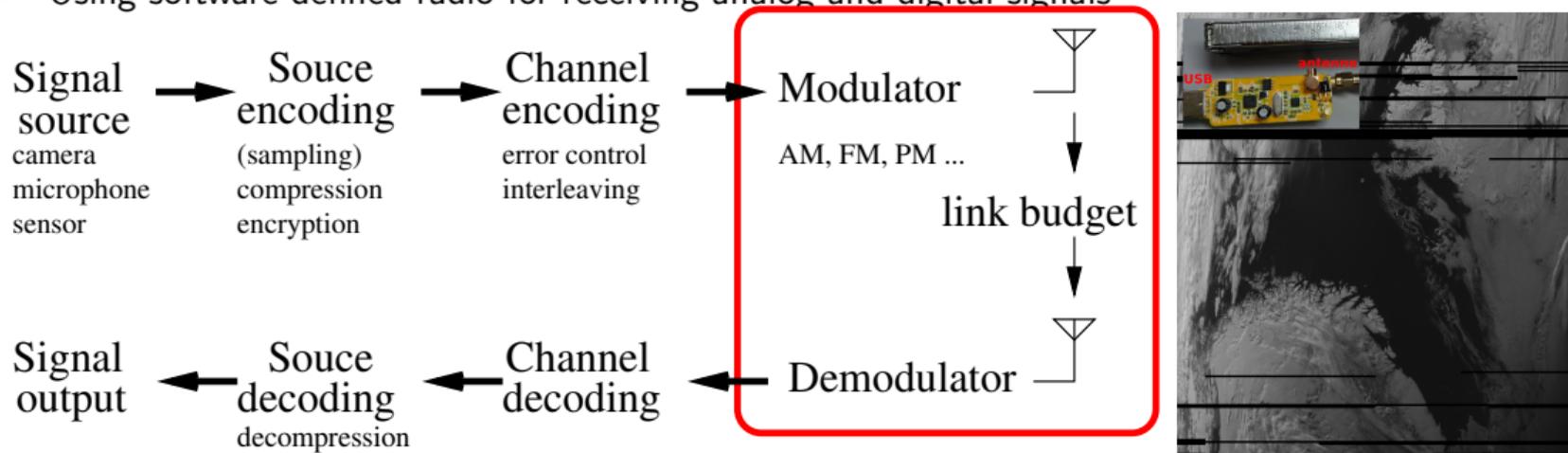
`jmfriedt@femto-st.fr`

slides and references at `jmfriedt.free.fr`

February 14, 2024

Objective of this presentation

- ▶ Receiving (digital) signals transmitted on a radiofrequency carrier ^{1 2}
- ▶ Characterizing complex signal characteristics defining their modulation scheme
- ▶ Using software defined radio for receiving analog and digital signals



¹J.-M Friedt, *Decoding Meteor M2: QPSK, Viterbi, Reed Solomon and JPEG*, FOSDEM 2019 (Free Software Radio devroom)

²J.-M Friedt, *Décodage d'images numériques issues de satellites météorologiques en orbite basse : le protocole LRPT de Meteor-M2* (3 parties), GNU/Linux Magazine France 226 (Mai 2019), 227 (Juin 2019), 228 (Juillet-Aout 2019)

Objective of this presentation

- ▶ Receiving (digital) signals transmitted on a radiofrequency carrier ^{1 2}
- ▶ Characterizing complex signal characteristics defining their modulation scheme
- ▶ Using software defined radio for receiving analog and digital signals

$$s(t) = A(t) \cos(\omega(t) \cdot t + \varphi(t))$$

⇒ AM, FM, PM

Tools:

- ▶ terrestrial digital video broadcast (DVB-T) television receiver RTL2832U + R820T(2)
- ▶ 25-1750 MHz, <10 \$
- ▶ GNU Radio

Try by yourself



(jmfriedt.free.fr/acars_audio_with_filter.mp4)

Bibliography: R.G. Lyons, *Understanding Digital Signal Processing* (Prentice Hall, 1997)

S.W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing, 2nd Ed* (1999) at www.dspguide.com/pdfbook.htm

¹J.-M Friedt, *Decoding Meteor M2: QPSK, Viterbi, Reed Solomon and JPEG*, FOSDEM 2019 (Free Software Radio devroom)

²J.-M Friedt, *Décodage d'images numériques issues de satellites météorologiques en orbite basse : le protocole LRPT de Meteor-M2* (3 parties), GNU/Linux Magazine France 226 (Mai 2019), 227 (Juin 2019), 228 (Juillet-Aout 2019)

Objective of this presentation

- ▶ Receiving (digital) signals transmitted on a radiofrequency carrier ^{1 2}
- ▶ Characterizing complex signal characteristics defining their modulation scheme
- ▶ Using software defined radio for receiving analog and digital signals



131.725 MHz, AM (airband – ACARS and Reims ATC) @
http://jmfriedt.free.fr/acars_audio_with_filter.mp4

Outline of the presentation

10 lessons of 1.5 h each

1. Basics of radiofrequency signal processing
2. Modulations: why and how
3. Sharing the radiofrequency spectrum and multichannel analysis (FDMA, TDMA, SDMA)
4. GPS decoding – sharing resources by coding (CDMA)
5. Radiofrequency link budget
6. Symbol synchronization, channel capacity (Shannon)
7. Digital radiofrequency signal software processing specificities.
8. Antenna: characteristics and design

5 laboratory sessions of 3 h each, practicing with GNU Radio and `gnuradio-companion`:

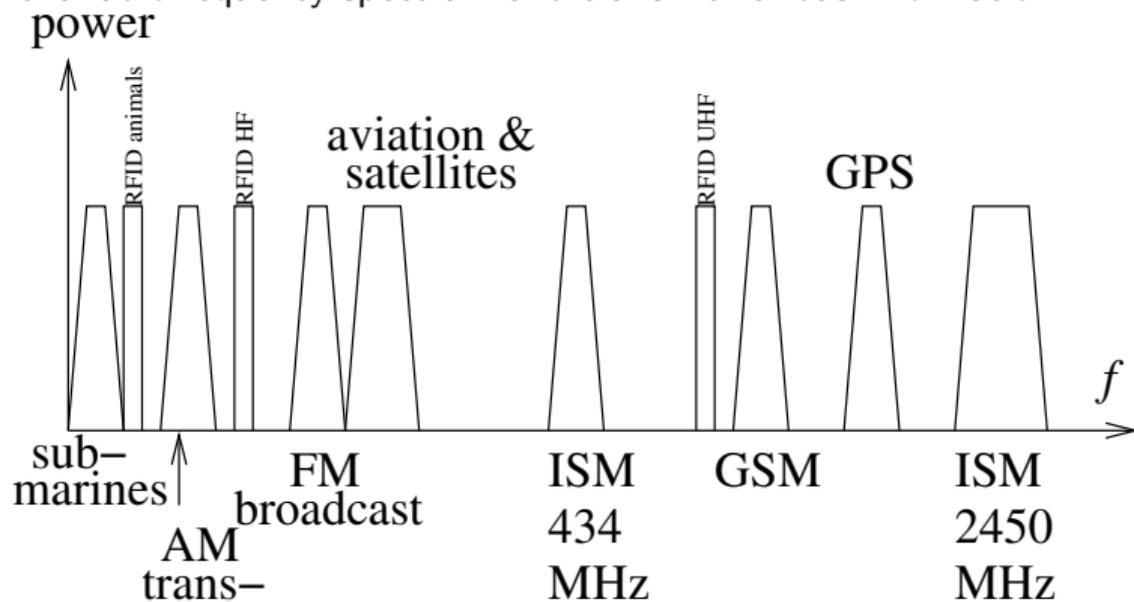
1. GNU Radio as a general purpose signal processing prototyping tool: application to AM and FM
2. PM demodulation & decoding multiple bands (analog FM and/or digital POCSAG)
3. CDMA/GPS signal characterization, correlations (GNU Radio & Octave)
4. antenna design using NEC2 and radiation pattern display
5. clock synchronization: RDS (digital information in FM broadcast)

Assumption: handling complex numbers with Matlab-GNU/Octave and/or Python/numpy

Radiofrequency link

- ▶ replace a wire with an electromagnetic wave
- ▶ $\lambda = c/f \Rightarrow \lambda[\text{m}] = 300/f[\text{MHz}]$
- ▶ VLF (Very Low Frequency) to penetrate conducting media/long communication range (over the horizon)
- ▶ VHF (Very High Frequency)-UHF (Ultra ...): 30-1000 MHz
- ▶ SHF (Super ... 2400 MHz) $\Rightarrow \lambda = 12.5 \text{ cm}$
- ▶ various carriers to multiplex the radiofrequency spectrum and transmit various information in different spectral bands

The carrier frequency is of little interest, **only the bandwidth** defines the amount of information that can be transmitted after transposition to baseband by the receiver.

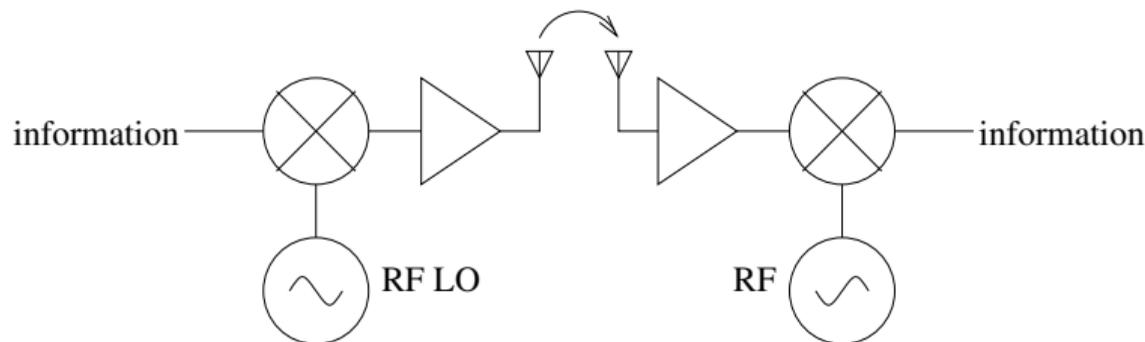


In the beginning ...

- ▶ Morse code: presence (1) or absence (0) of a carrier
- ▶ used in telegraphic communication, transposed to radiofrequency carrier
- ▶ poor rejection of noise
- ▶ named OOK (On-Off Keying) in the digital communication literature
- ▶ evolution to ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying), PSK (Phase Shift Keying)... for digital communication

Question: how to transpose a signal to a frequency band appropriate for the targeted application ?

3



Be it known that I, ALEXANDER GRAHAM BELL, of Salem, Massachusetts, have invented certain new and useful Improvements in Apparatus for Transmitting and Receiving Telegraphic Signals or Messages, of which the following is a specification:

In another application for Letters Patent I have described a method of and apparatus for transmitting two or more telegraphic messages or signals simultaneously along a single wire by the employment of transmitting-instruments, each of which transmits, persecond, impulses differing in number from the others, and receiving-instruments, each tuned to a pitch at which it will be put in vibration to produce its fundamental tone by one only of the transmitting-instruments.

³A.G. Bell, *Improvement in Transmitters and Receivers for Electric Telegraphs*, U.S. Patent 0,161,739 (April 6, 1875)

R820T2 internals ⁴: a radiofrequency receiver needs ...

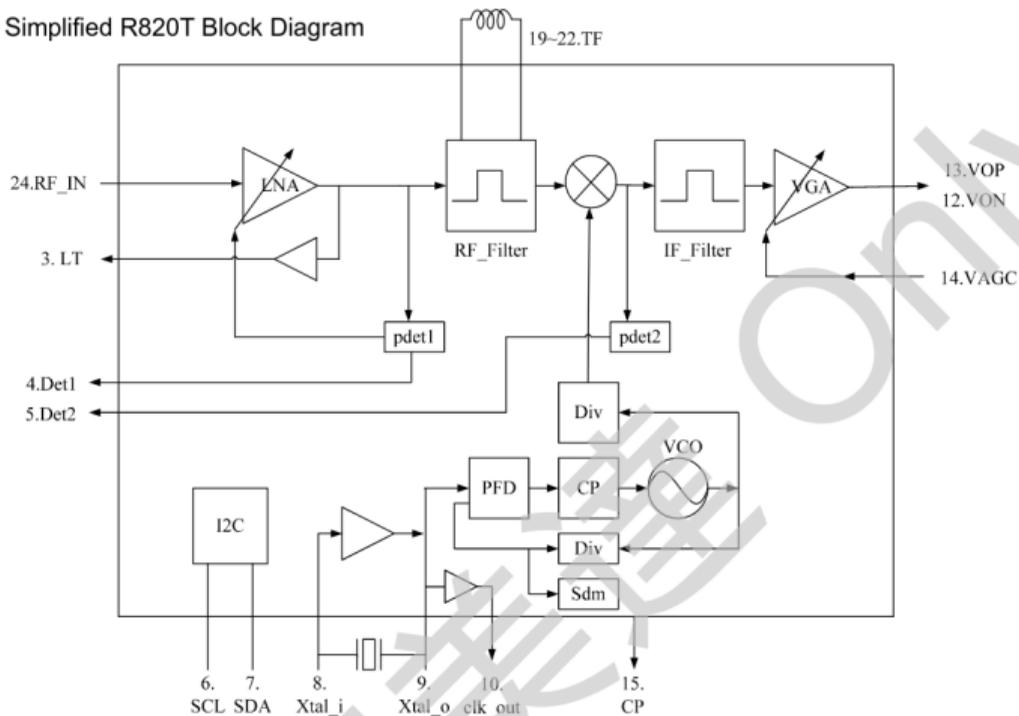
- ▶ a Low Noise Amplifier (LNA) to amplify the signal received from the antenna
- ▶ a local radiofrequency oscillator defining the center-frequency being monitored
- ▶ a mixer for frequency transposition
- ▶ possibly a Variable Gain Amplifier (VGA) to optimize the full-scale range of the analog to digital converter (ADC)



⁴[https:](https://www.rtl-sdr.com/wp-content/uploads/2013/04/R820T_datasheet-Non_R-20111130_unlocked1.pdf)

[//www.rtl-sdr.com/wp-content/uploads/2013/04/R820T_datasheet-Non_R-20111130_unlocked1.pdf](https://www.rtl-sdr.com/wp-content/uploads/2013/04/R820T_datasheet-Non_R-20111130_unlocked1.pdf)

Simplified R820T Block Diagram



Mixer for frequency transposition

- ▶ Linear Time Invariant (LTI) system:

$$f(A \exp(j\omega_1 t) + B \exp(j\omega_2 t)) = Af(\exp(j\omega_1 t)) + Bf(\exp(j\omega_2 t))$$

- ▶ ... allows for Fourier decomposition and solving differential equations in the stationary regime since

$$\frac{dA \exp(j\omega t)}{dt} = jA\omega \exp(j\omega t)$$

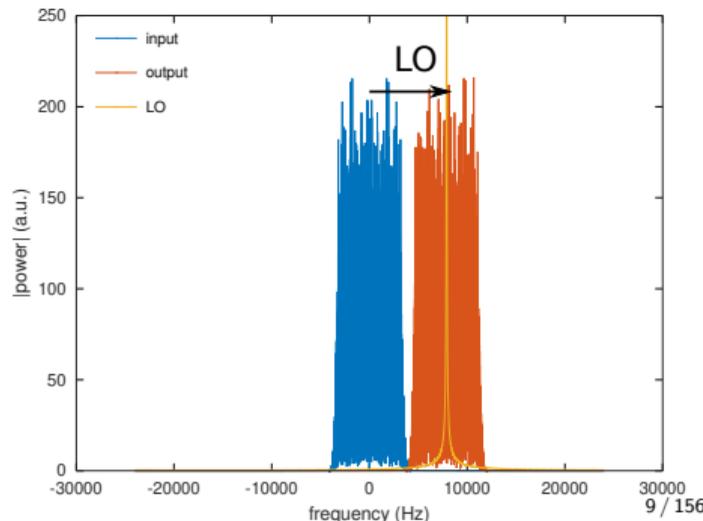
or analyzing f each frequency at a time (Fourier analysis).

- ▶ LTI: no frequency mixing, combining ω_1 and ω_2 can only create a linear combination of ω_1 and ω_2 , **never** $\omega_1 \pm \omega_2$
- ▶ **Mixer** (multiplication in the digital domain):

$$A \exp(j\omega_1 t) \times B \exp(j\omega_2 t) = A \cdot B \exp(j(\omega_1 + \omega_2)t)$$

Matlab-like syntax (GNU Octave):

```
N=65536;           % number of samples
fs=48000;         % sampling frequency
f=7900;           % transposition frequency
in=rand(N,1); in=in-mean(in);
b=firls(256,[0 2200 4400 fs/2]*2/fs,[1 1 0 0]);
in=filter(b,1,in); % ~~~~~ Nyquist freq,
t=[0:N-1]'/fs; % discrete time
lo=exp(j*2*pi*f*t); % LO
out=in.*lo;      % transposition
freq=linspace(-fs/2,fs/2-fs/N,N);
plot(freq,abs(fftshift(fft(in))));hold on
plot(freq,abs(fftshift(fft(out))));
```



Mixer for frequency transposition

- ▶ Linear Time Invariant (LTI) system:

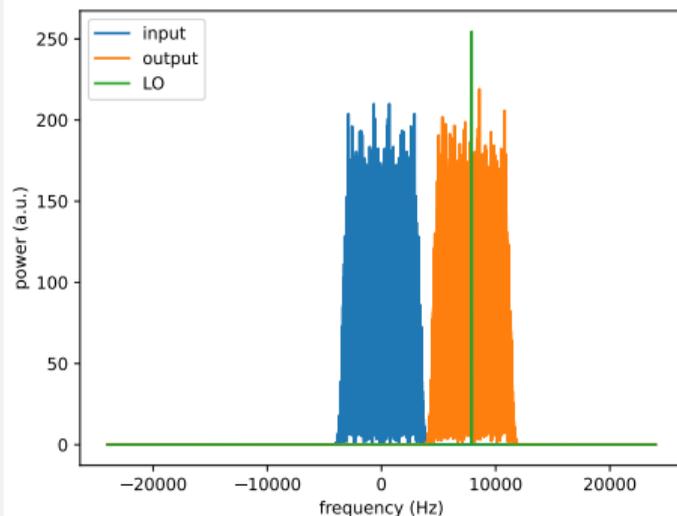
$$f(A \exp(j\omega_1 t) + B \exp(j\omega_2 t)) = Af(\exp(j\omega_1 t)) + Bf(\exp(j\omega_2 t))$$

- ▶ Mixer (multiplication in the digital domain):

$$A \exp(j\omega_1 t) \times B \exp(j\omega_2 t) = A \cdot B \exp(j(\omega_1 + \omega_2)t)$$

Python syntax:

```
import numpy as np
import scipy as sp
from matplotlib import pyplot as plt
N=65536;           # number of samples
fs=48000;         # sampling frequency
f=7900;           # transposition frequency
inv=np.random.rand(N); inv=inv-np.mean(inv);
b=sp.signal.firls(255,np.array([0, 2200, 4400, fs→
    ↪/2]),np.array([1, 1, 0, 0]),fs=fs);
inv=sp.signal.lfilter(b,1,inv);
t=np.arange(0,N)/fs; # discrete time
lo=np.exp(1j*2*np.pi*f*t); # LO
out=inv*lo;         # transposition
freq=np.linspace(-fs/2,fs/2,N);
plt.plot(freq,abs(np.fft.fftshift(np.fft.fft(inv)))→
    ↪,label='input');
plt.plot(freq,abs(np.fft.fftshift(np.fft.fft(out)))→
    ↪,label='output');
plt.xlabel('frequency (Hz)')
```



Discrete time signal processing

Discrete levels \Rightarrow limited dynamic range

Discrete time \Rightarrow finite bandwidth, baseband, aliasing

- ▶ sampling on **discrete** levels (quantization steps)

- ▶ $value = \frac{V}{V_{ref}} \times (2^N - 1)$

- ▶ dynamic range limited by the smallest detectable voltage variation $\Delta V = V_{ref} / (2^N - 1)$

- ▶ radiofrequency: conversion to logarithmic power $20 \log_{10}(\Delta V)$ is 48 dB on 8 bits, 60 dB on 10 bits, 72 dB on 12 bits and 96 dB on 16 bits

- ▶ sampling at **discrete time** steps

- ▶ sampling frequency f_s (sampling period $1/f_s$)

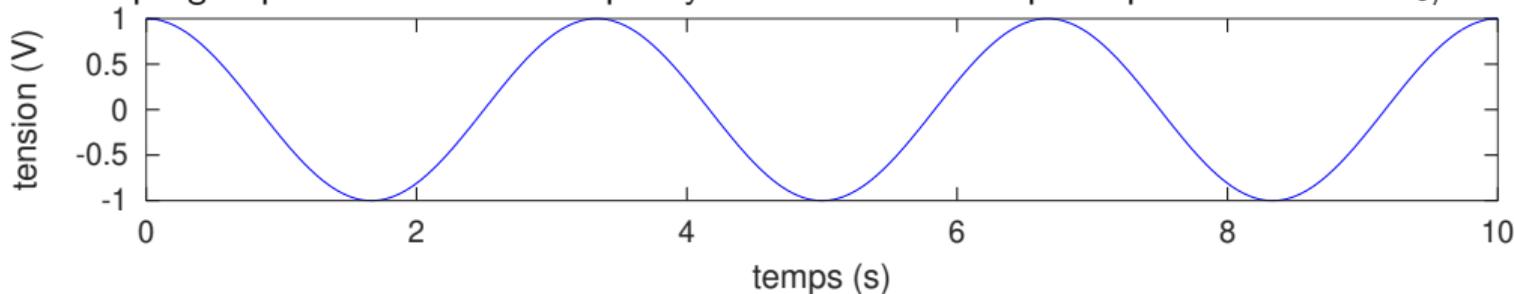
- ▶ spectrum periodicity hypothesis (cf aliasing)

- ▶ spectrum spanning from $-f_s/2$ to $+f_s/2$

- ▶ f_s known $\Rightarrow f_s/2$ is defined (Nyquist frequency) as 1 and express all frequencies f as normalized frequency

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

- ▶ sampling N points \Rightarrow discrete Frequency transform on N samples: spectral resolution f_s/N



Discrete time signal processing

Discrete levels \Rightarrow limited dynamic range

Discrete time \Rightarrow finite bandwidth, baseband, aliasing

▶ sampling on **discrete** levels (quantization steps)

▶ $value = \frac{V}{V_{ref}} \times (2^N - 1)$

▶ dynamic range limited by the smallest detectable voltage variation $\Delta V = V_{ref} / (2^N - 1)$

▶ radiofrequency: conversion to logarithmic power $20 \log_{10}(\Delta V)$ is 48 dB on 8 bits, 60 dB on 10 bits, 72 dB on 12 bits and 96 dB on 16 bits

▶ sampling at **discrete time** steps

▶ sampling frequency f_s (sampling period $1/f_s$)

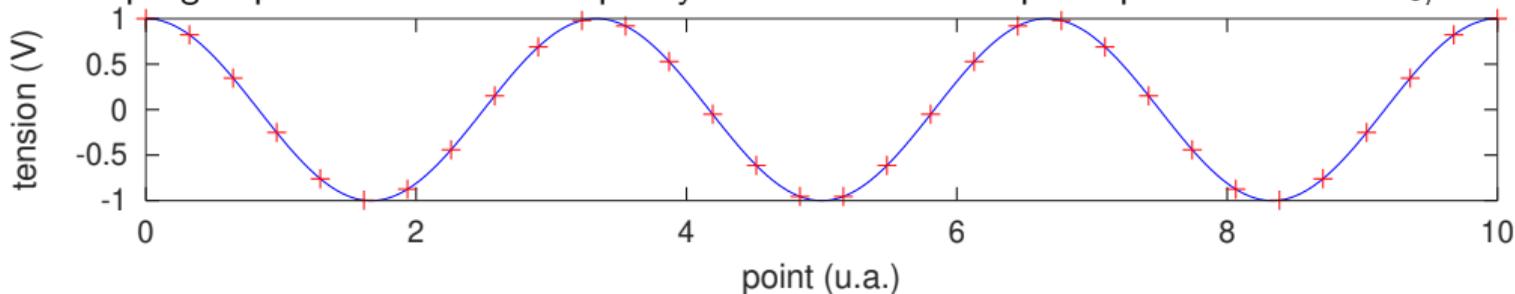
▶ spectrum periodicity hypothesis (cf aliasing)

▶ spectrum spanning from $-f_s/2$ to $+f_s/2$

▶ f_s known $\Rightarrow f_s/2$ is defined (Nyquist frequency) as 1 and express all frequencies f as normalized frequency

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

▶ sampling N points \Rightarrow discrete Frequency transform on N samples: spectral resolution f_s/N



Discrete time signal processing

Discrete levels \Rightarrow limited dynamic range

Discrete time \Rightarrow finite bandwidth, baseband, aliasing

- ▶ sampling on **discrete** levels (quantization steps)

- ▶ $value = \frac{V}{V_{ref}} \times (2^N - 1)$

- ▶ dynamic range limited by the smallest detectable voltage variation $\Delta V = V_{ref} / (2^N - 1)$

- ▶ radiofrequency: conversion to logarithmic power $20 \log_{10}(\Delta V)$ is 48 dB on 8 bits, 60 dB on 10 bits, 72 dB on 12 bits and 96 dB on 16 bits

- ▶ sampling at **discrete time** steps

- ▶ sampling frequency f_s (sampling period $1/f_s$)

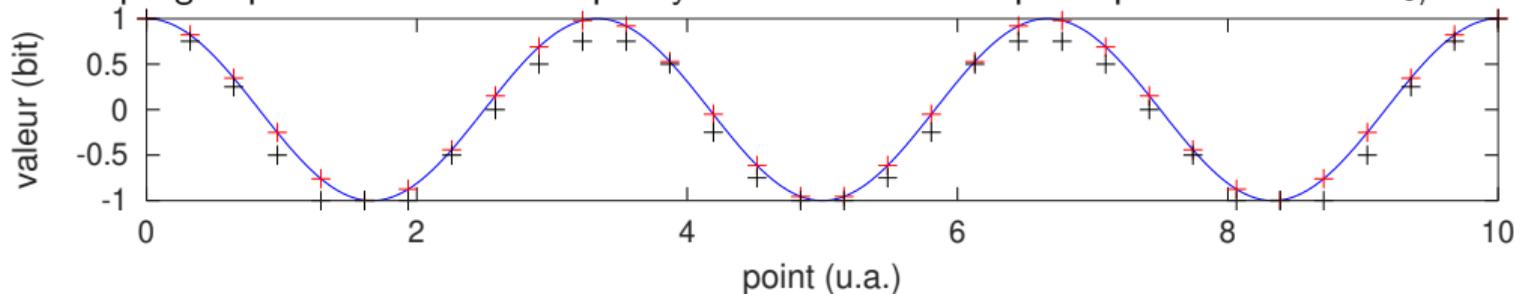
- ▶ spectrum periodicity hypothesis (cf aliasing)

- ▶ spectrum spanning from $-f_s/2$ to $+f_s/2$

- ▶ f_s known $\Rightarrow f_s/2$ is defined (Nyquist frequency) as 1 and express all frequencies f as normalized frequency

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

- ▶ sampling N points \Rightarrow discrete Frequency transform on N samples: spectral resolution f_s/N



Discrete time signal processing

Discrete levels \Rightarrow limited dynamic range

Discrete time \Rightarrow finite bandwidth, baseband, aliasing

- ▶ sampling on **discrete** levels (quantization steps)

- ▶ $value = \frac{V}{V_{ref}} \times (2^N - 1)$

- ▶ dynamic range limited by the smallest detectable voltage variation $\Delta V = V_{ref} / (2^N - 1)$

- ▶ radiofrequency: conversion to logarithmic power $20 \log_{10}(\Delta V)$ is 48 dB on 8 bits, 60 dB on 10 bits, 72 dB on 12 bits and 96 dB on 16 bits

- ▶ sampling at **discrete time** steps

- ▶ sampling frequency f_s (sampling period $1/f_s$)

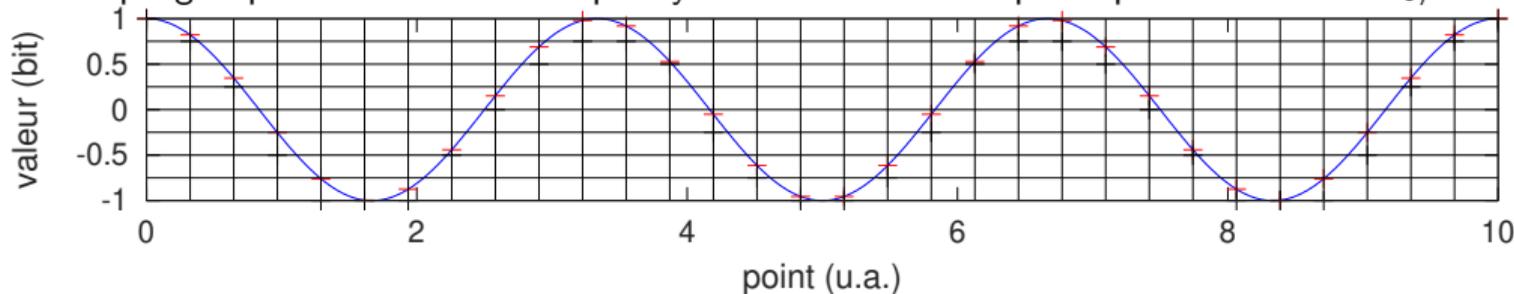
- ▶ spectrum periodicity hypothesis (cf aliasing)

- ▶ spectrum spanning from $-f_s/2$ to $+f_s/2$

- ▶ f_s known $\Rightarrow f_s/2$ is defined (Nyquist frequency) as 1 and express all frequencies f as normalized frequency

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

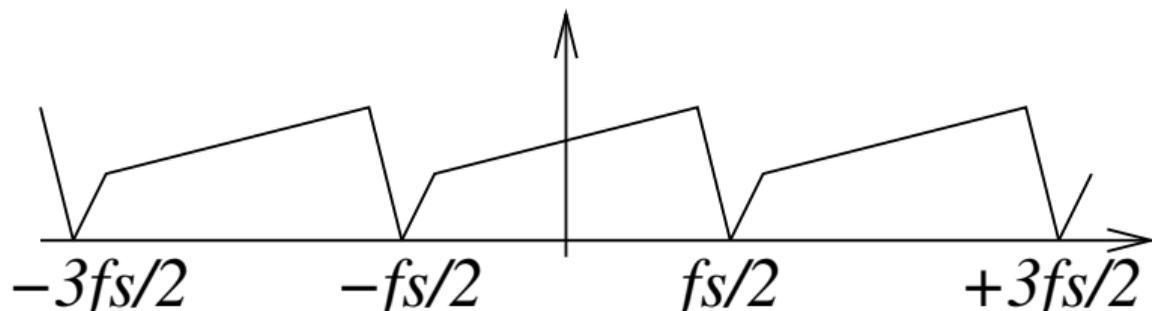
- ▶ sampling N points \Rightarrow discrete Frequency transform on N samples: spectral resolution f_s/N



Discrete time Fourier transform

Abscissa, aliasing and baseband

- ▶ continuous time domain signal is periodically sampled at a rate of f_s samples/s
- ▶ sampling theorem: the highest spectral component is $f_s/2$
- ▶ complex signal: positive and negative parts of the spectra are not related (real signal: $X(-f) = X^*(f) \Leftrightarrow |X(-f)| = |X(f)|$)
- ▶ Spectrum periodicity hypothesis
- ▶ $f_s/2$ is the **Nyquist**, defines the boundaries of **baseband** centered on 0 Hz
- ▶ Discrete frequency: the X-axis of the discrete Fourier transform ranges from $-f_s/2$ to $+f_s/2 - f_s/N$ and **includes 0 Hz the DC** component: `freq=linspace(-fs/2,fs/2-fs/N,N)`;

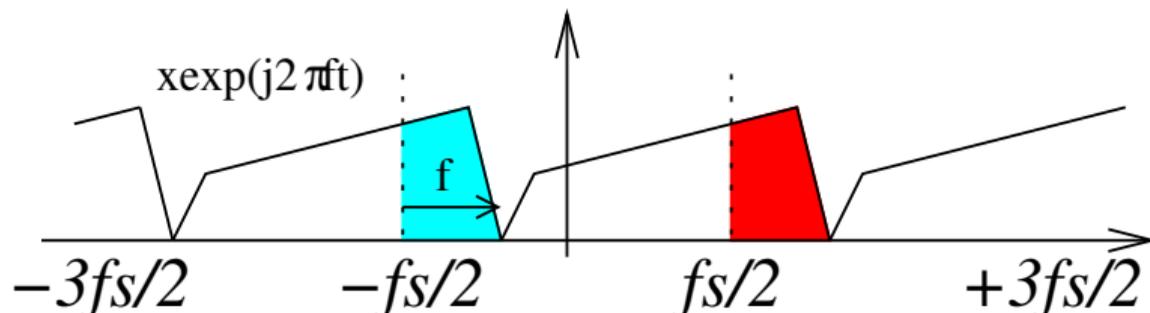


Remember: $\exp(j\omega_1 t) \cdot \exp(j\omega_2 t) = \exp(j(\omega_1 + \omega_2)t)$

Discrete time Fourier transform

Abscissa, aliasing and baseband

- ▶ continuous time domain signal is periodically sampled at a rate of f_s samples/s
- ▶ sampling theorem: the highest spectral component is $f_s/2$
- ▶ complex signal: positive and negative parts of the spectra are not related (real signal: $X(-f) = X^*(f) \Leftrightarrow |X(-f)| = |X(f)|$)
- ▶ Spectrum periodicity hypothesis
- ▶ $f_s/2$ is the **Nyquist**, defines the boundaries of **baseband** centered on 0 Hz
- ▶ Discrete frequency: the X-axis of the discrete Fourier transform ranges from $-f_s/2$ to $+f_s/2 - f_s/N$ and **includes 0 Hz the DC** component: `freq=linspace(-fs/2,fs/2-fs/N,N)`;



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

Mixer (I/Q)

- ▶ mixing for frequency transposition:

$$\underbrace{A_1 \sin(\omega_1 t + \varphi)}_{\text{received signal}} \cdot \underbrace{A_2 \cos(\omega_2 t)}_{\text{local reference}} \propto A_1 A_2 [\sin(\omega_1 + \omega_2 + \varphi) + \sin(\omega_1 - \omega_2 + \varphi)]$$
$$\propto A_1 \cdot A_2 \cdot \sin(\varphi) \text{ if } \omega_1 = \omega_2$$

- ▶ **I**(dentity) and **Q**(uadrature) coefficients: $A = \sqrt{I^2 + Q^2}$, $\varphi \simeq \arctan(Q/I)$
- ▶ a real signal (modulated carrier) has become complex at the output of the demodulator

Mixer (I/Q)

- ▶ mixing for frequency transposition:

$$\underbrace{A_1 \sin(\omega_1 t + \varphi)}_{\text{received signal}} \cdot \underbrace{A_2 \cos(\omega_2 t)}_{\text{local reference}} \propto A_1 A_2 [\sin(\omega_1 + \omega_2 + \varphi) + \sin(\omega_1 - \omega_2 + \varphi)]$$
$$\propto A_1 \cdot A_2 \cdot \sin(\varphi) \text{ if } \omega_1 = \omega_2$$

- ▶ the “frequency sum” is eliminated by a low-pass filter

- ▶ **I**(dentity) and **Q**(uadrature) coefficients: $A = \sqrt{I^2 + Q^2}$, $\varphi \simeq \arctan(Q/I)$
- ▶ a real signal (modulated carrier) has become complex at the output of the demodulator

Mixer (I/Q)

- ▶ mixing for frequency transposition:

$$\underbrace{A_1 \sin(\omega_1 t + \varphi)}_{\text{received signal}} \cdot \underbrace{A_2 \cos(\omega_2 t)}_{\text{local reference}} \propto A_1 A_2 [\sin(\omega_1 + \omega_2 + \varphi) + \sin(\omega_1 - \omega_2 + \varphi)]$$
$$\propto A_1 \cdot A_2 \cdot \sin(\varphi) \text{ if } \omega_1 = \omega_2$$

- ▶ the “frequency sum” is eliminated by a low-pass filter
- ▶ if $\varphi = 0$, the output is always null, whatever the value of A_1 , A_2
- ▶ use the same signal shifted by 90° as reference signal: $\cos \rightarrow \sin$

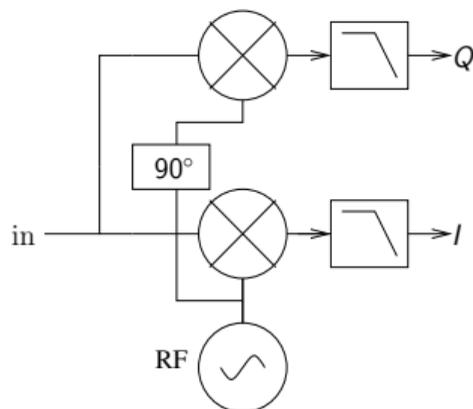
$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \sin(\omega_2 t) \propto A_1 A_2 [\cos(\omega_1 + \omega_2 + \varphi) + \cos(\omega_1 - \omega_2 + \varphi)]$$
$$\propto A_1 \cdot A_2 \cdot \cos(\varphi) \text{ if } \omega_1 = \omega_2$$

- ▶ **I**(dentity) and **Q**(uadrature) coefficients: $A = \sqrt{I^2 + Q^2}$, $\varphi \simeq \arctan(Q/I)$
- ▶ a real signal (modulated carrier) has become complex at the output of the demodulator

Mixer (I/Q)

- ▶ mixing for frequency transposition:

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



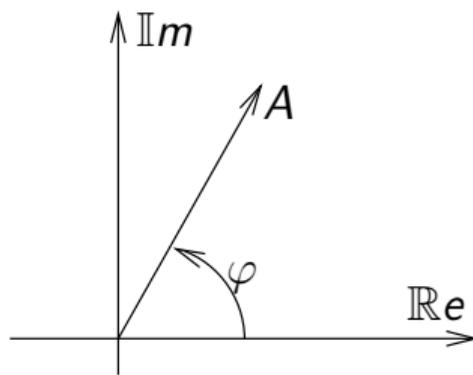
$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \sin(\omega_2 t) \propto A_1 \cdot A_2 \cdot \cos(\varphi) \text{ if } \omega_1 = \omega_2$$

- ▶ coefficients **I**(dentity) and **Q**(uadrature): $A \simeq \sqrt{I^2 + Q^2}$, $\varphi \simeq \arctan(Q/I)$
- ▶ a real signal (modulated carrier) has become complex at the output of the demodulator

Complex number algebra

Trigonometric relations and complex numbers

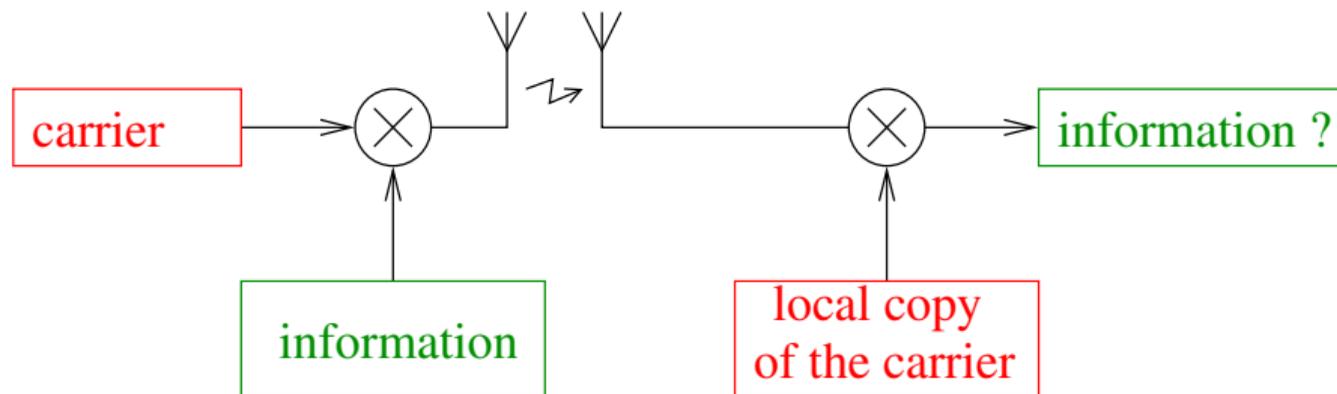
- ▶ $\exp(j \cdot \omega t) = \cos(\omega t) + j \cdot \sin(\omega t)$ where $j^2 = -1$
- ▶ trigonometric identities: $\cos(a) = \frac{1}{2} (\exp(j \cdot a) + \exp(-j \cdot a))$ and $\sin(a) = \frac{1}{2j} (\exp(j \cdot a) - \exp(-j \cdot a))$
- ▶ on the complex plane (*phasor diagram*), the distance to the origin is the magnitude of the complex number, and the angle to the abscissa axis is the phase: $A \cdot \exp(j \cdot \varphi)$.



- ▶ $\cos(a) \cdot \cos(b) = \frac{1}{2} (\cos(a + b) + \cos(a - b))$
- ▶ $(u + jv) \times (w + jx) = (uw - vx) + j(ux + vw)$
- ▶ $\frac{u+jv}{w+jx} = \frac{u+jv}{w+jx} \times \frac{w-jx}{w-jx} = \frac{(u+jv) \times (w-jx)}{w^2+x^2} = \frac{uw+vx}{w^2+x^2} + j \frac{vw-ux}{w^2+x^2}$

Basic principles (modulation)

- ▶ the carrier provides a timing reference to synchronize the emitter and the receiver
- ▶ a RF signal is characterized by 3 quantities (amplitude, frequency, phase), all of which can vary to convey an information
- ▶ on the receiver side, a local oscillator reproduces the carrier, which must be identified to be eliminated in order to extract the useful signal (modulation)



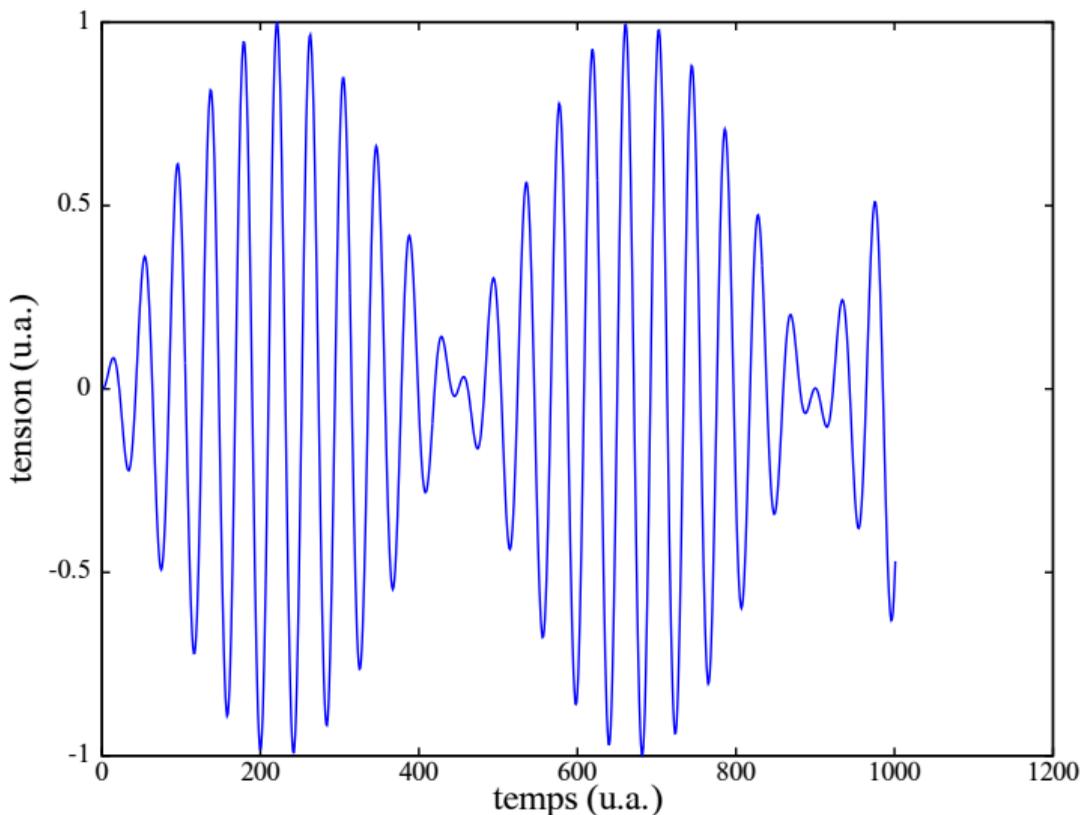
Non-linear processes (multiplication) are the only ones able to sum frequencies.

AM modulation

- ▶ the information is transmitted through the amplitude of the carrier
- ▶ used in airband (108–137 MHz)
- ▶ $y(t) = m(t) \cdot \sin(2\pi \cdot f_c \cdot t) = (1 + M \cos(2\pi \cdot f_m t)) \cdot \sin(2\pi \cdot f_c \cdot t)$
 $m(t) \in [0, 1]$ i.e. M modulation index
- ▶ modulation scheme: variable gain, controlled by a voltage (FET)
- ▶ product of $\sin(f_c t)$ with $\cos(f_m t) \Rightarrow$ 2 modulation sidebands $\pm f_m$ around the carrier f_c

Simulation using GNURadio ^a

^aT. Rondeau, *GNU Radio for Exploring Signals*, FOSDEM 2016, at https://archive.fosdem.org/2016/schedule/event/gnu_radio/

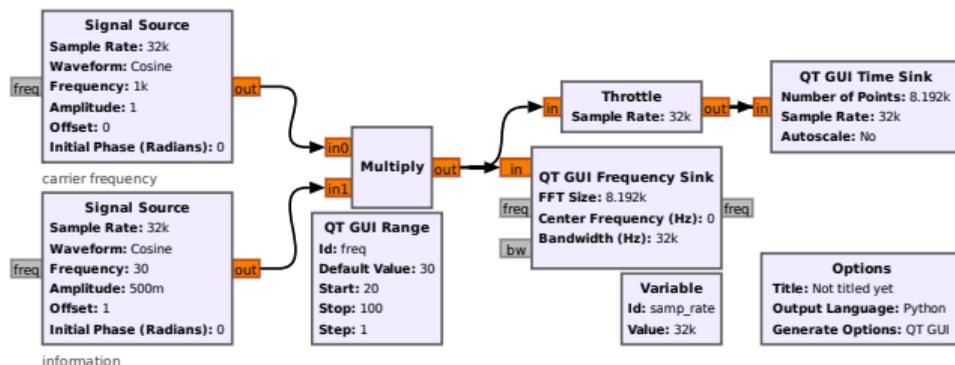
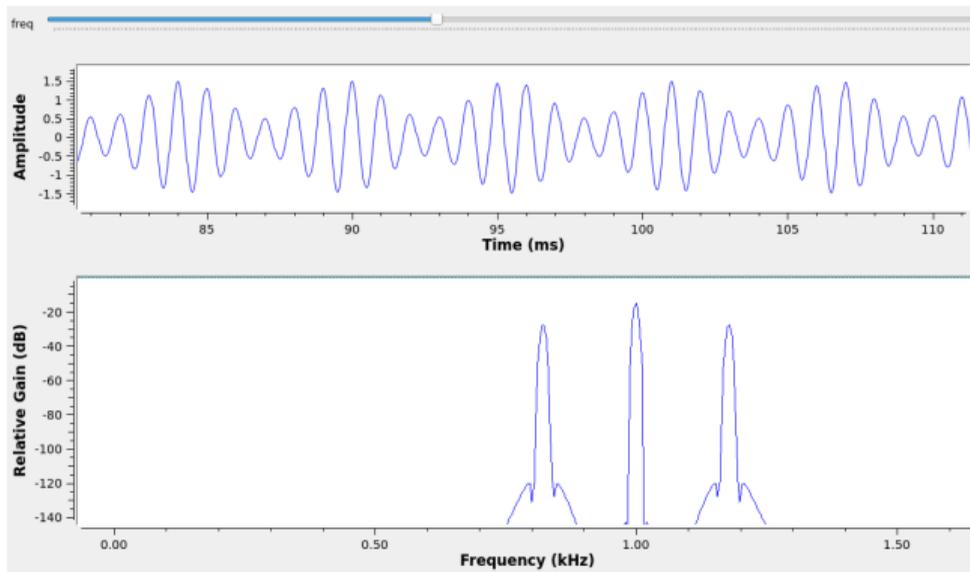


AM modulation

- ▶ the information is transmitted through the amplitude of the carrier
- ▶ used in airband (108–137 MHz)
- ▶ $y(t) = m(t) \cdot \sin(2\pi \cdot f_c \cdot t) = (1 + M \cos(2\pi \cdot f_m t)) \cdot \sin(2\pi \cdot f_c \cdot t)$
 $m(t) \in [0, 1]$ i.e. M modulation index
- ▶ modulation scheme: variable gain, controlled by a voltage (FET)
- ▶ product of $\sin(f_c t)$ with $\cos(f_m t) \Rightarrow$ 2 modulation sidebands $\pm f_m$ around the carrier f_c

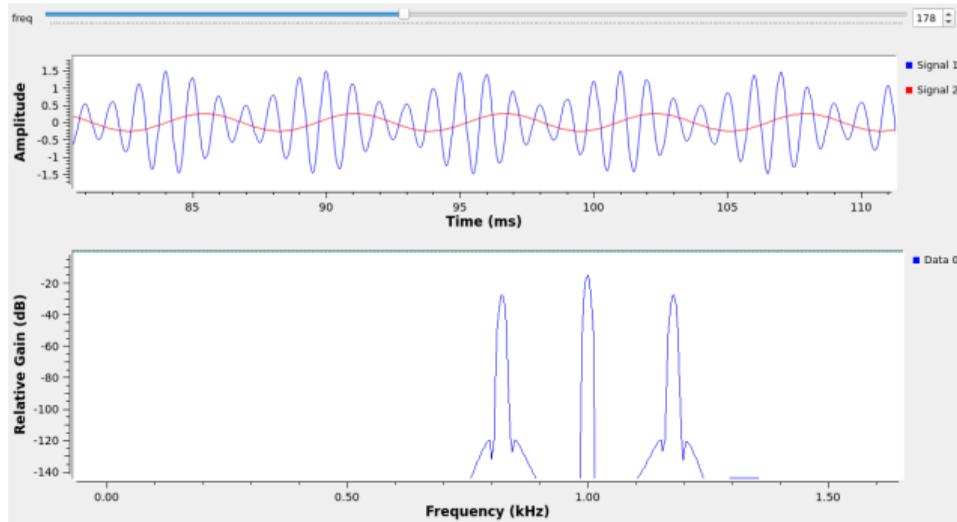
Simulation using GNURadio ^a

^aT. Rondeau, *GNU Radio for Exploring Signals*, FOSDEM 2016, at https://archive.fosdem.org/2016/schedule/event/gnu_radio/

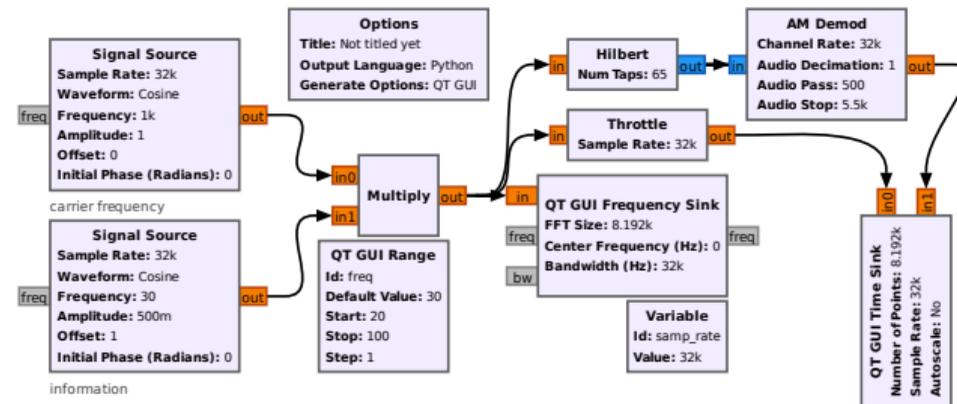


Throttle due to lack of hardware timing peripheral in the flowgraph

AM demodulation



- ▶ Analog AM demodulation: rectifier (diode bridge) + low pass filter
- ▶ Digital: remove DC offset, rectifier (absolute value) + FIR low pass filter
- ▶ Narrowband signal = low redundancy = poor immunity to noise or attenuation in the propagation channel

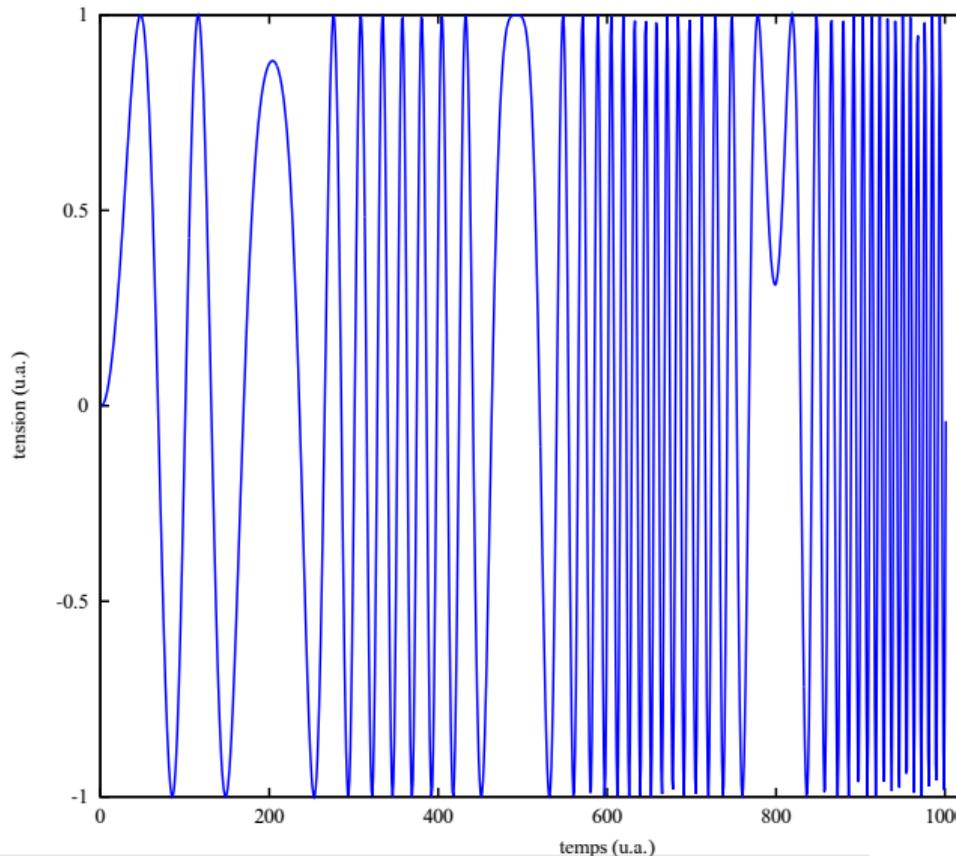


FM modulation

- ▶ information encoded on the *frequency*
- ▶ FSK (Frequency Shift Keying): two frequencies encode the two possible data states (0, 1)
- ▶ the output of the frequency demodulation is a voltage dependent on the carrier frequency offset

- ▶
$$y(t) = \sin \left(2\pi \cdot f_c \cdot t + 2\pi f_{\Delta} \underbrace{\int_0^t x(\tau) d\tau}_{\text{phase} \rightarrow \text{freq}} \right)$$

- ▶ if $x(\tau) = \cos(2\pi \cdot f_m t)$:
$$y(t) = \sin \left(2\pi \cdot f_c \cdot t + \frac{f_{\Delta}}{f_m} \sin(2\pi f_m t) \right)$$
- ▶ principle: tune the control voltage of a VCO
- ▶ broader spectral occupation than AM, but redundancy = robustness



Carson: $B \simeq 2 \cdot (f_{\Delta} + f_m)$, f_{Δ} =deviation, f_m highest spectral component of the signal

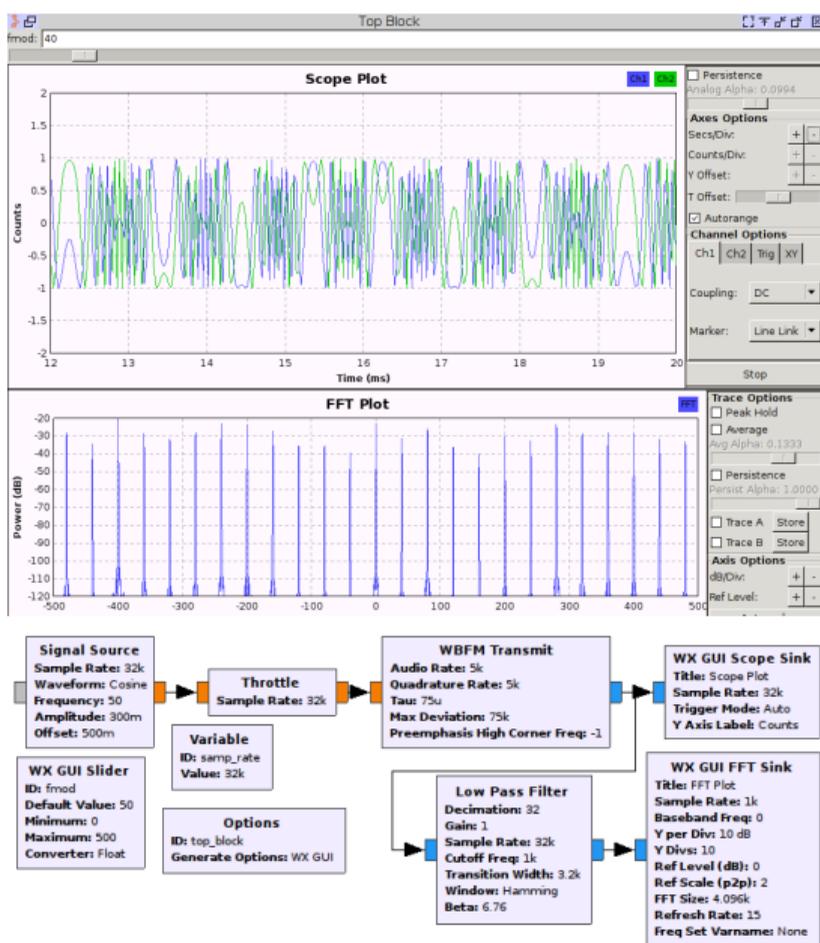
FM modulation

- ▶ information encoded on the *frequency*
- ▶ FSK (Frequency Shift Keying): two frequencies encode the two possible data states (0, 1)
- ▶ the output of the frequency demodulation is a voltage dependent on the carrier frequency offset

$$y(t) = \sin \left(2\pi \cdot f_c \cdot t + 2\pi f_{\Delta} \underbrace{\int_0^t x(\tau) d\tau}_{\text{phase} \rightarrow \text{freq}} \right)$$

- ▶ if $x(\tau) = \cos(2\pi \cdot f_m t)$:

$$y(t) = \sin \left(2\pi \cdot f_c \cdot t + \frac{f_{\Delta}}{f_m} \sin(2\pi f_m t) \right)$$
- ▶ principle: tune the control voltage of a VCO
- ▶ broader spectral occupation than AM, but redundancy = robustness



Carson: $B \simeq 2 \cdot (f_{\Delta} + f_m)$, f_{Δ} =deviation, f_m highest spectral component of the signal

FM demodulation

- ▶ $f_{\Delta} \gg f_m$: WBFM ; $f_{\Delta} \ll f_m$: NFM
- ▶ PLL = frequency to voltage converter
- ▶ FM demodulator = PLL VCO control voltage
- ▶ when VCO is at the carrier frequency, output signal is constant. If VCO is offset with respect to the carrier, output of the mixer is not null and VCO control voltage tracks the frequency offset
- ▶ Digital approach: asebnd I/Q signal is

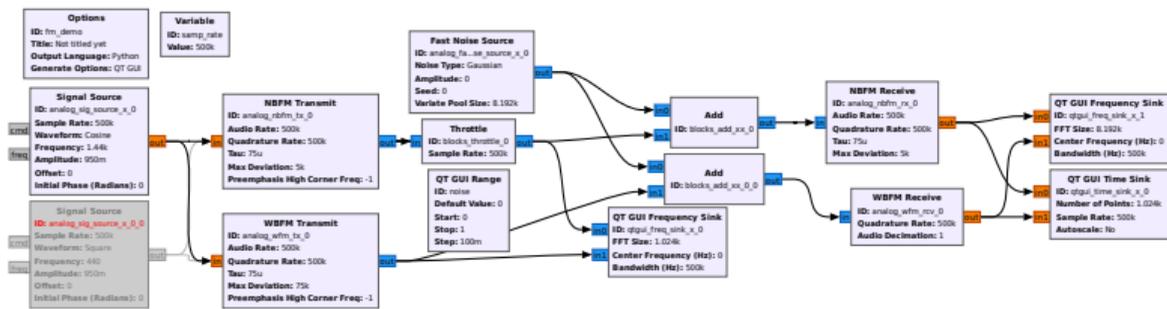
$$X(t) = A \cdot \exp(j\Delta\omega \cdot t + j\varphi(t)) \text{ with } \varphi(t) = D \int_{-\infty}^t m(\tau) d\tau$$

then

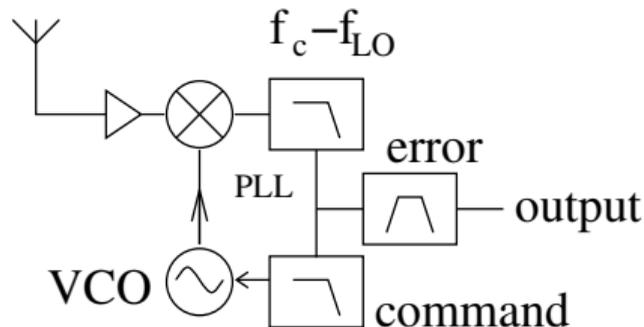
$$X_n \cdot X_{n+1}^* = A \cdot \exp \left(j\Delta\omega \cdot T_s + jD \int_{n \cdot T_s}^{(n+1) \cdot T_s} m(\tau) d\tau \right)$$

and $\int_{n \cdot T_s}^{(n+1) \cdot T_s} m(\tau) d\tau \simeq T_s \cdot m(n \cdot T_s)$ (rectangle) so that $\arg(X_n \cdot X_{n+1}^*) = \underbrace{\Delta\omega \cdot T_s}_{\text{offset}} + \underbrace{D \cdot T_s \cdot m(n \cdot T_s)}_{\text{gain}}$

which is multiplied with $\frac{1}{T_s}$: quadrature FM demodulator implementation of GNURadio ⁵



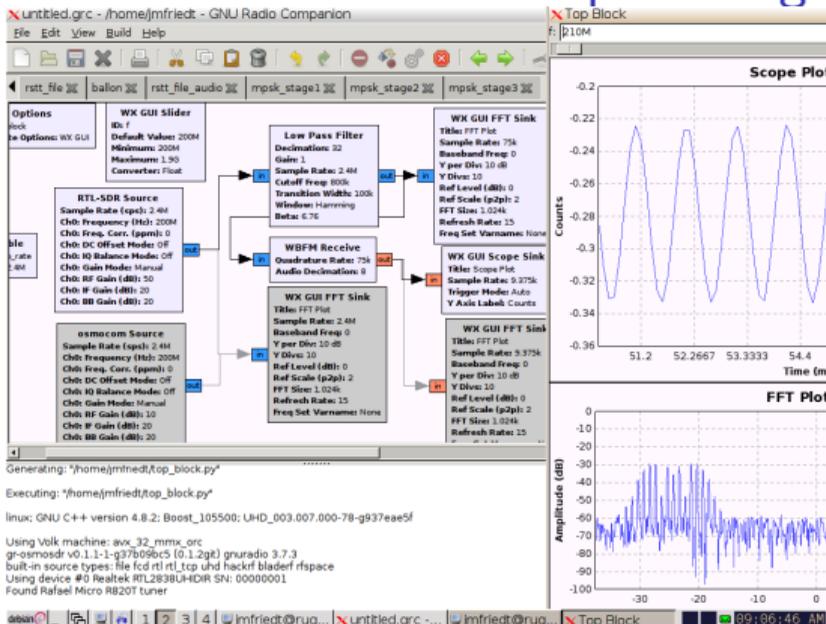
↑ Observe the bandwidth occupied by WBFM and NFM



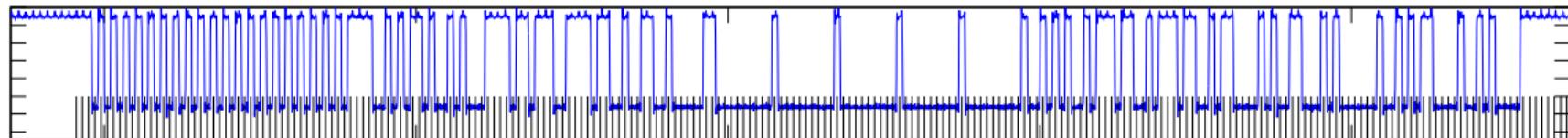
⁵D. Bederov, *Arithmetic based implementation of a quadrature FM Demodulator*, at

https://fosdem.org/2015/schedule/event/sdr_arithmetic/

FM demodulation: example of digital modem



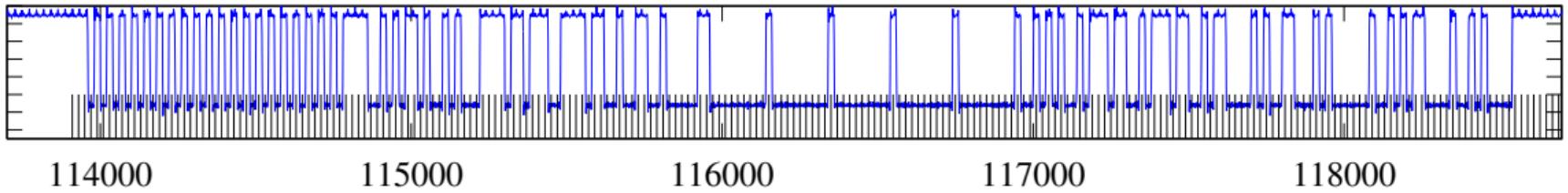
- ▶ Top: spectrum of the demodulated signal (1 kHz)
- ▶ Bottom: spectrum of the modulated signal
- ▶ Binary version: FSK (example: Semtech XE1203 radiomodem)
- ▶ software decoding: threshold + sampling at baudrate



FM demodulation: example of digital modem

Generating: "/home/jmfried/top_block.py"
 Executing: "/home/jmfried/top_block.py"
 linux: GNU C++ version 4.8.2: Boost_105500; UHD_003.007.000-78-g937eae5f
 Using Volk machine: svx_32_mmx_crc
 gr-osmosdr v0.1.1-1-g37b09bc (0.1.2gt) gnuradio 3.7.3
 built-in source types: file fcd rtl_tcp uhd hackrf bladerf rfspice
 Using device #0 Realtek RTL2838U-HDR SN: 00000001
 Found Rafael Micro RB20T tuner

- ▶ Top: spectrum of the demodulated signal (1 kHz)
- ▶ Bottom: spectrum of the modulated signal
- ▶ Binary version: FSK (example: Semtech XE1203 radiomodem)
- ▶ software decoding: threshold + sampling at baudrate



FM demodulation: example of LEO analog weather satellites

Problem: what if the carrier quickly shifts ?

- ▶ AFSK: Audio Frequency Shift Keying – the FM modulation carries an audiofrequency modulation
- ▶ the audio modulation can be a digital information (ACARS)
- ▶ the audio modulation can be an analog information (NOAA POES)

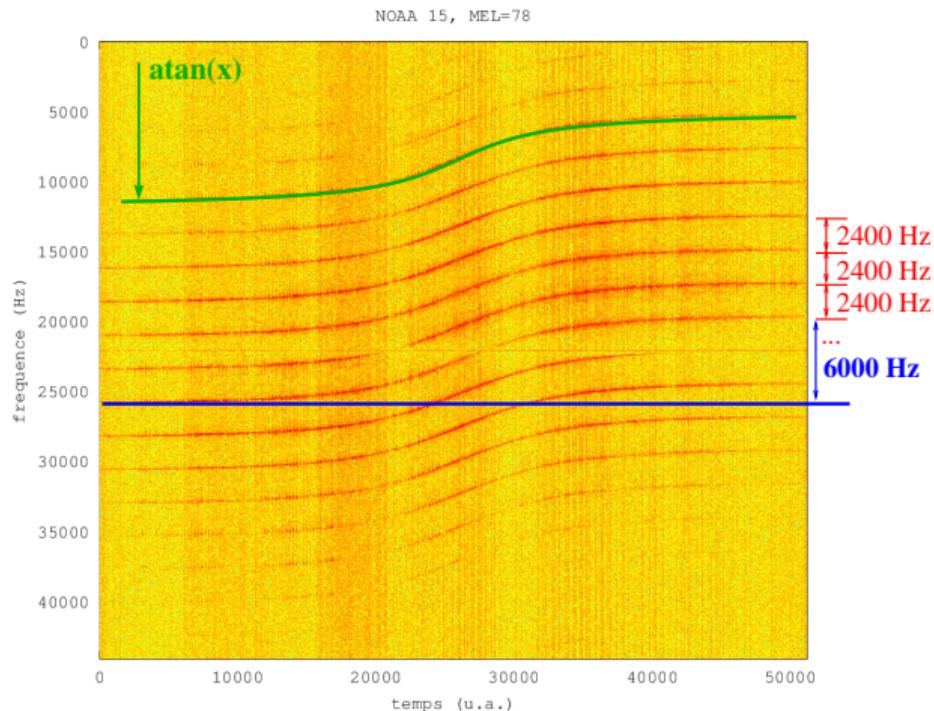
The carrier acts as a reference with respect to which sidebands carry the information

NOAA movie: 160221_noaa.ogv

FM demodulation: example of LEO analog weather satellites

- ▶ AFSK: Audio Frequency Shift Keying – the FM modulation carries an audiofrequency modulation
- ▶ the audio modulation can be a digital information (ACARS)
- ▶ the audio modulation can be an analog information (NOAA POES)

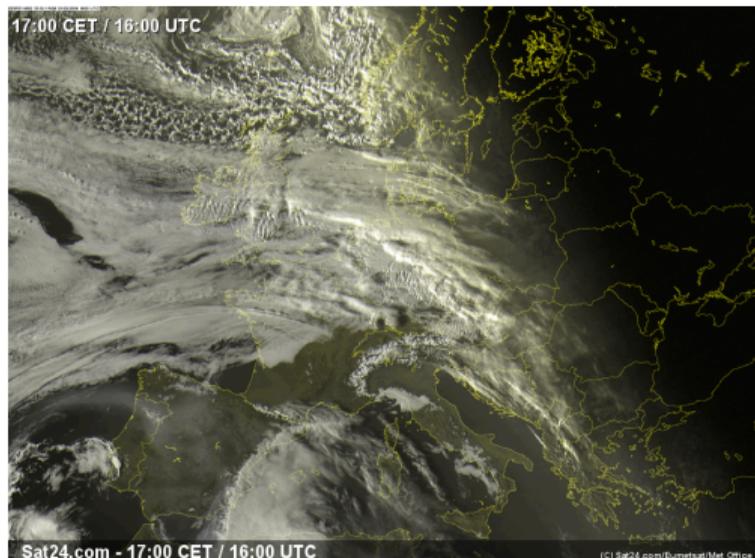
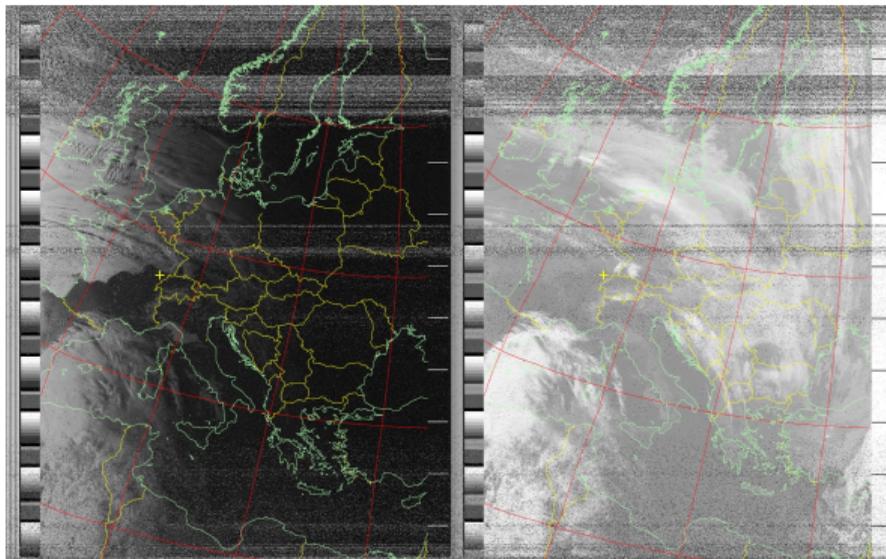
The carrier acts as a reference with respect to which sidebands carry the information



FM demodulation: example of LEO analog weather satellites

- ▶ AFSK: Audio Frequency Shift Keying – the FM modulation carries an audiofrequency modulation
- ▶ the audio modulation can be a digital information (ACARS)
- ▶ the audio modulation can be an analog information (NOAA POES)

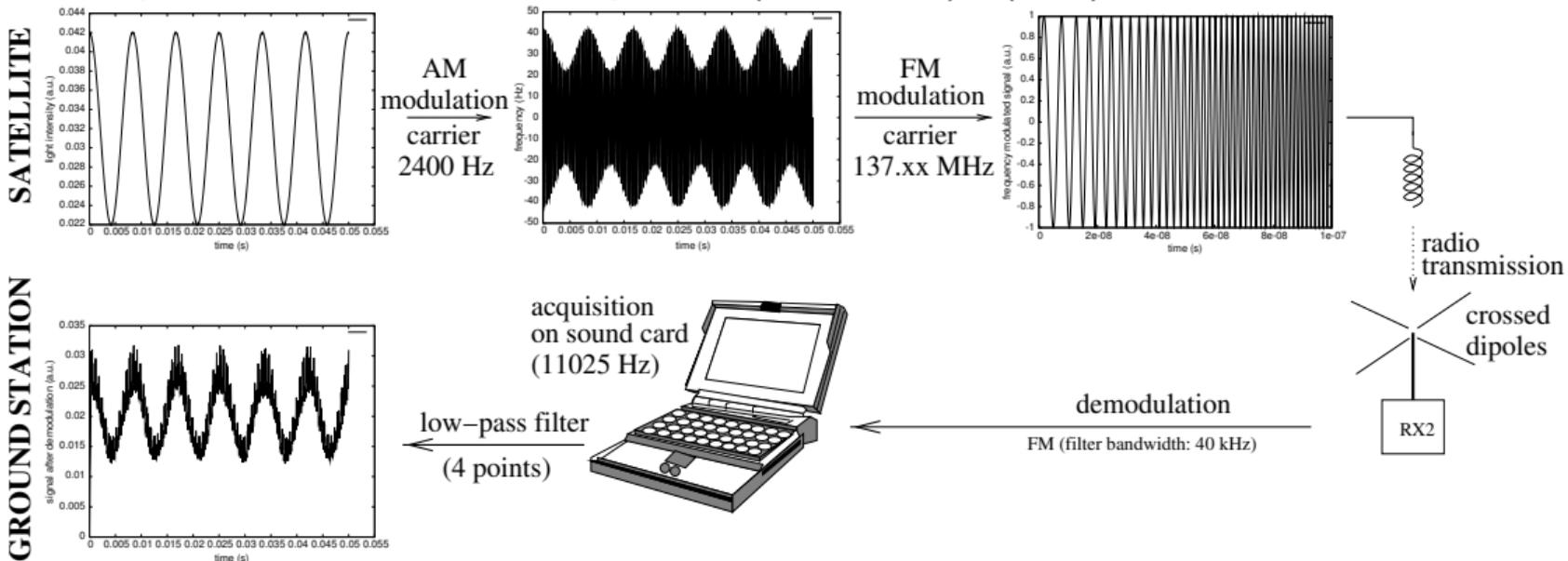
The carrier acts as a reference with respect to which sidebands carry the information



Without AFSK: FM is positively shifted (rising orbit Doppler shift), null and negatively shifted (setting orbit Doppler shift) \Rightarrow FM encoded pixel greyscale would vary along Y-axis

POES modulation scheme ⁷

The intensity of each pixel amplitude-modulates the audiofrequency carrier (2400 Hz) which itself frequency-modulates the radiofrequency carrier (≈ 137 MHz) ⁶ (APT).



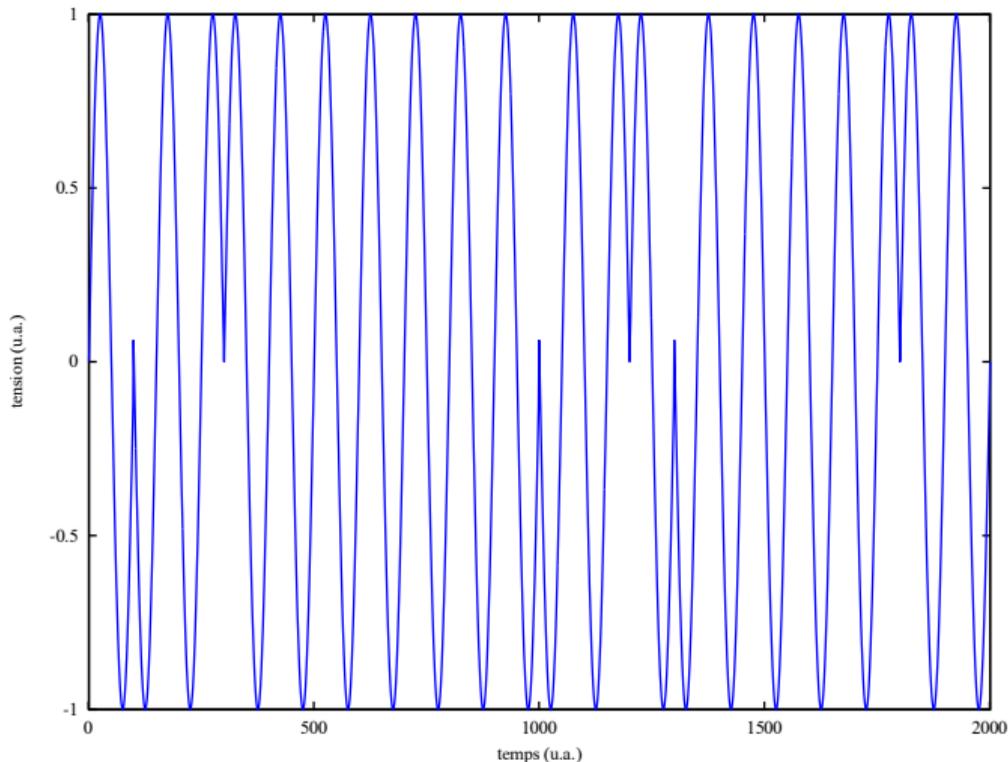
Doppler shift: $v = \frac{40000 + 2\pi \times 800 \text{ km}}{100 \text{ min}} = 7500 \text{ m/s}$ and $\Delta f = f \cdot \frac{v}{c} \cdot \frac{R}{R+r} = 137 \times \frac{7500}{300} \times \frac{6400}{6400+800} = 3040 \text{ Hz}$, $\times 2 \nearrow \searrow \approx 6 \text{ kHz}$

⁶J.-M Friedt, *Satellite image eavesdropping: a multidisciplinary science education project*, European Journal of Physics **26** (August 2005) pp.969-984

⁷Polar Orbiting Environmental Satellites – 800 km

Phase modulation (PM)

- ▶ PSK : Phase Shift Keying
- ▶ AM, FM: incoherent modulation schemes (no need to recover carrier), FM→AM with LPF
- ▶ PM: most efficient use of the spectrum
- ▶ $\varphi = \arctan(Q/I)$: output of the I/Q demodulator
- ▶ local oscillator stability – constellation diagram
- ▶ GPS: BPSK (Binary Phase Shift Keying) – demonstration using a saturated mixer controlled by the bits to be transmitted



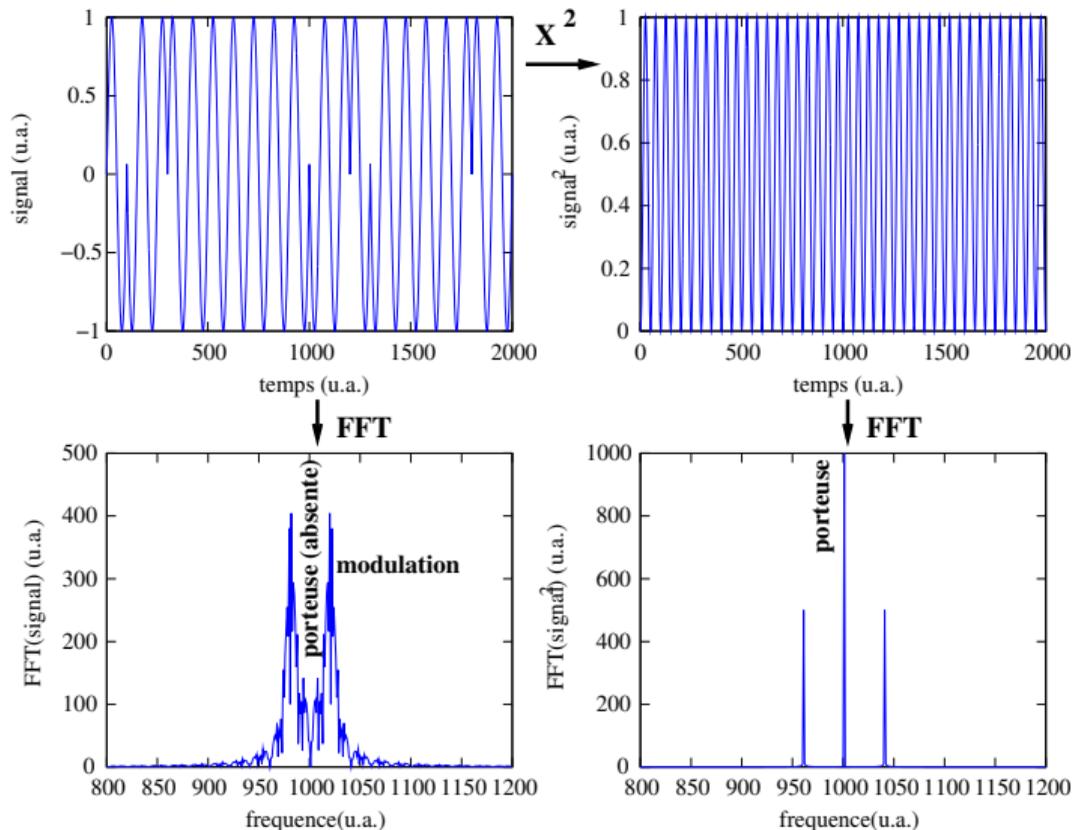
Coherent waveforms have greater noise immunity than non-coherent waveforms, but require more complex demodulation⁸.

⁸J. Mitola, *Software Defined Radio Architecture*, John Wiley & Sons (2000), p.403

Phase demodulation

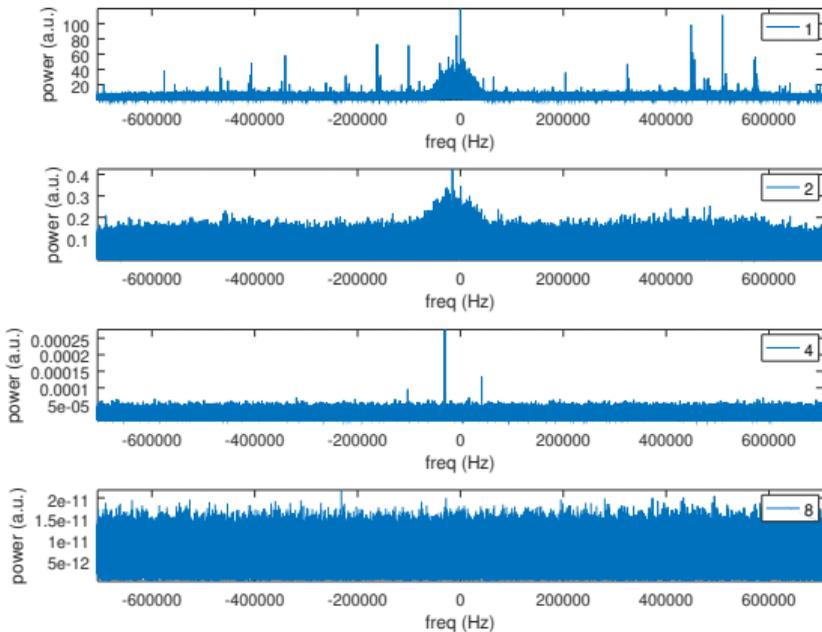
Requires accurate reproduction of the unmodulated carrier

Find N by raising the I/Q signal to various powers until modulation sidebands disappear: **try with the Meteor M2 signal:** jmfriedt.free.fr/extrait_acq.bin using github.com/UpYou/gnuradio-tools/blob/master/matlab/read_complex_binary.m



$$\exp(j(\Delta\omega t + \varphi))^N = \exp(j(N\Delta\omega t + N\varphi)) = \exp(jN\Delta\omega t) \text{ if } \varphi = 2\pi \cdot n/N$$

Meteor M2

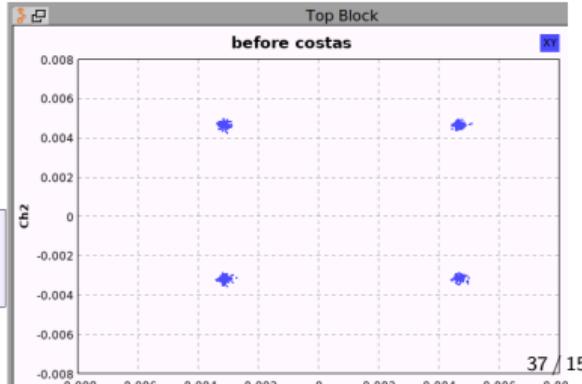
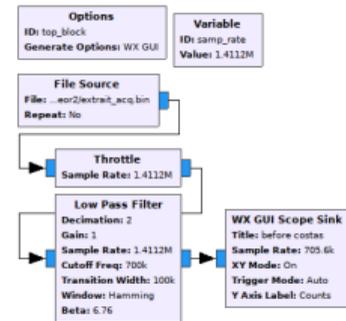


```
x=read_complex_binary('extrait_acq.bin');  
fs=11025*64*2; % 1.4112 MHz  
N=300e3;  
f=linspace(-fs/2, fs/2, N);  
for m=0:3  
    subplot(4,1,m+1);  
    plot(f, fftshift(abs(fft(x(1:N).^ (2^m)))));  
end
```

Tune N depending on SNR (e.g. Wifi – IEEE 802.11)

Constellation diagram: plot IQ samples in the complex plane ($\mathbb{R}, i\mathbb{R}$)

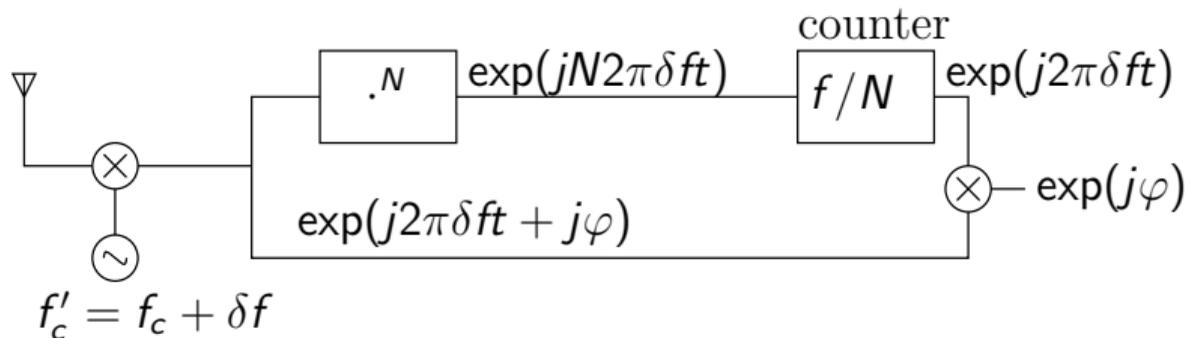
- ▶ poor frequency lock=constellation rotation
- ▶ phase modulation=discrete clusters at different angles
- ▶ amplitude modulation=clusters at different distances from origin (QAM: Quadrature Amplitude Modulation – 16-QAM appears as 4×4 clusters as square constellation)
- ▶ cluster extension given by noise
- ▶ separate clusters allows for associating with symbols, e.g. 00, 01, 10, 11 (Gray coding): soft→hard bit
- ▶ select modulation to maximize distance between clusters



Phase demodulation: carrier recovery

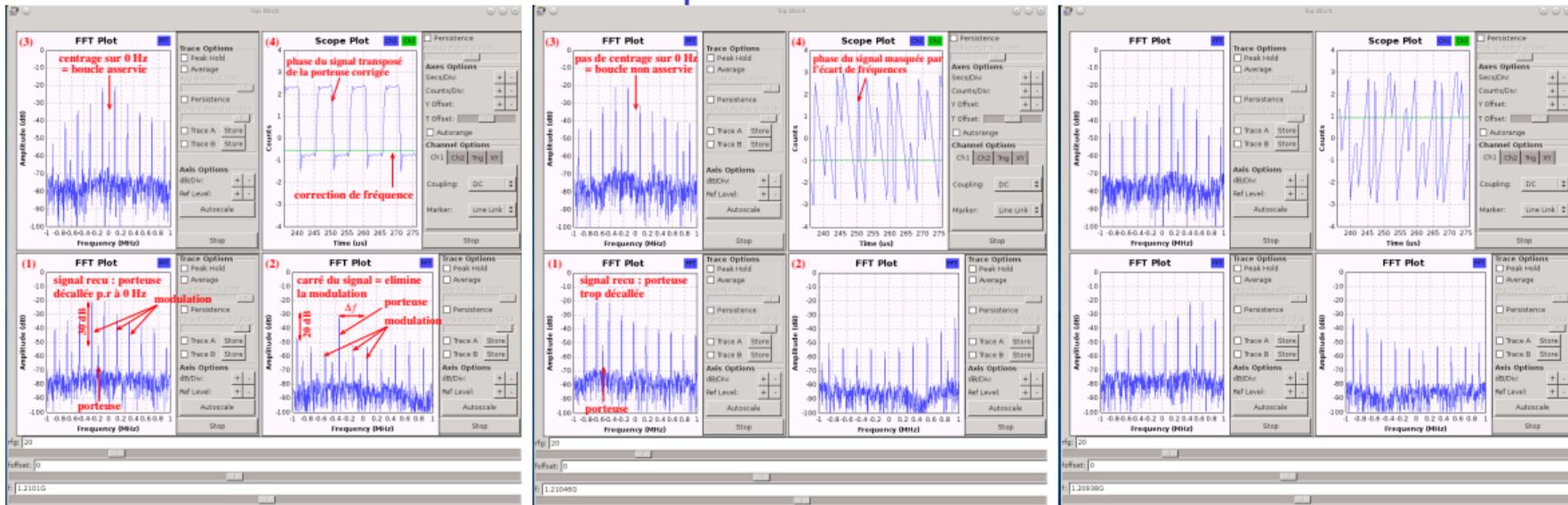
Coherent modulation scheme: an exact copy of the transmitter carrier must be reproduced at the receiver side

Recovery of the clean transmitted carrier as a reference from which the phase is defined:



but $N \times \delta f$ must remain within analysis band.

Phase demodulation: Costas loop



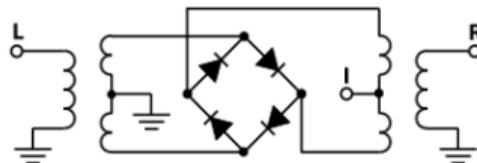
Experimental demonstration with a square wave modulating a BPSK signal (mixer):

Carrier offset: $\Delta\omega$

Modulation: $\varphi = [0; \pi]$

$\sin(\underbrace{\Delta\omega \cdot t + \varphi(t)}_{\text{separate}})$ with $\Delta\omega$ cancelled to recover $\varphi(t)$

Electrical Schematic



Make sure to avoid any DC offset into IF



Example of DAB (QPSK): efficient use of the spectrum (welle.io)

13 broadcast stations within a single 1.537 MHz wide channel, including pictures

The screenshot displays the welle.io DAB receiver interface, organized into several panels:

- Service Overview:** Shows the current service as "VOLTAGE" (CONNECTÉE à PARIS - VOLTAGE) with a DAB+ HE-AAC, 48 kHz Stereo @ 88 kBit/s. A "MOT Slide Show" is active, displaying a red and white logo with the word "voltage".
- Spectrum:** A waterfall plot showing the frequency spectrum. The current channel is centered at 219.664 MHz. A sensitivity slider is visible on the right.
- I/Q RAW Recorder (experimental):** Shows recording parameters: Ring buffer length [s] (240, 5, 10) and Ring buffer size (roughly): 21 MB. An "Init" button is present.
- Constellation Diagram:** A plot of DQPSK Angle [Deg] vs Subcarrier, showing the QPSK modulation points.
- Service Details:** Provides technical information: Device: Realtek, RTL2838UHIDIR, 00000001; Current channel: 11B (218.64 MHz); Frame sync: OK; FIC CRC: OK; Frame errors: 0; Frequency correction: -6139 Hz (28.08 ppm); SNR: 19 dB; RS errors: 0; AAC errors: 0; DAB date and time: Fri Dec 24 07:45:19 2021 GMT.
- Impulse Response:** A plot of Amplitude [..] vs Samples, showing the channel's impulse response.
- Console Output:** A log window showing debug and info messages from the RadioController, including service discovery and audio state changes.
- Null Symbol:** A plot of Amplitude [..] vs Frequency [MHz], showing the null symbol at 219.15 MHz.

At the bottom left, the "Manual channel" is set to "11B".

Example of DAB (QPSK): efficient use of the spectrum (welle.io)

13 broadcast stations within a single 1.537 MHz wide channel, including pictures
Multiplex Besançon local 1, Canal 9A - 202.928 MHz, début : 15/06/2023

Radio	Débit	Niveau de protection
Chante France	88 kbps	EEP 3-A
Chérie FM Besançon	88 kbps	EEP 3-A
Magnum La Radio	88 kbps	EEP 3-A
Plein Air	88 kbps	EEP 3-A
Plein Coeur	88 kbps	EEP 3-A
Radio BIP	88 kbps	EEP 3-A
Radio Campus Besançon	88 kbps	EEP 3-A
Radio Oméga 90.9	88 kbps	EEP 3-A
Radio Star	88 kbps	EEP 3-A
Radio Sud Besançon	88 kbps	EEP 3-A
Villages FM	88 kbps	EEP 3-A
<i>Disponible (ex Radio CAPSAO)</i>	88 kbps	EEP 3-A
<i>Disponible (ex Radio Pitchoun)</i>	88 kbps	EEP 3-A



Image : Arcom

Autorisation mux JORF

Les radios en *italique* démarreront ultérieurement.

Site d'émission

TDF, Fort de Montfaucon, Montfaucon (25)

PAR kW

3.0

Alt. eff. antenne (M)

684

Décision JORF

[Plus d'info](#)

Carte de couverture théorique du multiplex Besançon local 1

<https://www.dabplus.fr/besancon/multiplex/>

Conclusion on modulation schemes

Tradeoff between spectral width and robustness:

1. OOK: on/off, 0 cannot be distinguished from the lack of signal
2. AM: no feedback on the carrier, rectifier + low pass filter
3. FM: VCO frequency control with feedback on the carrier, and information detection as fast frequency fluctuations (PLL + high pass filter – example of compensation for Doppler shift)
4. PM: fine carrier control and phase extraction ($\varphi = \int f \cdot dt$) – coherent phase detection requires fine and *continuous* frequency tracking

Coherent waveforms have greater noise immunity than non-coherent waveforms, but require more complex demodulation⁹.

⁹J. Mitola, *Software Defined Radio Architecture*, John Wiley & Sons (2000), p.403

Spectrum sharing

Many users want to use the electromagnetic spectrum:

1. different times (TDMA ^a): GSM/2G
2. different frequencies (FDMA ^b): commercial FM, GLONASS, FHSS
3. different places/directions (SDMA ^c): laser communication
4. different identifiers (CDMA ^d): GPS

^a Time Division Multiple Access

^b Frequency Division Multiple Access

^c Spatial Division Multiple Access

^d Code Division Multiple Access

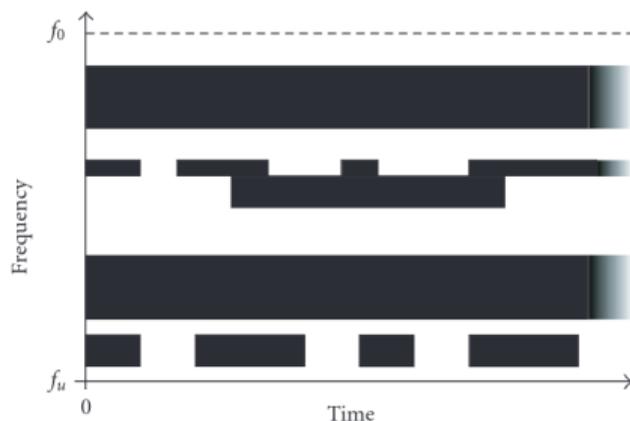


FIGURE 5: FDMA/TDMA signals over the time/frequency plane, spectrum pool.

TABLE 1: Parameters of selected air interfaces.

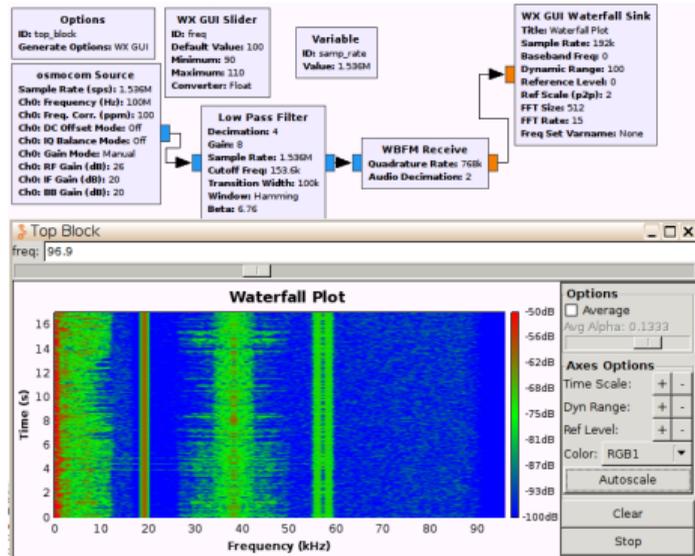
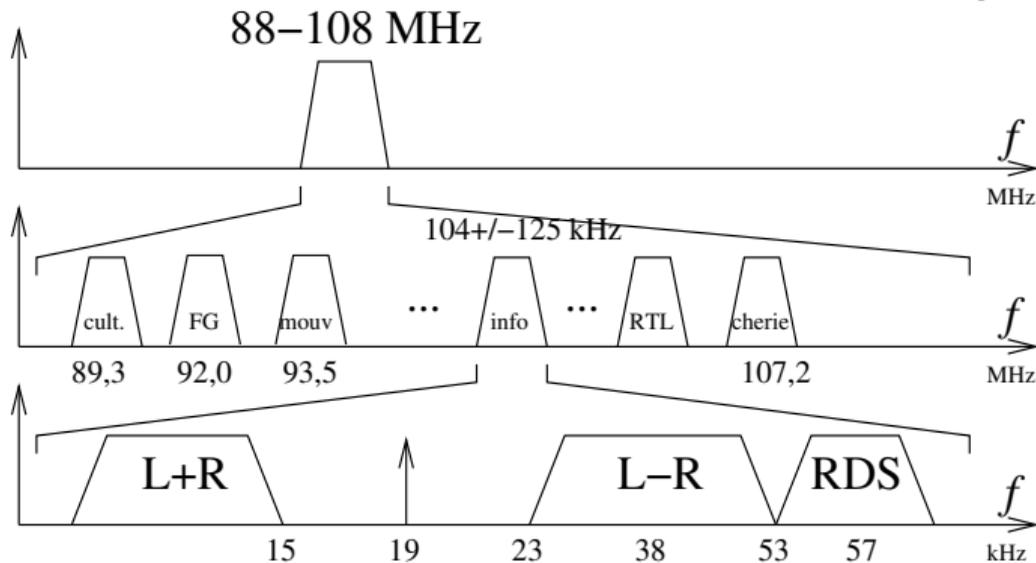
	Bluetooth	DECT	GSM	UTRA-FDD
Frequency range	2.4 GHz (ISM band)	1900 MHz	900, 1800, 1900 MHz	2 GHz
Channel bandwidth	1 MHz	1728 kHz	200 kHz	5 MHz
Access mode	TDMA	FDMA/TDMA	FDMA/TDMA	Direct sequence (DS) CDMA
Duplex mode	TDD	TDD	FDD	FDD
Users per carrier frequency	8 maximum	12	8	—
Modulation	FH sync. to master station, GFSK with modulation index between 0.28 and 0.35	GMSK	GMSK	QPSK
Error correction code	—	No (CRC)	CRC, convolutional	CRC, convolutional, turbo
Bit (chip) rate	1 Mbps	1152 kbps	270.833 kbps	3.840 Mchip/s
Number of bits (chips)/burst (slot)	625	480 (DECT P32)	156.25	2560
Frame duration	—	10 ms	4.615 ms	10 ms
Number of bursts (slots)/frame	—	24	8	15
Burst (slot) duration	0.625 ms	0.417 ms	0.577 ms	0.667 ms

F.K. Jondral, *Software-defined radio—basics and evolution to cognitive radio*, EURASIP J. on Wireless Communications and Networking (2005)

Spectrum sharing: FDMA

Commercial FM band:

- ▶ emitters are allocated a bandwidth centered on a carrier frequency
- ▶ select a station by mixing with the carrier freq. & low-pass filtering
- ▶ select the communication channel with a second mixing within the analyzed bandwidth ^{10 11}



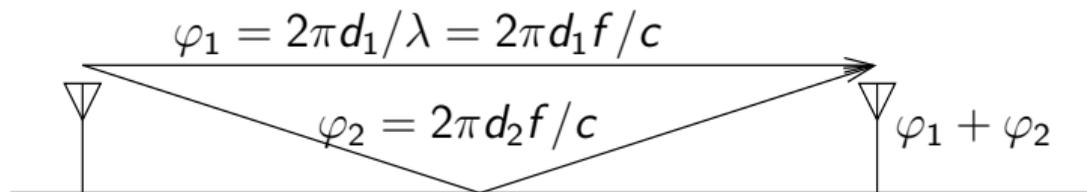
¹⁰B. Bloessl, First Steps in Receiving Digital Information with RDS/TMC, FOSDEM 2015, at fosdem.org/2015/schedule/event/sdr_rds_tmc/

¹¹D. Räisänen, *My Journey into FM-RDS*, 30C3 (2013), at https://media.ccc.de/v/30C3_-_5588_-_en_-_saal_g_-_201312281600_-_my_journey_into_fm-rds_-_oona_raisanen

Spectrum sharing: FHSS

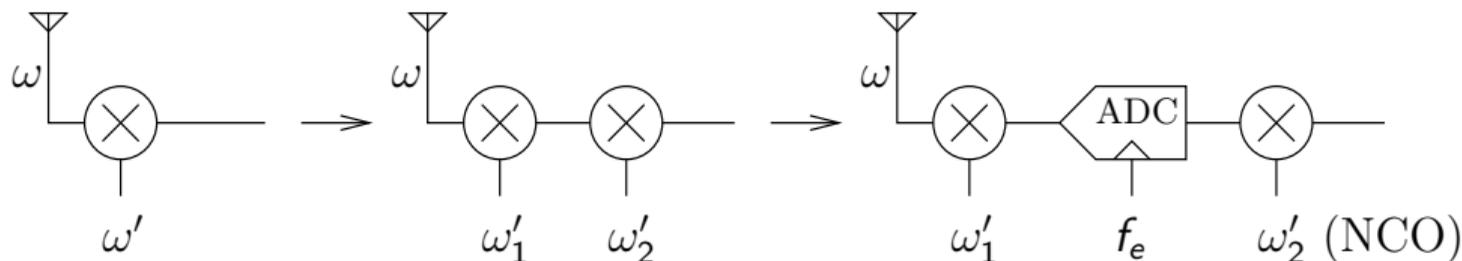
FHSS : Frequency Hopping Spread Spectrum

- ▶ a given frequency band is used by many users
- ▶ these users aim at reducing the risk of interference: frequency hopping
- ▶ the oscillator is too slow to be programmed to a new frequency and stabilize
- ▶ solution: mix with a programmable software-defined oscillator (*NCO, DDS*)
- ▶ used for example in Wi-Fi
- ▶ reduces risks of destructive interferences (channel fade out)



Spectrum sharing: FHSS

- ▶ Technical challenge of frequency hopping: a PLL is slow to stabilize, a DDS slow to configure
- ▶ Two step frequency transposition:
 $\exp(j \cdot (\omega - \omega') \cdot t) = \exp(j \cdot (\omega - \omega'_1 - \omega'_2) \cdot t)$ if $\omega' = \omega'_1 + \omega'_2$
- ▶ We select ω'_1 in the radiofrequency band to bring the signal back to baseband within the bandwidth of the analog-to-digital converter, then ω'_2 below $f_e/2$ to bring the frequency under investigation close to 0 (baseband)
- ▶ classical architecture implemented as a GNURadio processing block: Frequency Xlating FIR Filter

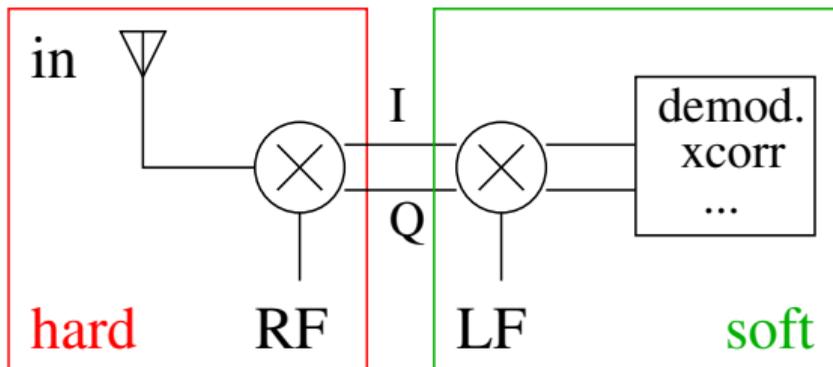


Practical demonstration: simultaneous demodulation of two channels of the commercial FM broadcast band.

Spectrum sharing: FHSS

- ▶ Technical challenge of frequency hopping: a PLL is slow to stabilize, a DDS slow to configure
- ▶ Two step frequency transposition:
 $\exp(j \cdot (\omega - \omega') \cdot t) = \exp(j \cdot (\omega - \omega'_1 - \omega'_2) \cdot t)$ if $\omega' = \omega'_1 + \omega'_2$
- ▶ We select ω'_1 in the radiofrequency band to bring the signal back to baseband within the bandwidth of the analog-to-digital converter, then ω'_2 below $f_e/2$ to bring the frequency under investigation close to 0 (baseband)
- ▶ argument: Python description of a filter ¹²

```
firdes.low_pass(gain,samp_rate,cutoff,transition,firdes.WIN_HAMMING)
```

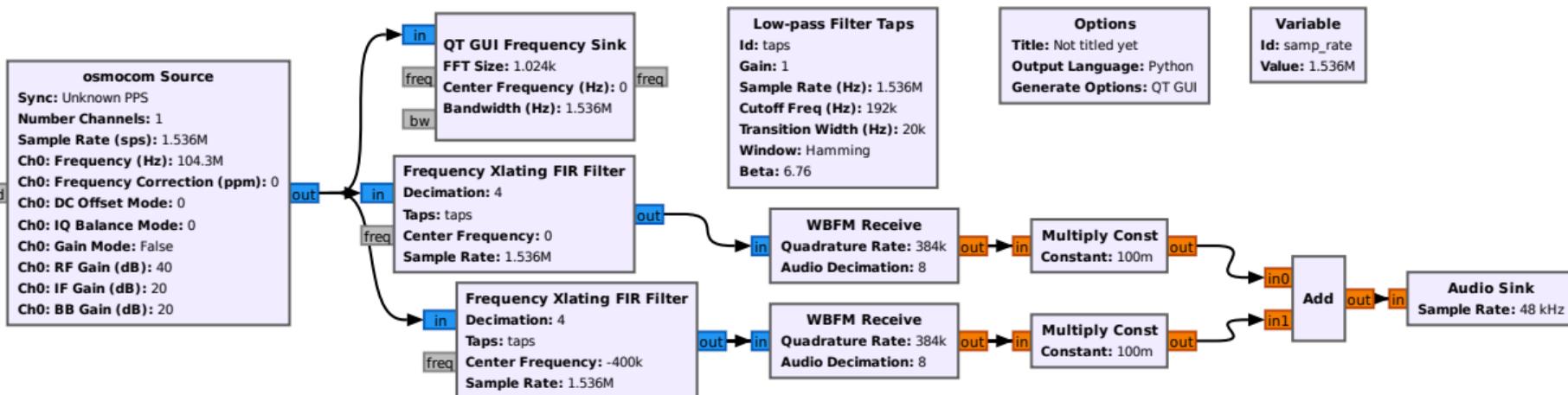


GNURadio's Frequency Xlating FIR Filter: frequency transposition followed by a low-pass filter, using as arguments the frequency offset (NCO) and the cutoff frequency (FIR taps).

¹²https://gnuradio.org/doc/doxygen/classgr_1_1filter_1_1firdes.html

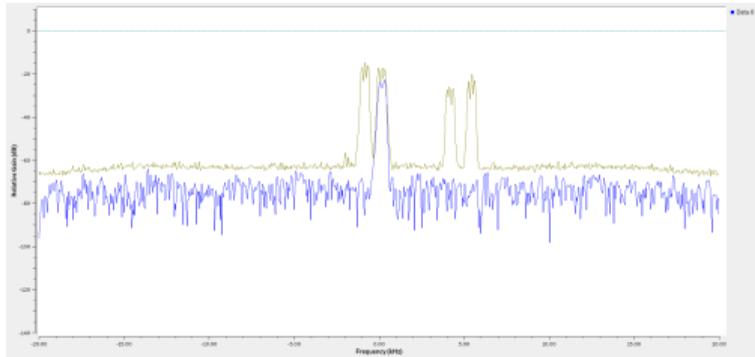
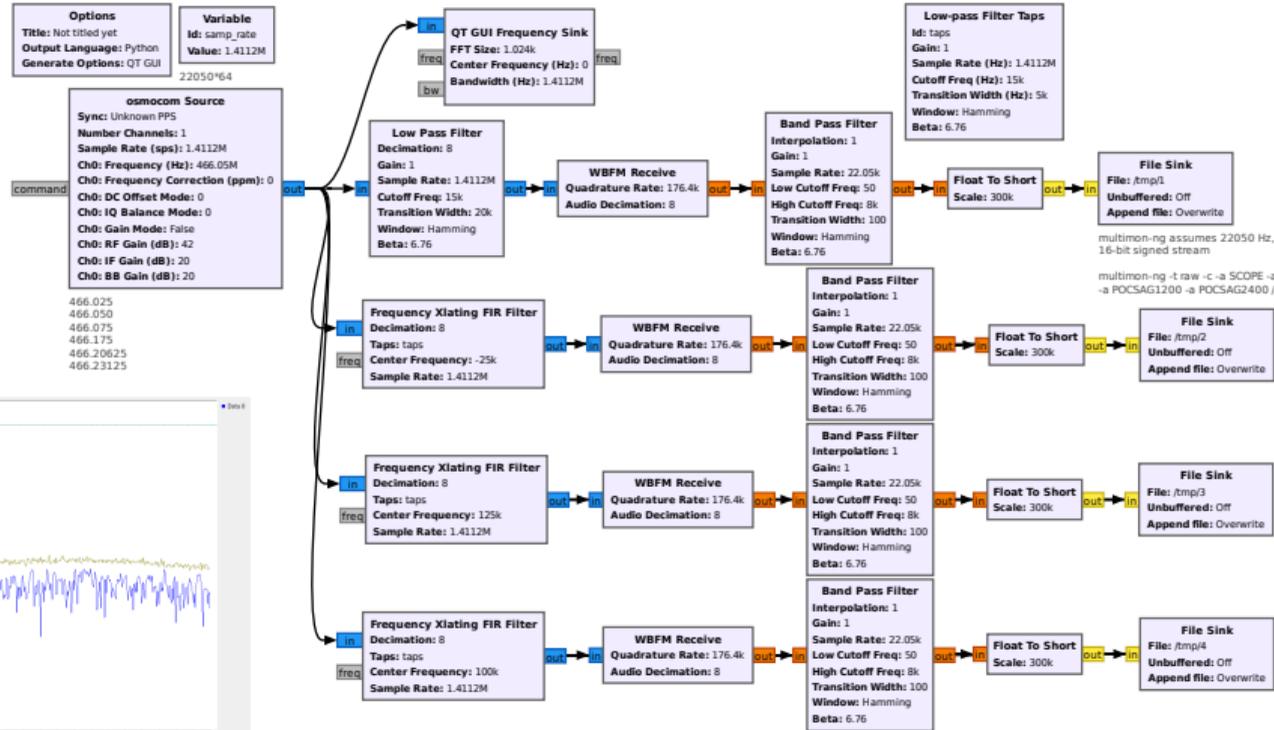
Spectrum sharing: FHSS

- ▶ Technical challenge of frequency hopping: a PLL is slow to stabilize, a DDS slow to configure
- ▶ Two step frequency transposition:
$$\exp(j \cdot (\omega - \omega') \cdot t) = \exp(j \cdot (\omega - \omega'_1 - \omega'_2) \cdot t)$$
 if $\omega' = \omega'_1 + \omega'_2$
- ▶ We select ω'_1 in the radiofrequency band to bring the signal back to baseband within the bandwidth of the analog-to-digital converter, then ω'_2 below $f_e/2$ to bring the frequency under investigation close to 0 (baseband)



FDMA: multichannel analysis

- analyze multiple communication channels at the same time (limited by computational power) – 4 pipes for decoding 4 communication channels
- $\exp(j\omega_{RF} \cdot t) \times \exp(j\omega_{LF} \cdot t) = \exp(j(\omega_{RF} + \omega_{LF}) \cdot t)$, here $f_{RF} = 466.015$ MHz and $f_{LF} \in [0, 25, 115, 150]$ kHz

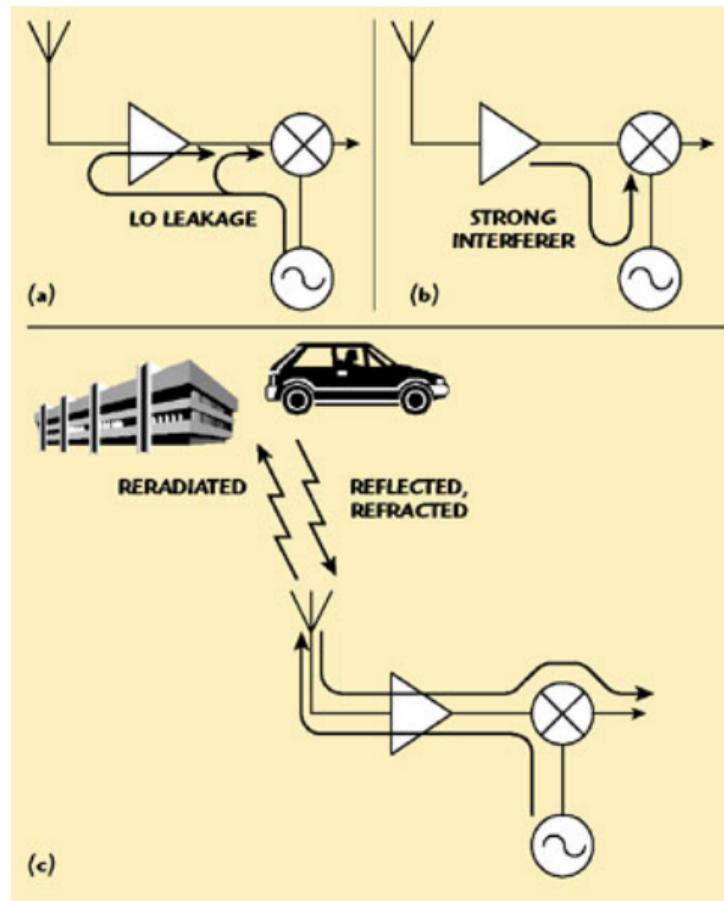


Direct conversion and DC offset

- ▶ Mixer exhibits imperfect isolation while fed on one port with LO ^{a b}
- ▶ Leakage of LO signal into LNA and antenna yielding a signal at mixer input at the same frequency than LO
- ▶ DC offset following mixing, possibly varying with time if antenna impedance varies
- ▶ Solution: “software super-heterodyne” in which LO is shifted with respect to the carrier, and a **Xlating FIR Filter** brings the sampled signal back to baseband (digital software IF).

^aA. Mashhour, W. Domino, N. Beamish, *On the Direct Conversion Receiver – A Tutorial*, Microwave Journal (2001), available at <http://www.microwavejournal.com/articles/3226-on-the-direct-conversion-receiver-a-tutorial>

^bB. Razavi, *Design considerations for Direct-Conversion Receivers*, IEEE Trans. on Circuits and Systems **44** (6), June 1997 (428–)



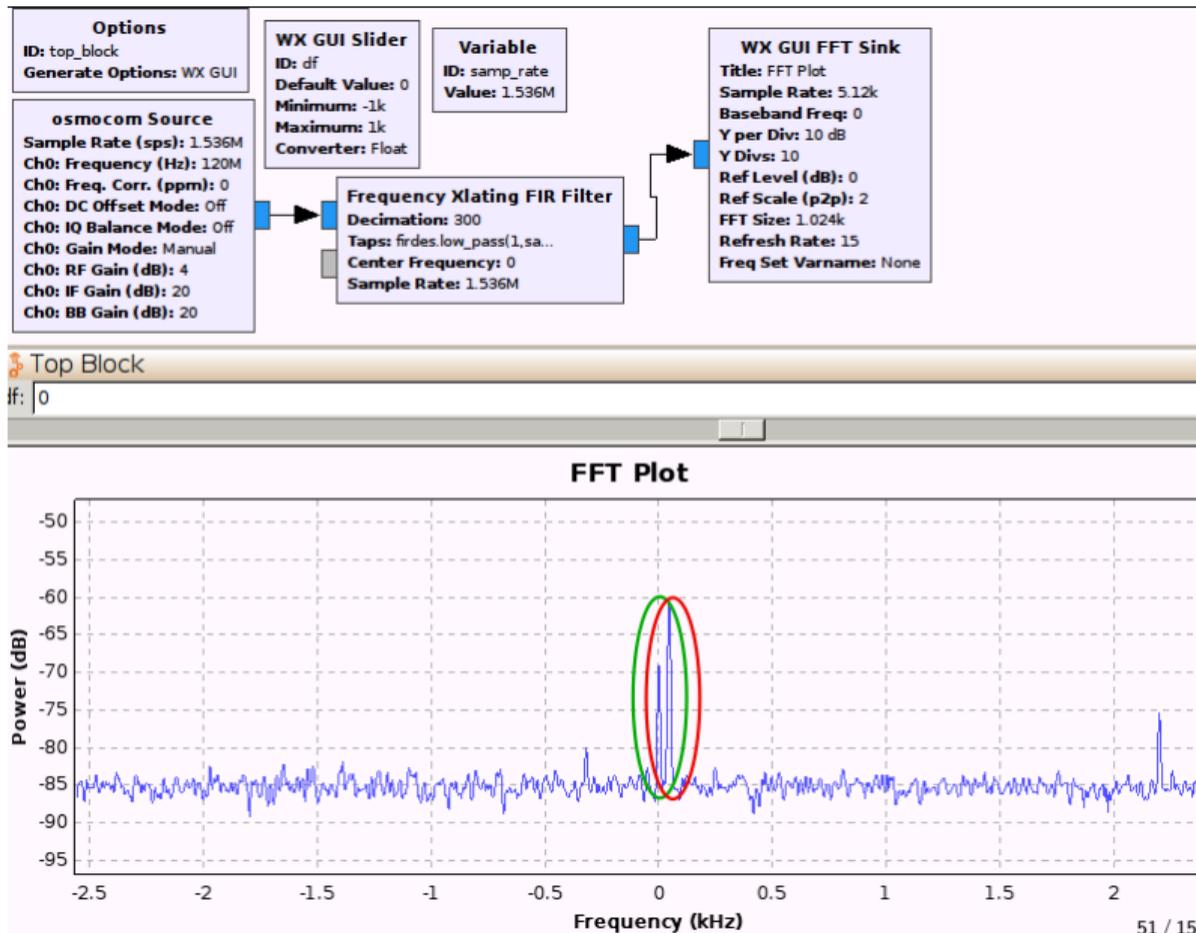
▲ Fig. 8 DC offset mechanisms.

Direct conversion and DC offset

Example: receiving a low power (pure unmodulated carrier) around 120 MHz

Problem of the DC component stronger than the signal to be measured:

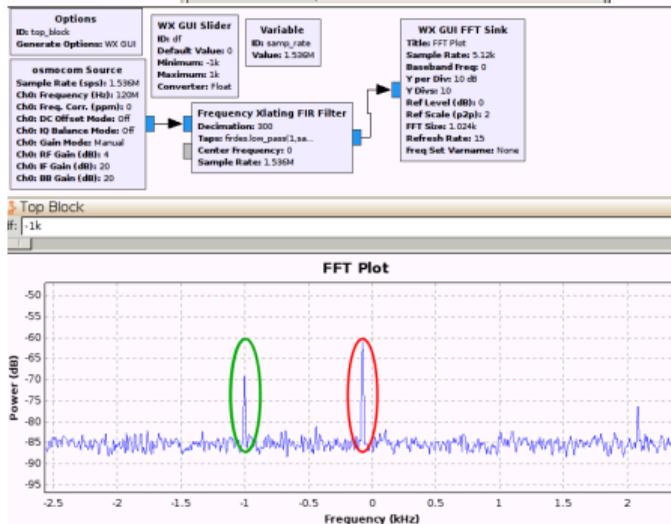
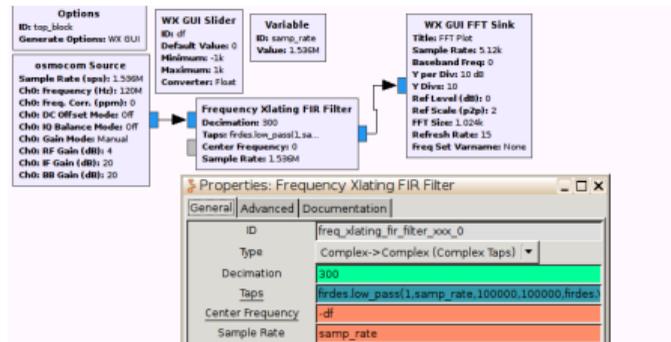
Green=LO leakage, red=wanted signal



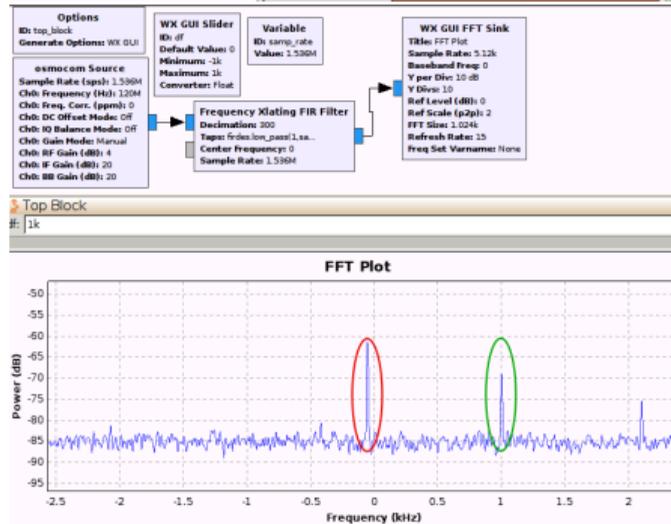
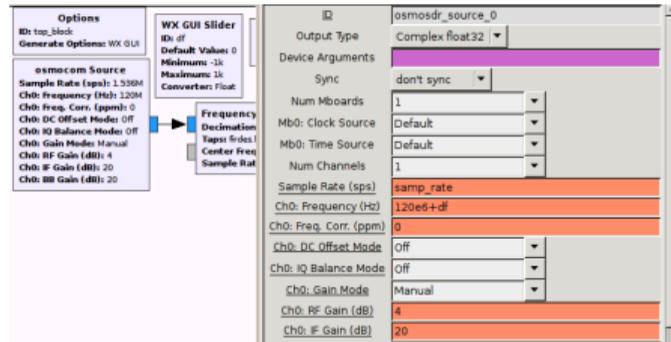
Direct conversion and DC offset

Solution: transpose the signal by Δf by shifting LO to $RF \pm \Delta f$

Effect only visible close to carrier with large decimation factors (e.g. analyzing the carrier characteristics)



-1 kHz



+1 kHz

Spectrum sharing: SDMA

- ▶ Short range: little interference risks (RFID, NFC)
- ▶ Mobile directional antenna
- ▶ Electronic beam steering (destructive/constructive interference in one direction): MIMO
- ▶ Measurement of the time of flight to identify the source position (*time gating*)



Optical link in Ny-Ålesund (2.45 and 5.8 GHz band use prohibited since reserved for radiotelescope deep space measurements).

CDMA: software decoding of GPS

- ▶ GPS: 31-satellite fleet ¹² ¹³ orbiting Earth at a distance of 20000 km
- ▶ Time reference (Cs+Rb and then Rb only)
- ▶ Time of flight computation for positioning
- ▶ Offsets introduced by electromagnetic wave velocity fluctuations (ionosphere, troposphere) impossible to compensate for if a single frequency carrier is monitored
- ▶ Satellite ephemeris + time of flight = position of receiver on Earth
- ▶ Multiple applications beyond positioning ¹⁴ ¹⁵

All satellites transmit on the same carrier frequency

¹²<http://spaceflightnow.com/2014/10/13/gps-modernization-continues-with-quick-pace-of-launches/>

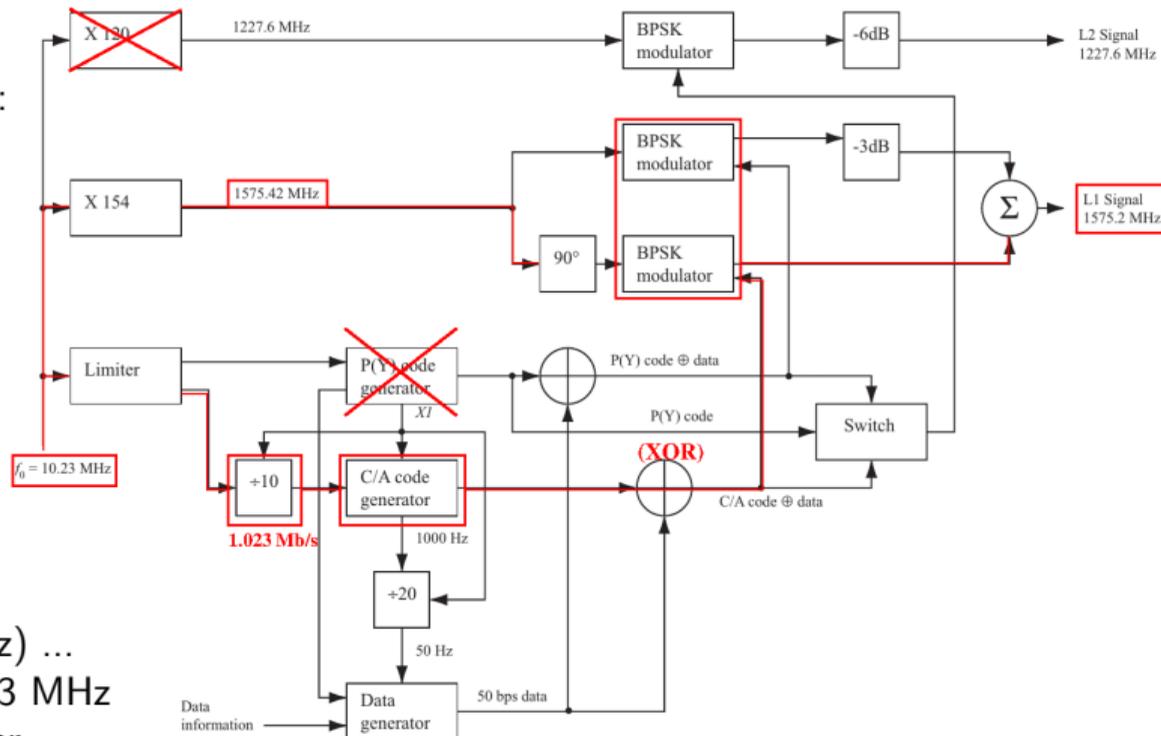
¹³B. Hubert, *How do GPS/Galileo really work & how the galmon.eu monitors all navigation satellites*, MCH (2022), <https://program.mch2022.org/mch2022/talk/QTUAXG/> at <https://www.youtube.com/watch?v=XjRH12Rr9tQ>

¹⁴J.-M Friedt, G. Cabodevila, *Exploitation de signaux des satellites GPS reçus par récepteur de télévision numérique terrestre DVB-T*, OpenSilicium 15, Juil.-Sept. 2015

¹⁵L. Lestarquit et al., *Reflectometry With an Open-Source Software GNSS Receiver: Use Case With Carrier Phase Altimetry*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **9** (10), pp. 4843–4853 (2016)

CDMA: decoding GPS

GPS signal encoding principle¹⁶:



- ▶ the carrier generated by an atomic clock (1575.42 MHz) ...
- ▶ ... is phase modulated at 1.023 MHz with a unique satellite identifier ...
- ▶ ... and again (XOR) phase-modulated with the navigation message (50 bps).
- ▶ On the receiver: known code, so that XORing the received signal with the code pattern returns the NAVigation message ($XOR(x, x) = 0$)

¹⁶K. Borre et al., *A Software-Defined GPS and Galileo Receiver – A Single-Frequency Approach*, Birkhäuser Boston, 2007

CDMA: cross-correlations for decoding GPS

Cross-correlation: search for a (known) pattern $p(t)$ in the received signal $s(t)$.

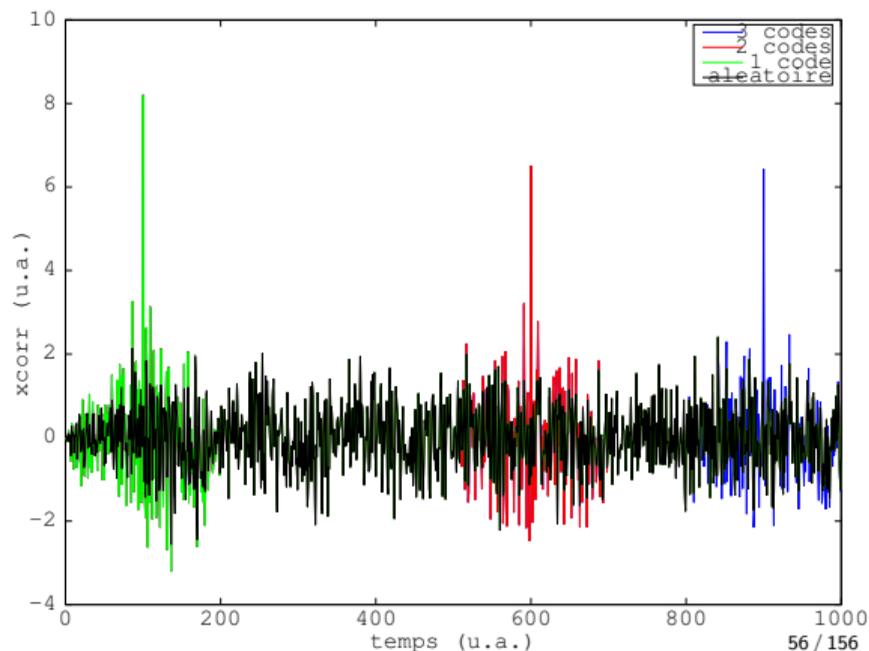
$$xcorr(\tau) = \int_{-\infty}^{+\infty} s(t) \times p^*(t + \tau) dt$$

becoming for discrete time

$$xcorr(n) = \sum_{k=-\infty}^{+\infty} s(k) \times p^*(k + n)$$

Searching for a known pattern in an apparently random sequence (GNU/Octave):

```
rnd=rand(1000,1); rnd=rnd-mean(rnd);  
code=rand(100,1); code=code-mean(code);  
rnd(1:100)=rnd(1:100)+code;  
plot(xcorr(code,rnd))  
rnd(end-99:end)=rnd(end-99:end)+code;  
plot(xcorr(code,rnd))  
rnd(end-500-99:end-500)=\  
    rnd(end-500-99:end-500)+code;  
plot(xcorr(code,rnd))
```



CDMA: cross-correlations for decoding GPS

Cross-correlation: search for a (known) pattern $p(t)$ in the received signal $s(t)$.

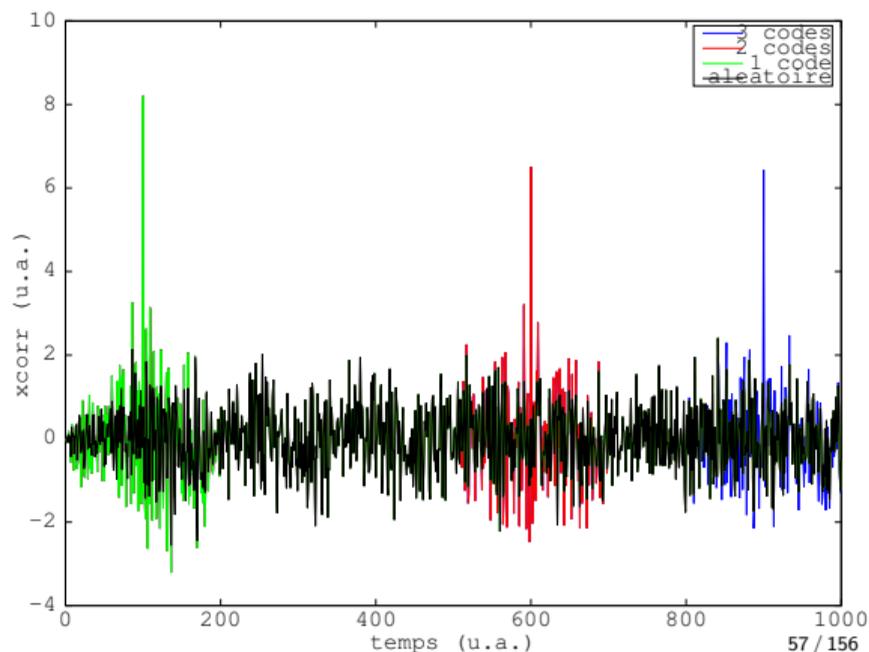
$$xcorr(\tau) = \int_{-\infty}^{+\infty} s(t) \times p^*(t + \tau) dt$$

becoming for discrete time

$$xcorr(n) = \sum_{k=-\infty}^{+\infty} s(k) \times p^*(k + n)$$

Searching for a known pattern in an apparently random sequence (Python):

```
import numpy as np
import matplotlib.pyplot as plt
r=np.random.rand(1000);r=r-np.mean(r)
c=np.random.rand(100); c=c-np.mean(c)
r[100:200]=r[100:200]+c
plt.plot(np.correlate(c,r))
r[-201:-101]=r[-201:-101]+c
plt.plot(np.correlate(c,r))
r[-501-100:-501]=r[-501-100:-501]+c
plt.plot(np.correlate(c,r))
plt.show()
```



CDMA: decoding GPS

Cross-correlation: search for a (known) pattern $p(t)$ in the received signal $s(t)$.

$$xcorr(\tau) = \int_{-\infty}^{+\infty} \underbrace{s(t) \exp(j\varphi)}_{\text{Phase modulated signal}} \times p^*(t + \tau) dt = \exp(j\varphi) \int_{-\infty}^{+\infty} s(t) \times p^*(t + \tau) dt$$

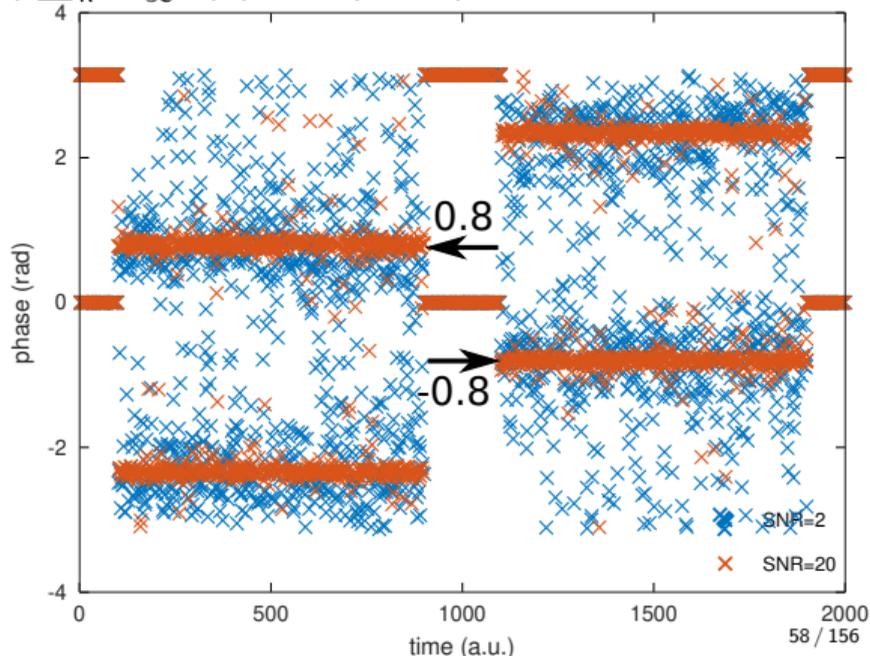
becoming for discrete time

$$xcorr(n) = \sum_{k=-\infty}^{+\infty} \underbrace{s(k) \exp(j\varphi)}_{\text{Phase modulated signal}} \times p^*(k + n) = \exp(j\varphi) \sum_{k=-\infty}^{+\infty} s(k) \times p^*(k + n)$$

Searching for a known pattern in an apparently random sequence (GNU/Octave version):

```
snr=20;  
a=rand(2000,1); a=a-mean(a);  
code=rand(800,1); code=code-mean(code);  
P=[101:900];  
a(P)=a(P)+snr*code*exp(j*0.8);  
P=[1101:1900];  
a(P)=a(P)+snr*code*exp(-j*0.8);  
plot(angle(a),'x')
```

+ phase of the complex signal is transferred to the cross-correlation



CDMA: decoding GPS

Cross-correlation: search for a (known) pattern $p(t)$ in the received signal $s(t)$.

$$xcorr(\tau) = \int_{-\infty}^{+\infty} \underbrace{s(t) \exp(j\varphi)}_{\text{Phase modulated signal}} \times p^*(t + \tau) dt = \exp(j\varphi) \int_{-\infty}^{+\infty} s(t) \times p^*(t + \tau) dt$$

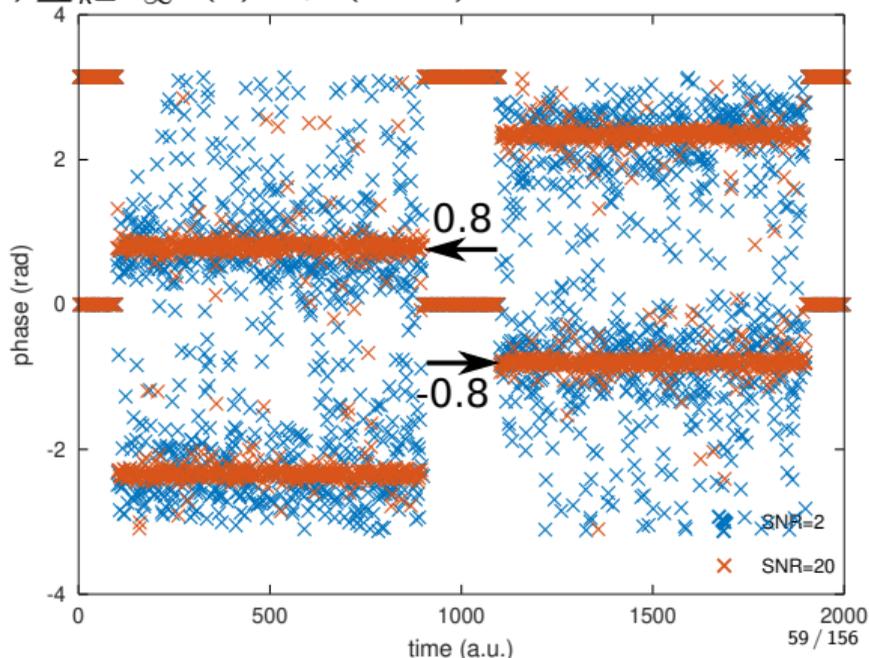
becoming for discrete time

$$xcorr(n) = \sum_{k=-\infty}^{+\infty} \underbrace{s(k) \exp(j\varphi)}_{\text{Phase modulated signal}} \times p^*(k + n) = \exp(j\varphi) \sum_{k=-\infty}^{+\infty} s(k) \times p^*(k + n)$$

Searching for a known pattern in an apparently random sequence (Python version):

```
import numpy as np
import matplotlib.pyplot as plt
snr=20
a=np.random.rand(2000)+1j*0; a=a-np.mean(a)
c=np.random.rand(800)+1j*0; c=c-np.mean(c)
P=np.arange(101,901)
a[P]=a[P]+snr*c*np.exp(1j*0.8)
P=np.arange(1101,1901)
a[P]=a[P]+snr*c*np.exp(-1j*0.8)
plt.plot(np.angle(np.correlate(a,c)), 'x')
plt.show()
```

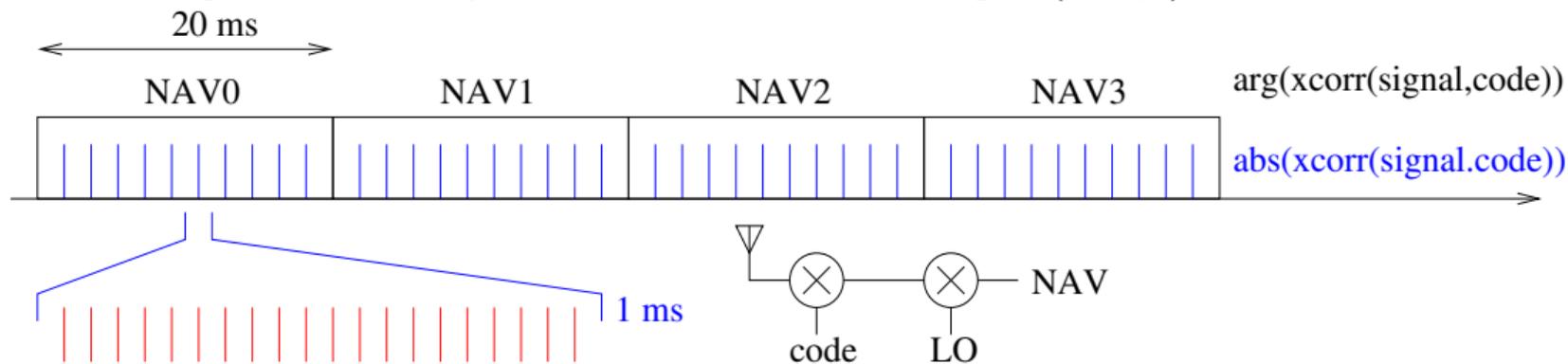
+ **phase** of the complex signal is transferred to the cross-correlation



CDMA: decoding GPS

Modulation steps:

- ▶ the carrier is binary-phase shift keying modulated with the satellite identifier at a rate of 1.023 MHz (phase rotations 0-180°)
- ▶ the message is additionally XORed¹⁷ over the previous signal (50 bps)



$1/1.023 \text{ us}$

- ▶ demodulation for message recovery: first eliminate the code, to identify and eliminate the carrier offset, but code removal requires knowing the carrier !
- ▶ varying carrier offset due to Doppler shift and LO offset
- ▶ brute force search: **what LO offset is acceptable for code correlation?**

¹⁷C. Moore, *Spread Spectrum Satcom Hacking*, DEFCON 23 (2015) <https://www.youtube.com/watch?v=2aBXpho5b7w>

CDMA: decoding GPS

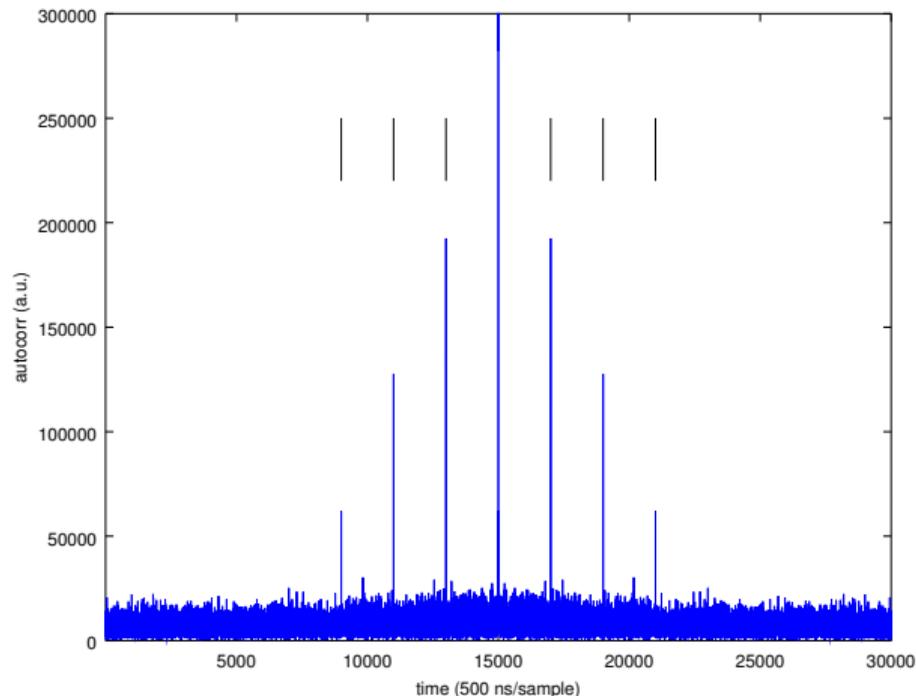
Even if we did not know the GPS encoding scheme, knowing that this code repeats is enough to assess whether a GPS signal is usable, \forall LO offset:

autocorrelation

```
f=fopen('filename.bin');  
d=fread(f,inf,'uchar');fclose(f);  
d=d(1:2:end)-127+i*(d(2:2:end)-127);  
time=[-10000:10000];  
dx=abs(xcorr(d-mean(d),d-mean(d)));  
plot(time,dx(2e6-10000:2e6+10000));  
ylim([0 1e6]) % 2 MHz
```

$xcorr(x, y) = \int x(t) \cdot y^*(t + \tau) dt$ and autocorrelation is
 $xcorr(x, x) = \int x(t) \cdot x^*(t + \tau) dt$ so if x is Doppler shifted PRN code $p(t)$:
 $xcorr(x, x) = \int p(t) \exp(j2\pi\delta f t) \cdot p^*(t + \tau) \exp(j2\pi\delta f (t + \tau)) dt = \exp(j2\pi\delta f \tau) \int p(t) \cdot p^*(t + \tau) dt = \exp(j2\pi\delta f \tau) xcorr(p, p)$

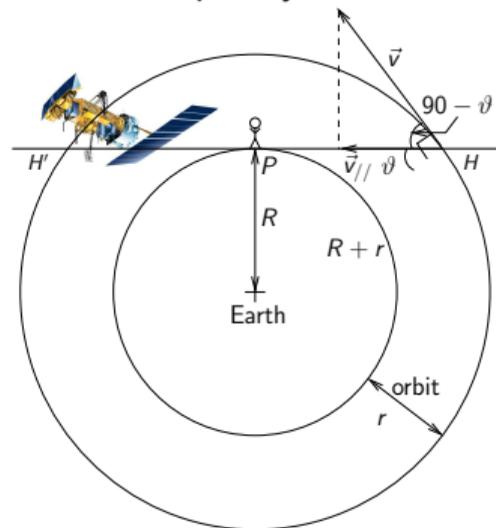
$$|xcorr(x, x)| = |xcorr(p, p)| \quad \forall \delta f$$



Repetition every 1 ms at 2 MS/s \Rightarrow max(autocorr) every 2000 samples

CDMA: decoding GPS

- ▶ Decoding GPS is *only* possible if the carrier frequency is accurately known ...
- ▶ ... which can only be identified after removing the code from the received signal !
- ▶ Initial **exhaustive** (*Acquisition*) search of all possible codes and frequency offsets (brute force) for later only *tracking* satellites known to be visible.
- ▶ What frequency offset should we look for ?



Doppler shift: $(R+r) = 20000 + 6400$ km in 12 h ($T^2/R^3 = \text{cst}$)

$\Rightarrow |\vec{v}| = 3830$ m/s

Since $\sin(\theta) = \frac{R}{r+R}$ or $R \simeq 6400$ km

$\Rightarrow |\vec{v}_{//}| = |\vec{v}| \cos(90 - \theta) = |\vec{v}| \sin(\theta) = |\vec{v}| \frac{R}{r+R}$

Result: $|\vec{v}_{//}| \in [\pm 4880]$ Hz

+ local oscillator contribution (bias and random fluctuations) !

Application: decode an acquired signal, using the GPS pseudo-random code generator available at

fr.mathworks.com/matlabcentral/fileexchange/14670-gps-c-a-code-generator/

Basics on GPS encoding

1. CDMA (Code Division Multiple Access): all satellites transmit on the same frequency and their messages are encoded with individual orthogonal codes (Gold Codes)
2. Satellite identification: $xcorr(signal, code)$
3. Code orthogonality: $xcorr(code_i, code_j) = \delta_{i,j}$
4. Doppler shift: need to compensate for remote clock frequency wrt ground clock & local clock offset wrt remote atomic clocks

Intensive use of correlations ¹⁸

$$xcorr(x, y)(\tau) = \int x(t)y^*(t + \tau)dt$$

or through the convolution theorem: $FFT(xcorr(x, y)(\tau)) = FFT(x) \cdot FFT^*(y)$

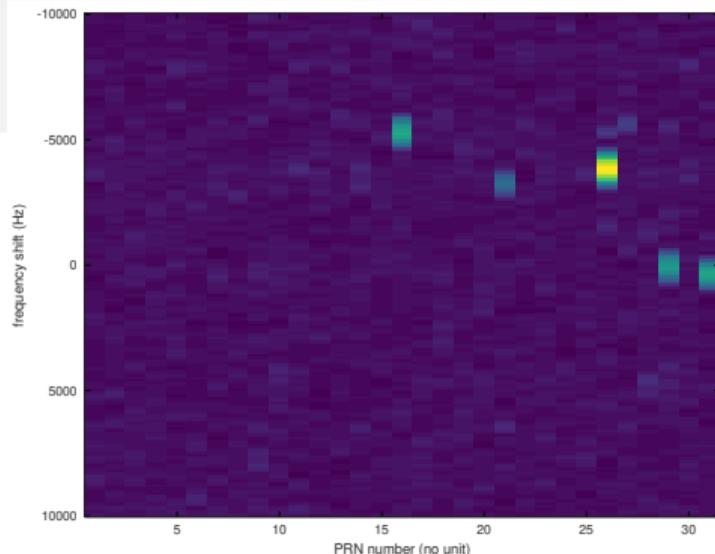
¹⁸Time-domain implementation on FPGA allows for pipelined computation as samples are collected

Basics on GPS encoding

GPS acquisition in 10 lines of Matlab program ¹⁹ (two nested loops – satellite number and frequency)

```
pkg load signal
x=read_complex_binary(filename,1024*128); fs=1.023; % sampling rate in MHz
x=x-mean(x);
freq0=[-10.5e3:500:10.5e3]; % Doppler range
time=[0:1/fs/1e6:length(x)/fs/1e6]';time=time(1:end-1);
for m=[1:31] % loop on all satellites
    a=cacode(m,fs/1.023); a=a-mean(a);
    l=1;
    for freq=freq0 % loop on all frequency offsets
        mysine=exp(j*2*pi*(-freq)*time);
        xx=x.*mysine; % frequency shift the signal
        [u(l,m),v(l,m)]=max(abs(xcorr(a,xx,'none'))); % find xcorr max.
        l=l+1;
    end
end
end
```

- ▶ Orbital mechanics: $Doppler \in [-5000, 5000]$ Hz
- ▶ Map $xcorr$ max as a function of space vehicle number and frequency shift
- ▶ When a satellite is visible, sharp $xcorr$ peak when frequency offset is compensated for



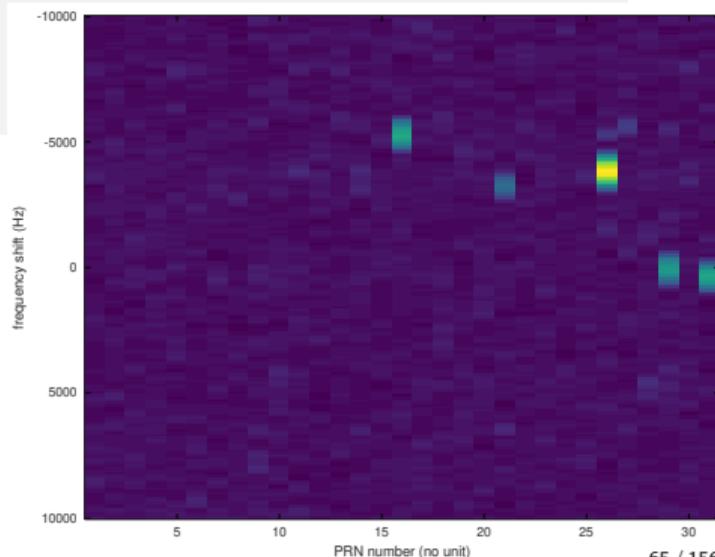
¹⁹ using the C/A code generator <https://www.mathworks.com/matlabcentral/fileexchange/14670-gps-c-a-code-generator>

Basics on GPS encoding

GPS acquisition in 10 lines of Matlab program (single loop on space vehicle number)

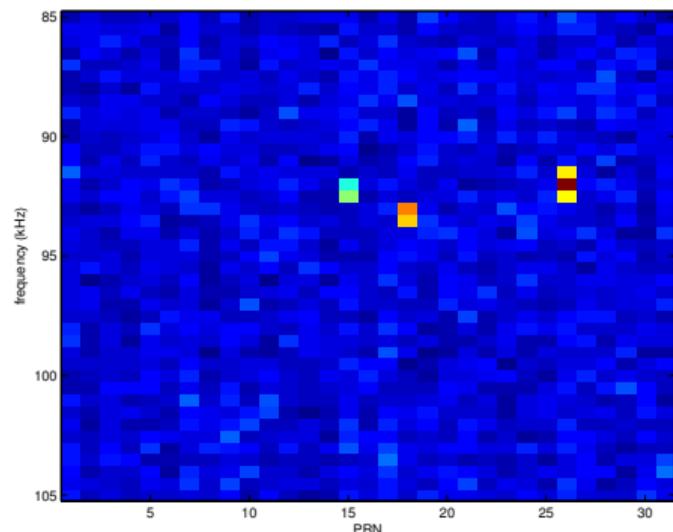
```
pkg load signal
x=read_complex_binary(filename,1024*128); fs=1.023; % sampling rate in MHz
x=x-mean(x);
freq0=[-10.5e3:500:10.5e3]; % Doppler range
time=[0:1/fs/1e6:length(x)/fs/1e6]';time=time(1:end-1);
% doppler frequency shift matrix whose FFT is computed
doppler=exp(j*2*pi*freq0*time'); % 43x131072 matrix
data=ones(43,1)*x';
all=doppler.*data; % Doppler-shifted data
allf=fft(all');
for m=[1:31] % loop on all satellites
    a=cacode(m,fs/1.023); % CA code of satellite m
    a=[a zeros(1,length(all)-length(a))]; % zero padding
    a=a-mean(a);
    pattern=ones(43,1)*a; % 43x131072 matrix
    af=fft(pattern');
    correlation=ifft(af.*conj(allf))';
end
```

- ▶ Replace loops (inefficient) with matrix multiplication
- ▶ Parallelizing the frequency operations halves the computation time



CDMA: decoding GPS

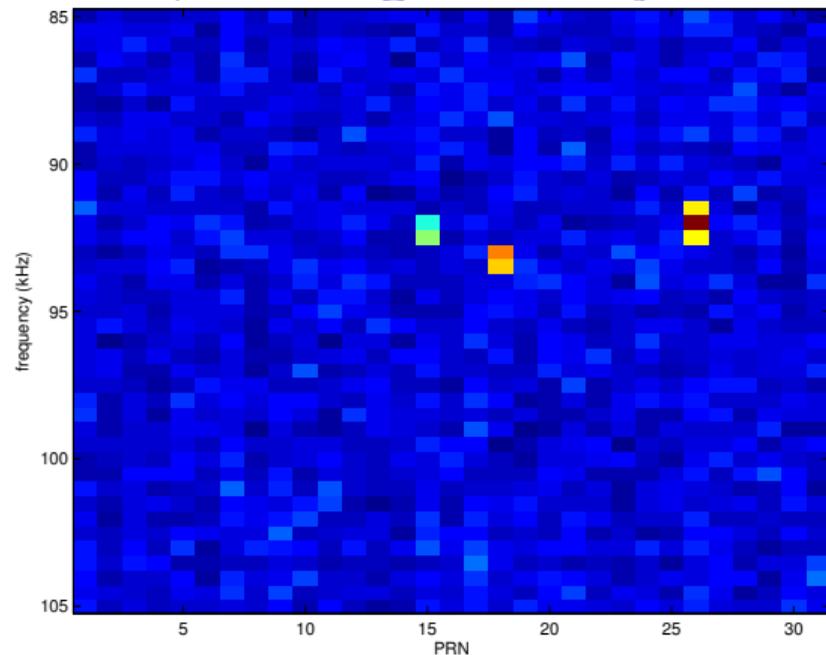
- ▶ CDMA basics: each useful bit (*navigation data*) is transmitted with its associated satellite identifier (PRN).
- ▶ All satellites transmit on the same carrier (1575.42 MHz), only their unique identifier allows differentiating each source.
- ▶ Each identifier is repeated every millisecond, NAV is at 50 bps so 20 samples/bit.



GNU/Octave v.s. gnss-sdr: PRN 15, 18, 26 visible

```
Searching for GPS Satellites in L1 band...
[ ... . 15 . . . 18 . . . . . 24 . 26 . . . . ]
Total signal acquisition run time 11.5126 [seconds]
Reference Time:
  GPS Week: 811
  GPS TOW: 483088 38647.040000
  ~ UTC:   Fri Mar 13 15:11:29 2015
Current TOW obtained from SUPL assistance = 483089
Reference location (defined in config file):
Latitude=47.3 [o]
Longitude=6 [o]
Altitude=10 [m]
Doppler analysis results:
SV ID Measured [Hz] Predicted [Hz]
  15  93437.50  1420.63
  18  94500.00  2337.54
  24  95000.00  2886.61
  26  93250.00  (Eph not found)
Parameters estimation for Elonics E4000 Front-End:
Sampling frequency =1999883.21 [Hz]
IF bias present in baseband=91995.03 [Hz]
Reference oscillator error =-58.39 [ppm]
Corrected Doppler vs. Predicted
SV ID Corrected [Hz] Predicted [Hz]
  15  1442.47  1420.63
  18  2504.97  2337.54
  24  3004.97  2886.61
  26  1254.97  (Eph not found)
GNSS-SDR Front-end calibration program ended.
```

CDMA/decoding GPS: why do we need accurate oscillators ?



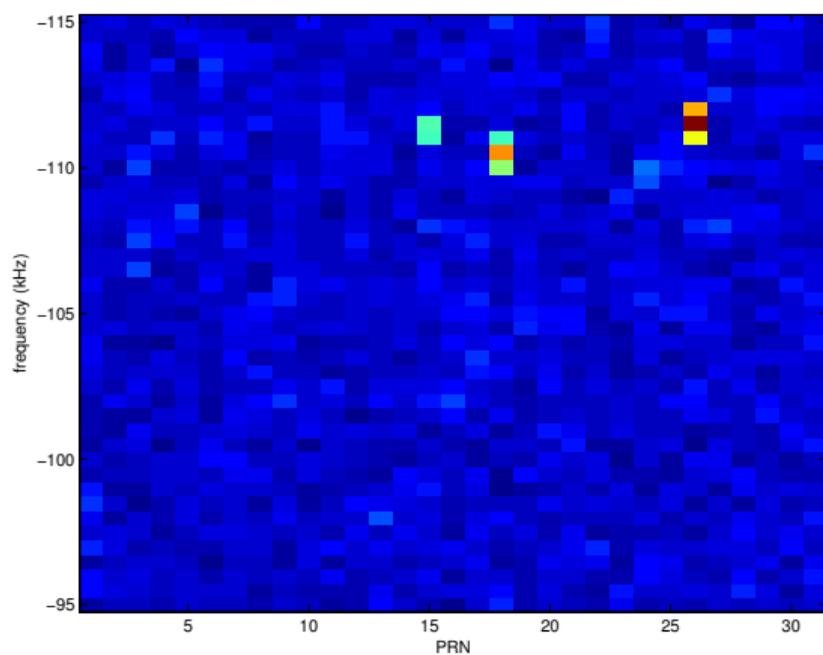
E4000 DVB-T

+59 ppm bias

= +91 kHz at 1575.42 GHz

Instead of searching a ± 5 kHz range (Doppler) with 500 Hz steps, need for ± 150 kHz range search

\Rightarrow computation time multiplied by 30 ! 20 kHz range with 500 Hz steps on $2 \cdot 10^5$ samples: 302 seconds with Matlab R2010, 342 seconds with GNU/Octave 3.8.2



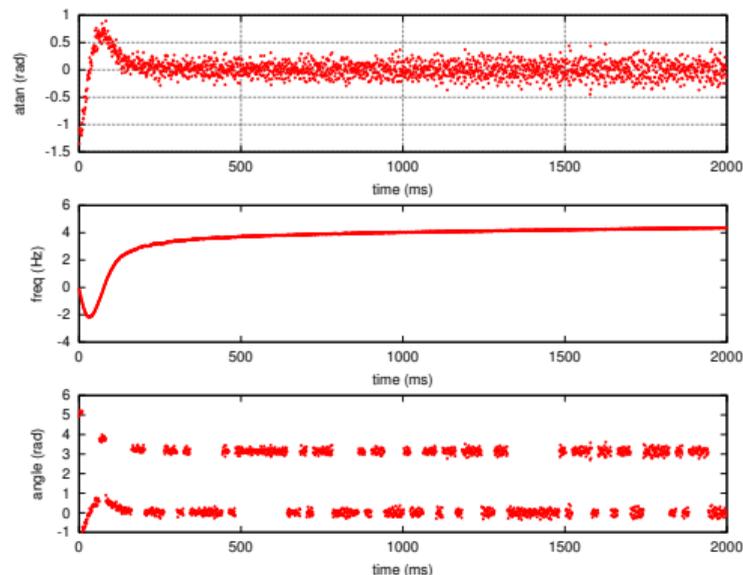
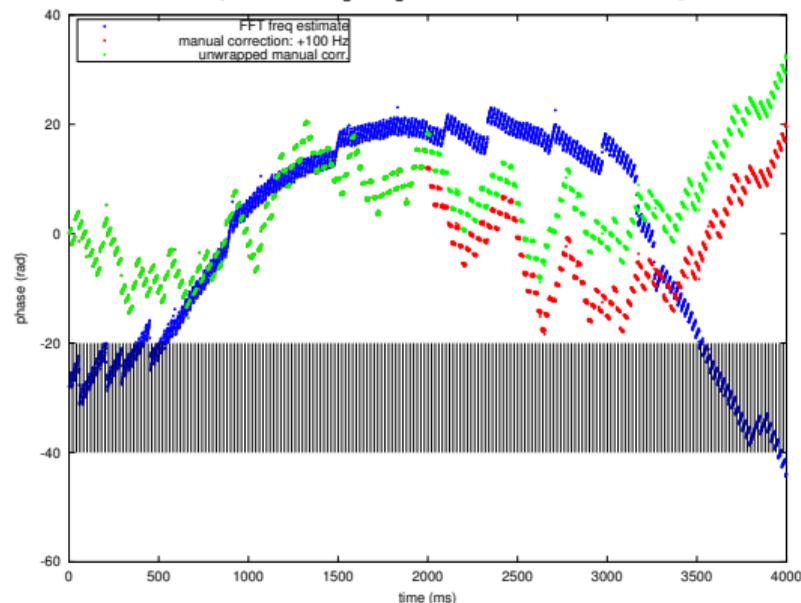
R820T DVB-T

-68 ppm bias

= -107 kHz at 1575.42 GHz

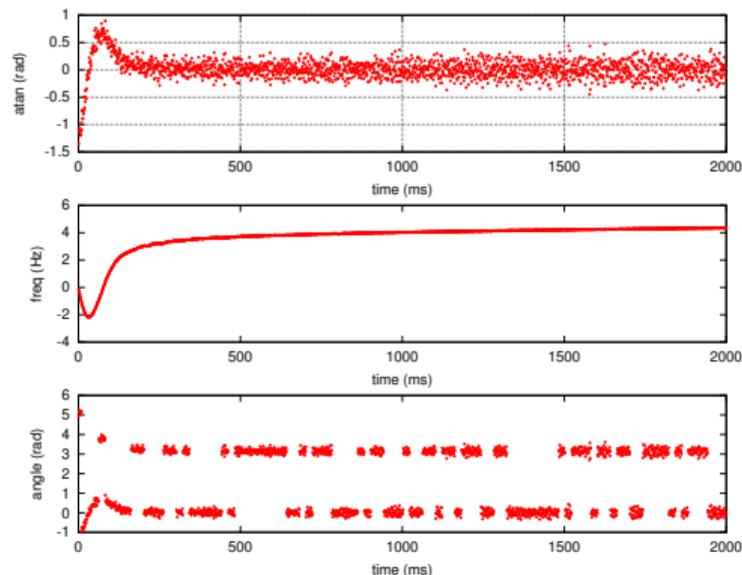
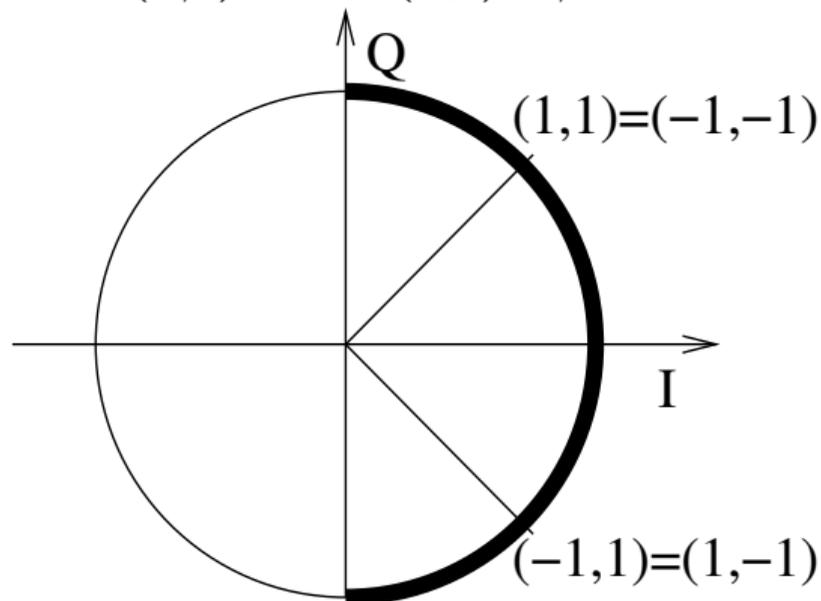
CDMA: decoding GPS

- ▶ Cross-correlating the received RF signal with orthogonal codes allows for identifying the source of the signal, but the message is lost
- ▶ once the **acquisition** phase is completed, **tracking** by controlling LO on the received carrier
- ▶ challenge: the phase is used both to encode the message and track the carrier
- ▶ how to eliminate the phase modulation to control the frequency ?
- ▶ N-PSK : $\varphi^N = 0[2\pi]$ but reduction by a factor N of the allowed frequency offset



CDMA: decoding GPS

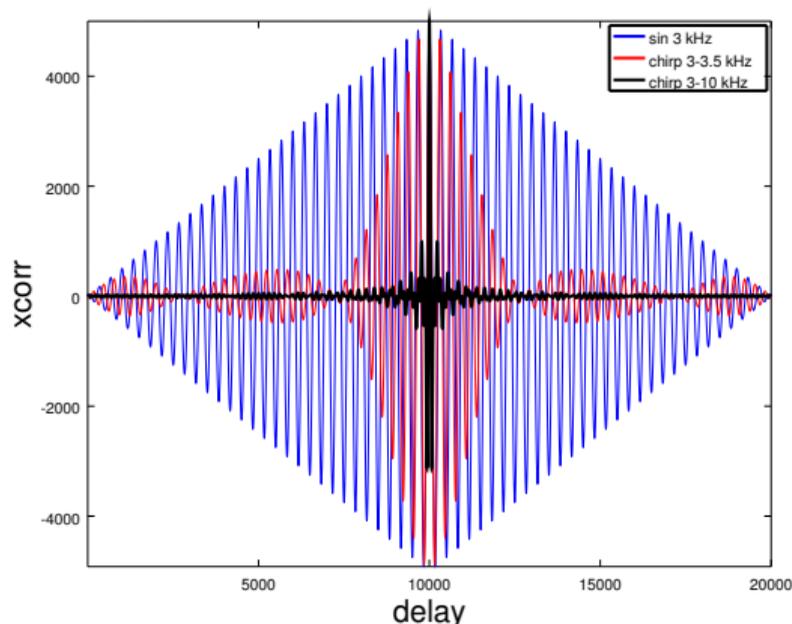
- ▶ Cross-correlating the received RF signal with orthogonal codes allows for identifying the source of the signal, but the message is lost
- ▶ once the **acquisition** phase is completed, **tracking** by controlling LO on the received carrier
- ▶ challenge: the phase is used both to encode the message and track the carrier
- ▶ how to eliminate the phase modulation to control the frequency ?
- ▶ $\text{atan}(Q/I)$ v.s $\text{atan2}(Q, I)$: Q/I cannot detect 180° phase rotation, while atan2 provides NAV..



Pulse compression basics

- ▶ The longer the code (T), the longer the time during which the integral of `xcorr` accumulates energy and **smoothes noise**,
- ▶ but long pulse induces **loss of time resolution** \Rightarrow cross-correlation is a broad peak
- ▶ strong variation of code over time \Rightarrow increased bandwidth $B \Rightarrow$ cross correlation peak width $1/B$

$$\text{pulse compression ratio (PCR)} = B \cdot T$$



```
time=[0:1e-6:1e-2]; %samp. rate=1 us
```

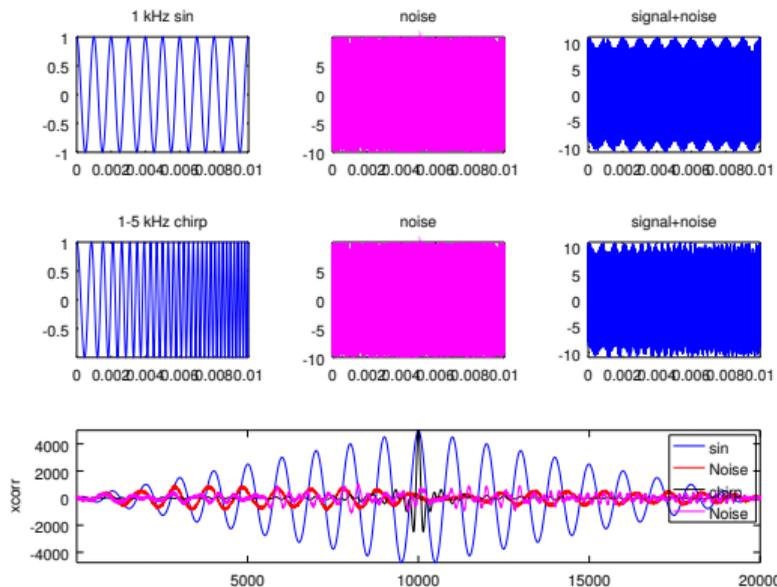
```
x=chirp(time,1e3,time(end),1e3);  
noise=20*rand(length(x),1)';  
noise=noise-mean(noise);  
xx=xcorr(x,x); xb=xcorr(x,noise);  
plot(xx,'b-');hold on;plot(xb,'r-');
```

```
x=chirp(time,1e3,time(end),5e3);  
xx=xcorr(x,x); xb=xcorr(x,noise);  
plot(xx,'k-');hold on;plot(xb,'m-');
```

Pulse compression basics

- ▶ The longer the code (T), the longer the time during which the integral of $xcorr$ accumulates energy and **smoothes noise**,
- ▶ but long pulse induces **loss of time resolution** \Rightarrow cross-correlation is a broad peak
- ▶ strong variation of code over time \Rightarrow increased bandwidth $B \Rightarrow$ cross correlation peak width $1/B$

$$\text{pulse compression ratio (PCR)} = B \cdot T$$



```
time=[0:1e-6:1e-2]; %samp. rate=1 us
```

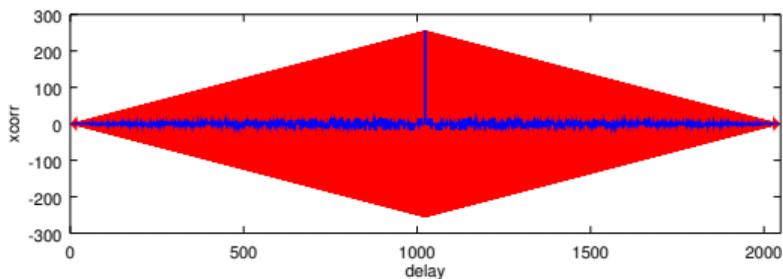
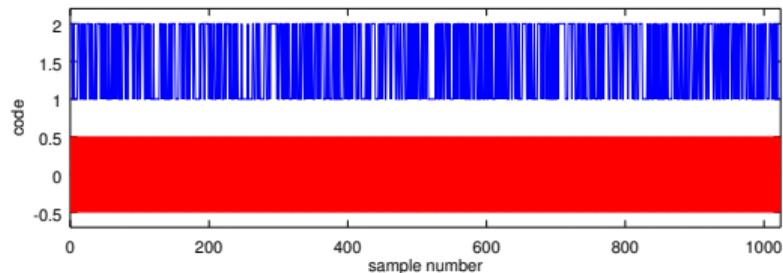
```
x=chirp(time,1e3,time(end),1e3);  
noise=20*rand(length(x),1)';  
noise=noise-mean(noise);  
xx=xcorr(x,x); xb=xcorr(x,noise);  
plot(xx,'b-');hold on;plot(xb,'r-');
```

```
x=chirp(time,1e3,time(end),5e3);  
xx=xcorr(x,x); xb=xcorr(x,noise);  
plot(xx,'k-');hold on;plot(xb,'m-');
```

Pulse compression basics

- ▶ The longer the code (T), the longer the time during which the integral of $xcorr$ accumulates energy and **smoothes noise**,
- ▶ but long pulse induces **loss of time resolution** \Rightarrow cross-correlation is a broad peak
- ▶ strong variation of code over time \Rightarrow increased bandwidth $B \Rightarrow$ cross correlation peak width $1/B$

$$\text{pulse compression ratio (PCR)} = B \cdot T$$



Remember: GPS is designed for **timing signals** with better than one “chip” resolution.

```
noise=rand(1023,1) '*7;  
noise=noise-mean(noise);  
b=[1:1023];  
b=mod(b,2);b=b-mean(b);  
plot(xcorr(b+noise,b),'r');hold on  
  
a=cacode(1,1);a=a-mean(a);  
plot(xcorr(a+noise,a));  
plot(a+1.5);hold on;plot(b,'r');
```

Amplifier

- ▶ gain assumed to be constant for a given (low) input power range
- ▶ finite output power: gain compression at high input power

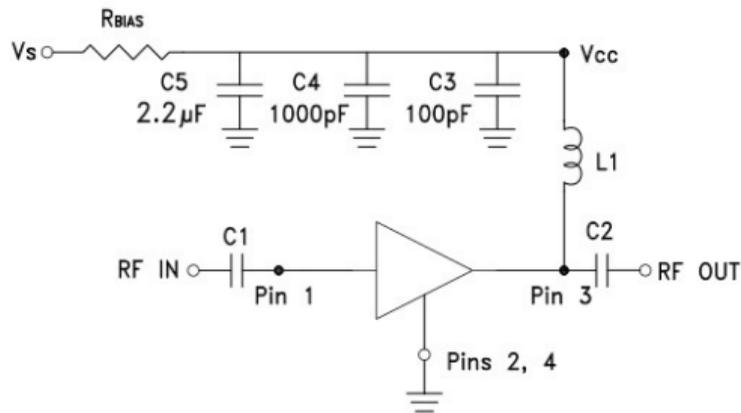
▶ $F = SNR_{in}/SNR_{out}$ (>1): noise factor (given in datasheets)

▶ noise spectral density proportional to equivalent temperature and input power: $P_N = k_B \cdot T \cdot F$ where $k_B = 1.38 \cdot 10^{-23} \text{ J.K}^{-1}$

Express P_N in dBm/Hz^a, i.e. normalized to 1 mW, if $F = 1$.

▶ smallest detectable input signal due to thermal noise (excluding correlation techniques): $P_N \times BW$ ($P_{NO} = G \times P_N \times BW$) thermal noise floor

Application Circuit



Recommended Bias Resistor Values for $I_{cc} = 35 \text{ mA}$, $R_{bias} = (V_s - V_{cc}) / I_{cc}$

Supply Voltage (V_s)	5V	8V	10V	12V
RBIAS VALUE	56 Ω	130 Ω	180 Ω	240 Ω
RBIAS POWER RATING	1/8 W	1/4 W	1/4 W	1/2 W

Note:

1. External blocking capacitors are required on RFIN and RFOUT.
2. R_{BIA}S provides DC bias stability over temperature.

^aSince $J = W \cdot s$, then $k_B \times T$ is J=W/Hz

Link budget

- ▶ a radiofrequency (electrical) power is emitted, either isotropically or in a directional pattern with an antenna gain G_1 : $P_E \times G_1$
- ▶ this power spreads on a sphere centered on the emitter: in the case of isotropic emitter, the area of this sphere is, at a distance d , $4\pi d^2$
- ▶ if $G_1 > 1$, then only a fraction $4\pi d^2/G_1$ of the sphere is illuminated
- ▶ this sphere intersects the receiver, which can detect any incoming signal on a 4π -steradian sphere on a typical area of λ^2
- ▶ this receiver might exhibit a reception antenna gain G_2 normalized over 4π steradians

$$\frac{P_R}{P_E} = G_1 G_2 \left(\frac{\lambda}{4\pi d} \right)^2 : \text{Friis } ^a \text{ equation}$$

leading to Free Space Propagation Loss (FSPL):

$$FSPL = 20 \log_{10}(f) + 20 \log_{10}(d) - 147.55 \text{ dB}$$

since $20 \log_{10}(c/4/\pi) = 147.5 \text{ dB}$

^aH.T.Friis *A Note on a Simple Transmission Formula*,
Proc.I.R.E. 254--256 (1946) →

DERIVATION OF TRANSMISSION FORMULA (1)

Having defined the effective area of an antenna, it is a simple matter to derive (1). As shown in Fig. 1, consider a radio circuit made up of an isotropic transmitting

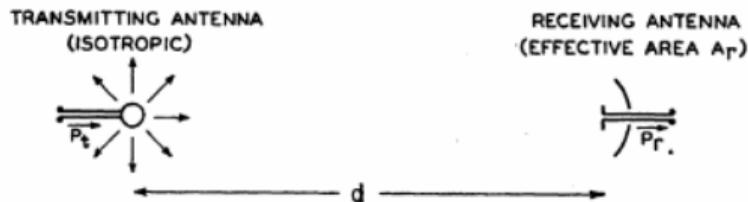


Fig. 1—Free-space radio circuit.

Link budget

- ▶ a radiofrequency (electrical) power is emitted, either isotropically or in a directional pattern with an antenna gain G_1 : $P_E \times G_1$
- ▶ this power spreads on a sphere centered on the emitter: in the case of isotropic emitter, the area of this sphere is, at a distance d , $4\pi d^2$
- ▶ if $G_1 > 1$, then only a fraction $4\pi d^2 / G_1$ of the sphere is illuminated
- ▶ this sphere intersects the receiver, which can detect any incoming signal on a 4π -steradian sphere on a typical area of λ^2
- ▶ this receiver might exhibit a reception antenna gain G_2 normalized over 4π steradians

Application:

1. a GPS satellite emits 50 W (17 dBW=47 dBm) at 1575.42 MHz with an antenna gain of 13 dBi and flies at 20000 km over the Earth
2. $FSPL = 182 \text{ dB} \Rightarrow P_R = -152 \text{ dBW} = -122 \text{ dBm}$
3. receiver sensitivity: typically around -159 dBm
(usglobalsat.com/store/download/53/et312_ug.pdf)
4. DVB-T: detection limit around -95 dBm (10 dB SNR) + 27 dB antenna gain = **-122 dBm**
detection limit

Link budget

- ▶ a radiofrequency (electrical) power is emitted, either isotropically or in a directional pattern with an antenna gain G_1 : $P_E \times G_1$
- ▶ this power spreads on a sphere centered on the emitter: in the case of isotropic emitter, the area of this sphere is, at a distance d , $4\pi d^2$
- ▶ if $G_1 > 1$, then only a fraction $4\pi d^2/G_1$ of the sphere is illuminated
- ▶ this sphere intersects the receiver, which can detect any incoming signal on a 4π -steradian sphere on a typical area of λ^2
- ▶ this receiver might exhibit a reception antenna gain G_2 normalized over 4π steradians

What is the thermal noise power ?

1. 2 MHz bandwidth (1023 kHz C) so that $10 \log_{10}(10^6) = 63$ dB
2. $-174 + 63 = -111$ dBm > -122 dBm !
3. but 30 dB=1023 kHz/1 kHz pulse compression: $-122 + 30 = -92 > 111$ dBm ($SNR \simeq 22$ dB after compression)
4. the cross-correlation brings the signal out of the noise: a spectral analysis (FFT) **cannot display** the GPS signal !

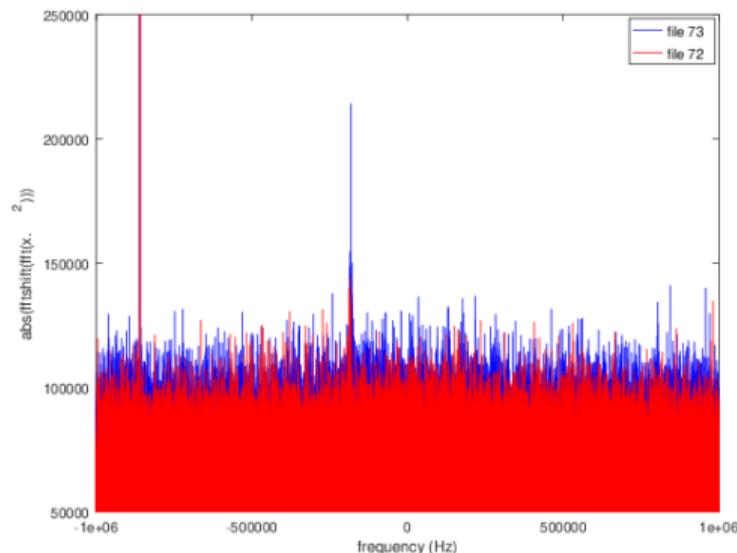
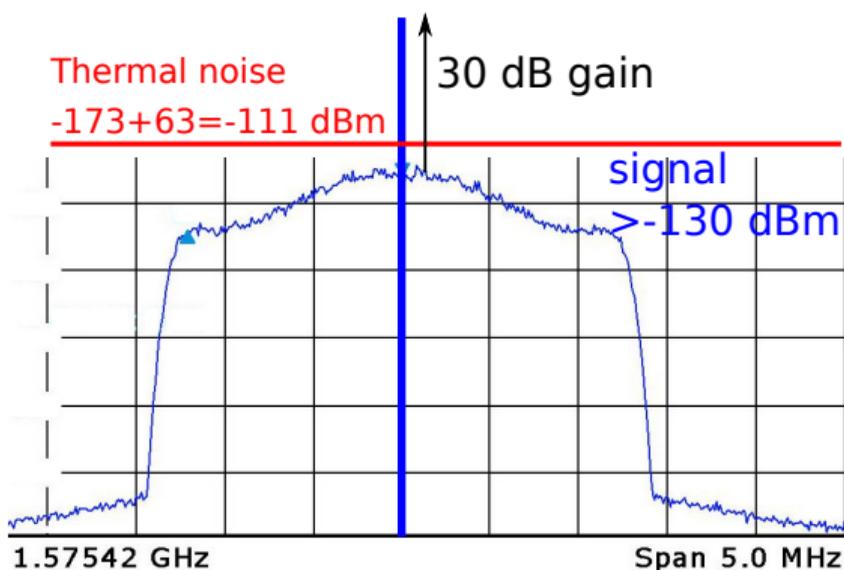
Back to GPS (BPSK)

Squaring a BPSK signal gets rid of modulation and collects all the energy in the carrier

but doubles the noise (+3 dB).

$$\text{Noise floor: } \underbrace{k_B = 1.38 \cdot 10^{-23} \text{ J/K}}_{\text{Boltzman constant}} \underbrace{\times T}_{=293 \text{ K}} = -204 \text{ W/Hz} = -174 \text{ dBm/Hz}$$

$$\times 2 \text{ MHz or } +63 \text{ dB} = -111 \text{ dBm}$$



Coarse estimate of (twice) the Doppler shift+frequency offset²⁰

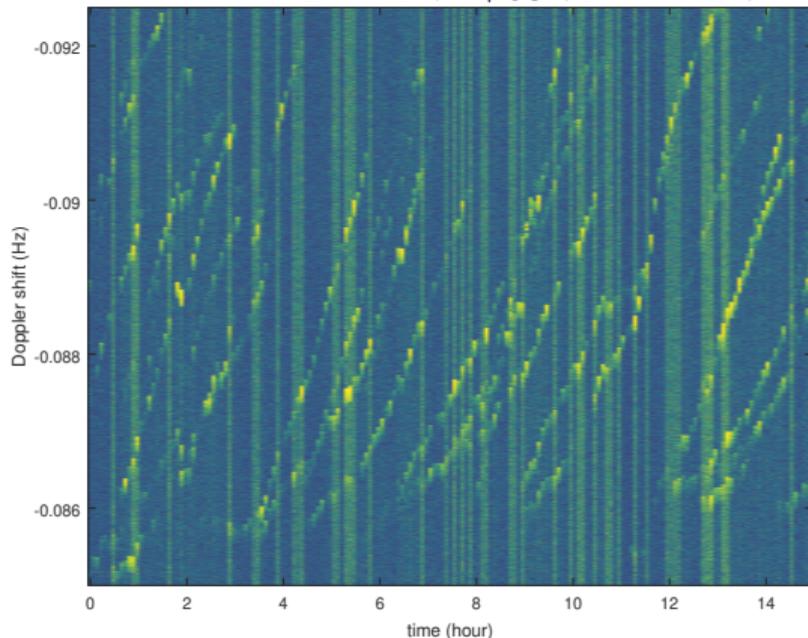
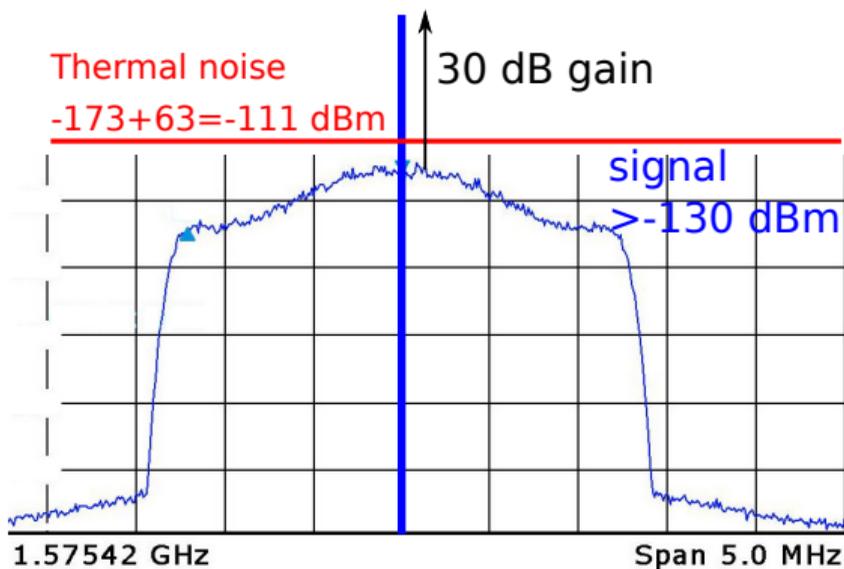
²⁰P. Boven, *Observe, Hack, Make: GPS* (2013): used in Vaisala RS80 radiosonde

Back to GPS (BPSK)

Squaring a BPSK signal gets rid of modulation and collects all the energy in the carrier
but doubles the noise (+3 dB).

Noise floor: $k_B = 1.38 \cdot 10^{-23} \text{ J/K} \times T = -204 \text{ W/Hz} = -174 \text{ dBm/Hz}$
 Boltzman constant = 293 K

$\times 2 \text{ MHz or } +63 \text{ dB} = -111 \text{ dBm}$



Coarse estimate of (twice) the Doppler shift+frequency offset²⁰

²⁰P. Boyen, *Observe Hack Make: GPS* (2013); used in Vaisala RS80 radiosonde

Link budget (ground)

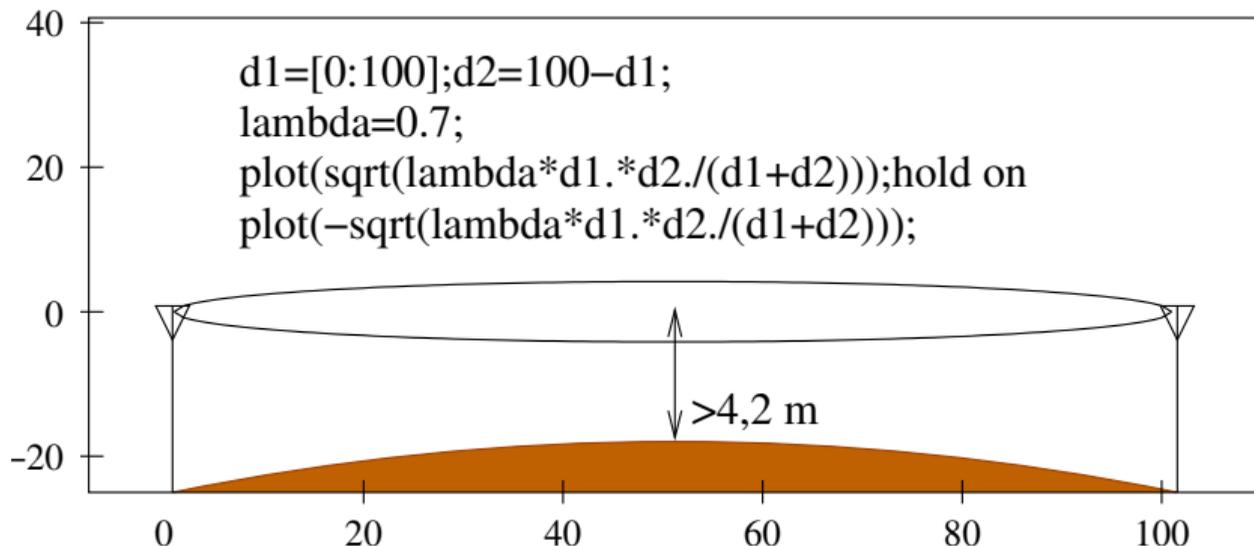
- ▶ reflections/multiple paths (destructive interferences, fading)

- ▶ Fresnel zone (volume in which reflections interfere)

$$r = \sqrt{\frac{n\lambda \cdot d_1 \cdot d_2}{d_1 + d_2}}, \quad n \in \mathbb{N} \quad (n = 2p \text{ additive interferences, } n = 2p + 1 \text{ destructive interferences})$$

- ▶ $r < \sqrt{\lambda D}/2$ if $D = d_1 + d_2$ (cf near field v.s. far field)

- ▶ absorption by the environment (water ²¹, ground soil)



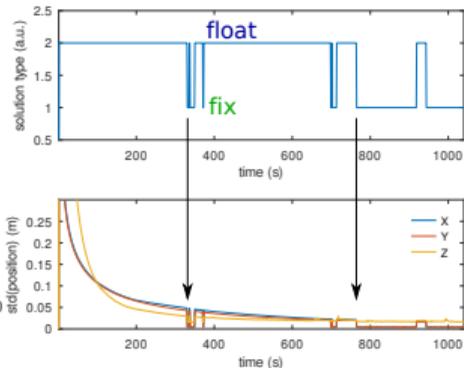
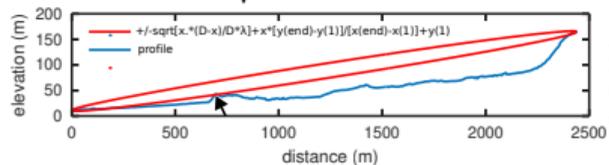
Example of Fresnel zone: talking to a glacier!

868-MHz LoRa ("Long Range") protocol to communicate from a basestation to sensors in the moraine of an Arctic glacier

Range: $D = 2.4$ km (moraine)

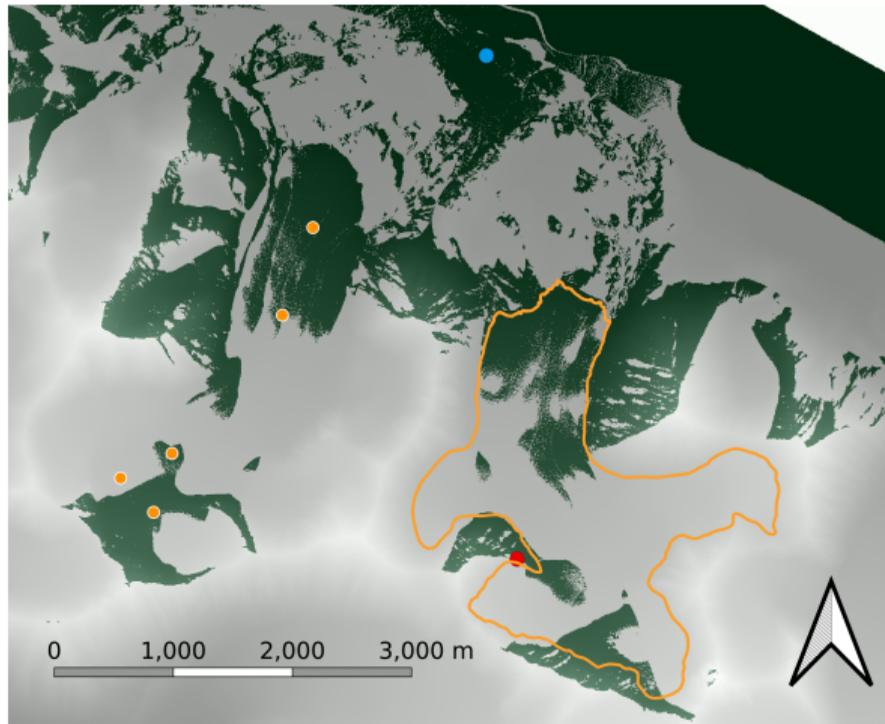
Fresnel zone clearance:

$$R > \frac{1}{2} \sqrt{\frac{cD}{f}} \approx 15 \text{ m}$$

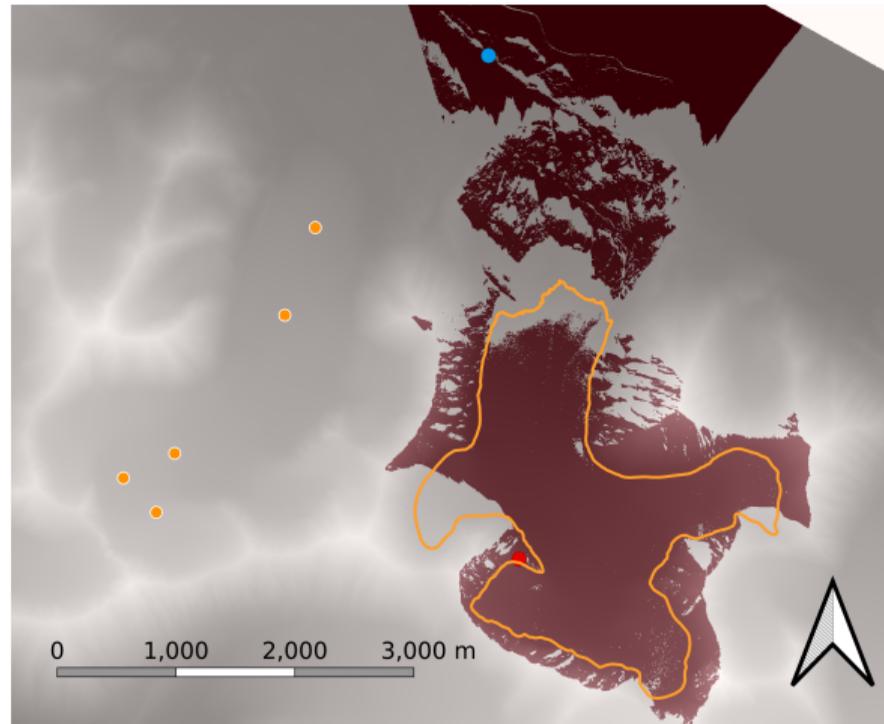


Radiofrequency communication range (QGis: viewshed analysis)

Using Digital Elevation Model of the area being surveyed to predict communication range



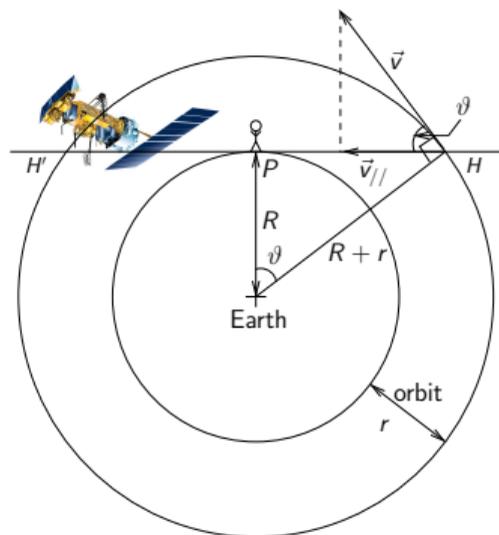
From Corbel station (blue dot)
including Midtre Lovén glacier (S. Filhol, Oslo Univ.)



From the ridge (red dot)
blue+red = full coverage

Communication range

Earth curvature and distance to the horizon limits the communication range



Distance d at which an emitter at altitude r can communicate with a receiver:

$$\cos(\vartheta) = \frac{R}{r+R} \text{ and } d = R \times \vartheta$$

$$\Rightarrow d = R \arccos\left(\frac{R}{r+R}\right)$$

where $R \simeq 6400$ km

(LEO: $r = 800$ km $\Rightarrow d \simeq 3000$ km)

(ISS: $r \simeq 400$ km $\Rightarrow d \simeq 2200$ km)

(Puy de Dome: $r \simeq 1.5$ km $\Rightarrow d \simeq 140$ km)

\Rightarrow raise emitter/relay altitude, all the way to space (geostationary satellite at 36000 km: $d = 9100$ km but only up to $\pm 70^\circ$ latitude)

Communication range

Earth curvature and distance to the horizon limits the communication range



Distance d at which an emitter at altitude r can communicate with a receiver:

$$\cos(\vartheta) = \frac{R}{r+R} \text{ and } d = R \times \vartheta$$

$$\Rightarrow d = R \arccos\left(\frac{R}{r+R}\right)$$

where $R \simeq 6400$ km

(LEO: $r = 800$ km $\Rightarrow d \simeq 3000$ km)

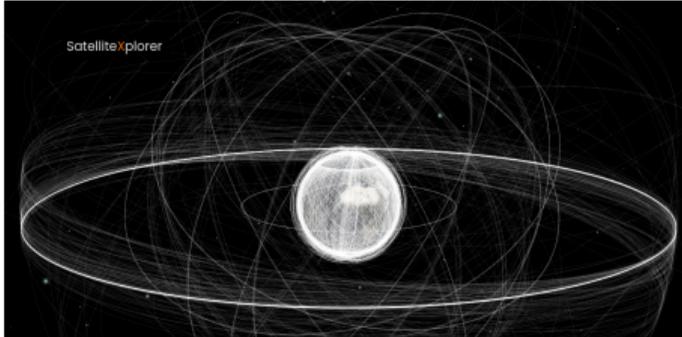
(ISS: $r \simeq 400$ km $\Rightarrow d \simeq 2200$ km)

(Puy de Dome: $r \simeq 1.5$ km $\Rightarrow d \simeq 140$ km)

\Rightarrow raise emitter/relay altitude, all the way to space (geostationary satellite at 36000 km: $d = 9100$ km but only up to $\pm 70^\circ$ latitude)

Communication range: Low Earth Orbiting satellites

LEO: polar orbit benefits from the rotation of the Earth to cover different areas, with a period of ≈ 100 min (Earth monitoring satellites, communication satellite e.g Starlink ...) \Rightarrow communication stations close to the poles



Tromsø, Norway (70°N)



<https://geoxc-apps.bd.esri.com/space/satellite-explorer/>

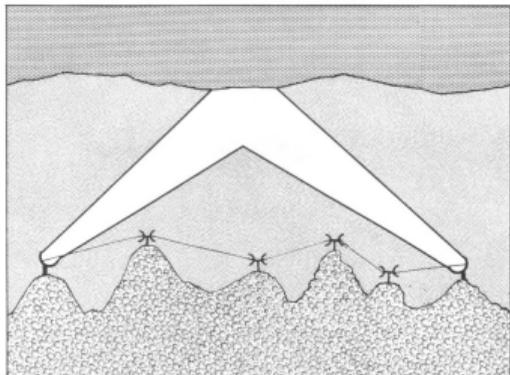
also in Loneyarbyen (78°N)
B. Wormdal, *The Satellite War* (2011)

Beyond the horizon: tropospheric scatter

Eiffel Tower: 324 m-high emitter

⇒ 66 km communication range.

How to link two transceivers at distances > 100 km ?



Troposphere: VHF-SHF (\gg VLF interacting with ionosphere)

Effective Earth radius increases due to refraction of electromagnetic beam.

T.M. Rienzi, *Vietnam Studies – Communications Electronics 1962–1970*,
Department of the Army, 1972



Beyond the horizon: tropospheric scatter

Eiffel Tower: 324 m-high emitter
⇒ 66 km communication range.

How to link two transceivers at distances > 100 km ?



Standard atmosphere (lower N_s in cold regions):

$N_s \approx 300 \Rightarrow R \approx 8500$ km $\Rightarrow d \approx 88$ km

$N_s \approx 400 \Rightarrow R \approx 11000$ km $\Rightarrow d \approx 110$ km

Used during the Cold War for long range communication (Alaska, Siberia),
made obsolete by satellite communications

en.wikipedia.org/wiki/White_Alice_Communications_System
archive.org/details/land-of-white-alice-1960
trrlsever.org/SEVER/SV-NOS/sv-nos.html

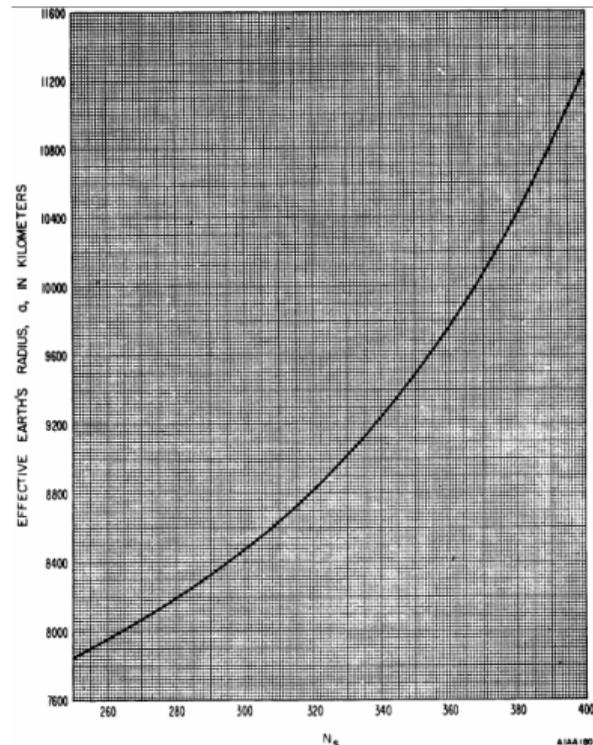


Figure 6-12. Effective Earth's Radius, a , Versus Surface Refractivity, N_s

Navalex 0101,112, *Naval Shore Electronics Criteria – Line of Sight Microwave and Tropospheric Scatter Communication Systems*, Department of the Navy (May 1972)

Antenna

- ▶ radiating element replacing the cable between the emitter and the receiver
- ▶ characteristic dimensions: $\lambda/2$ or, if a conducting plane is present, $\lambda/4$ by using mirror charges,
- ▶ characteristics: impedance, directivity, polarization (NEC2, www.nec2.org)



Antenna

- ▶ radiating element replacing the cable between the emitter and the receiver
- ▶ characteristic dimensions: $\lambda/2$ or, if a conducting plane is present, $\lambda/4$ by using mirror charges,
- ▶ characteristics: impedance, directivity, polarization (NEC2, www.nec2.org)

- ▶ Linear polarization (horizontal, e.g. TV)
- ▶ Linear polarization (vertical, e.g. FM)
- ▶ Circular polarisation (moving target, e.g. satellite)
- ▶ Circular polarisation (RHCP v.s LHCP)
- ▶ linear polarization error
- ▶ Gain G determined by effective area

$$A_{eff} = G \frac{\lambda^2}{4\pi}$$



Antenna

- ▶ radiating element replacing the cable between the emitter and the receiver
- ▶ characteristic dimensions: $\lambda/2$ or, if a conducting plane is present, $\lambda/4$ by using mirror charges,
- ▶ characteristics: impedance, directivity, polarization (NEC2, www.nec2.org)

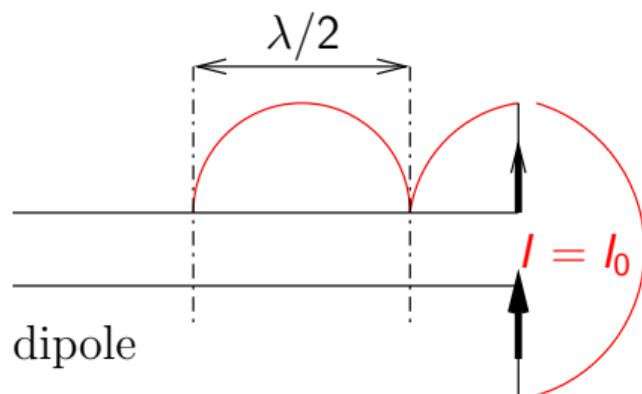
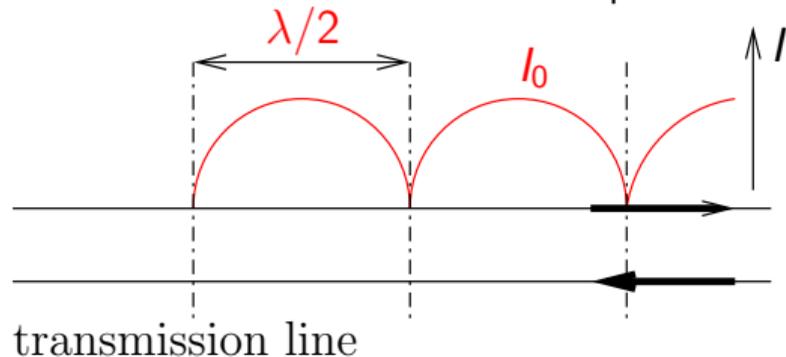
- ▶ Linear polarization (horizontal, e.g. TV)
- ▶ Linear polarization (vertical, e.g. FM)
- ▶ Circular polarisation (moving target, e.g. satellite)
- ▶ Circular polarisation (RHCP v.s LHCP)
- ▶ linear polarization error
- ▶ Gain G determined by effective area

$$A_{eff} = G \frac{\lambda^2}{4\pi}$$



Dipole antenna basics

From the transmission line to the dipole radiating element ²²



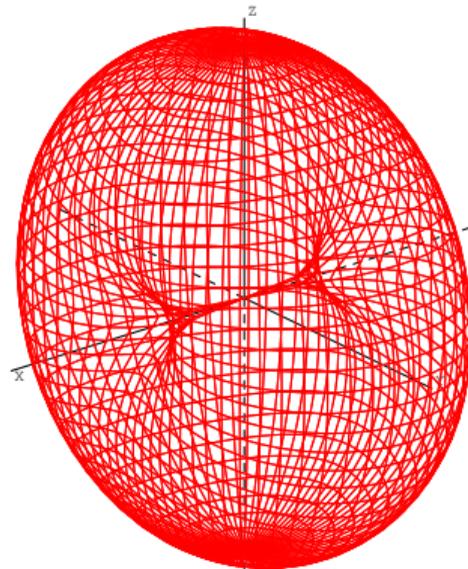
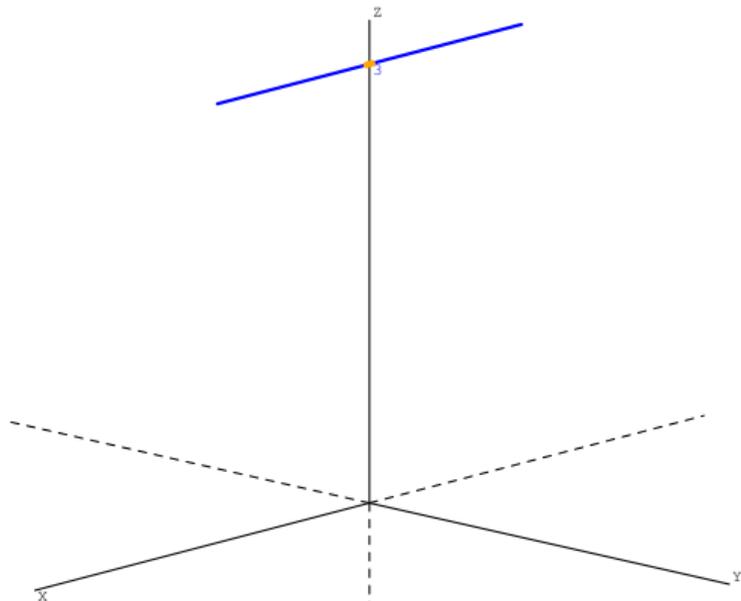
- ▶ Field lines cancel along the transmission line
- ▶ Field lines add in the radiating element
- ▶ Maximum current at the feed point and nul at the end of the radiating element
- ▶ Wire length $\lambda/4$ for a dipole of length: $\lambda/2$)
- ▶ Linear polarization
- ▶ $Z = 73 + j \cdot 42,5 \Omega$ – shorten dipole to cancel imaginary part

²²C.A. Balanis, *Antenna Theory, Analysis and Design, 3rd Ed.*, Wiley Interscience (2005)

Dipole basics: radiation diagram... ²³

of a dipole simulated using NEC2 (nec2c package for Debian/GNU Linux), viewed using xnecview

- ▶ Free space: null along the dipole axis

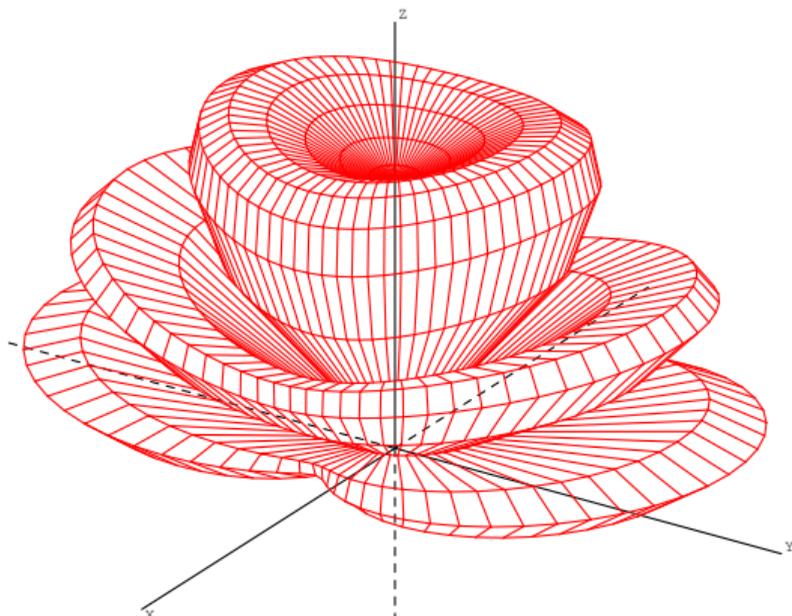
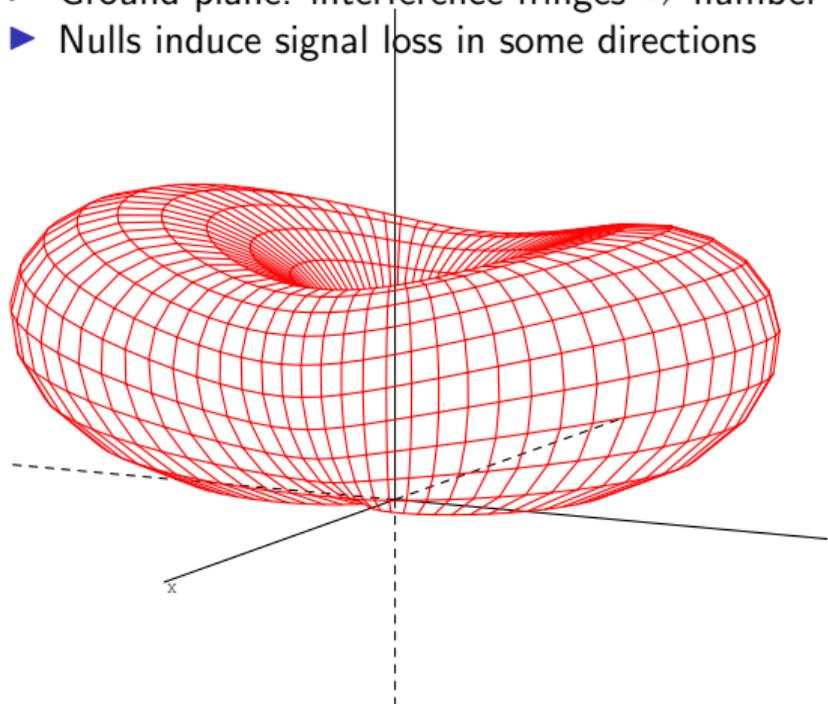


²³J.-L. Smith, *Basic NEC with broadcast applications*, Focal Press (2008)

Dipole basics: radiation diagram... ²³

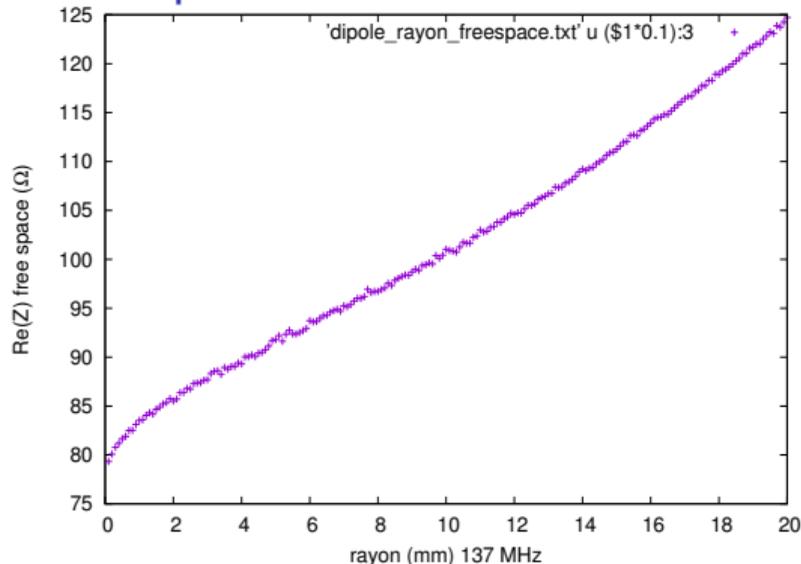
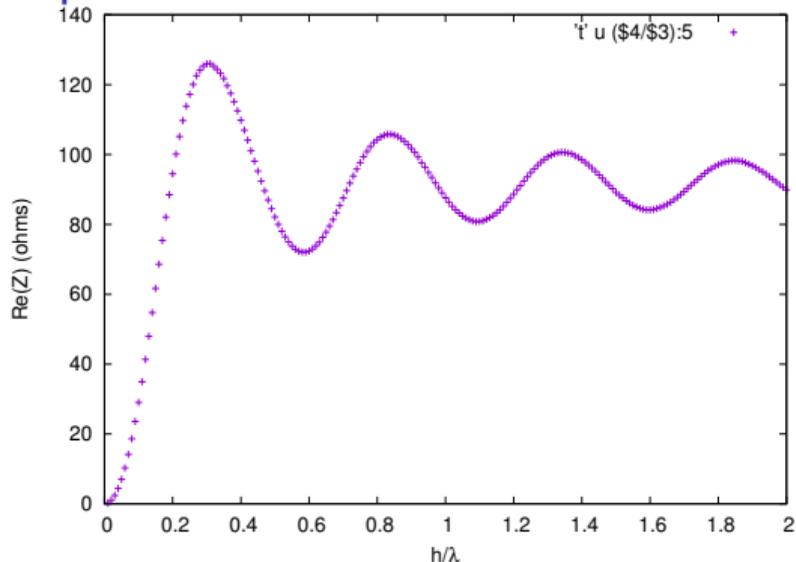
of a dipole simulated using NEC2 (nec2c package for Debian/GNU Linux), viewed using xnecview

- ▶ Ground plane: interference fringes \Rightarrow number of petals = $2 \cdot \text{height} / \lambda + 1$
- ▶ Nulls induce signal loss in some directions



²³ J.-M Friedt, É. Carry, O. Testault, *Petites antennes réalisées par impression additive : de la conception à la visualisation des diagrammes de rayonnement (en vrai et en virtuel)*, Hackable **31**, pp.80-96 (Oct-Nov. 2019) on 3D printing radiation patterns and visualising in augmented reality.

Dipole basics: from the ideal to the real dipole



► Z depends on the radius of the radiating wire and its distance to ground

► Simulation example using NEC2:

CM dipole : GN=1 pour gnd plane, -1 pour free space

CE

GW 3 51 -5.4745E-01 0.00E-01 0.1E01 5.47450E-01 0.00E-01 0.1E01 0.1E-2

GE 1

GN 1

EX 0 3 26 0 1.00000E+00 0.00000E+00

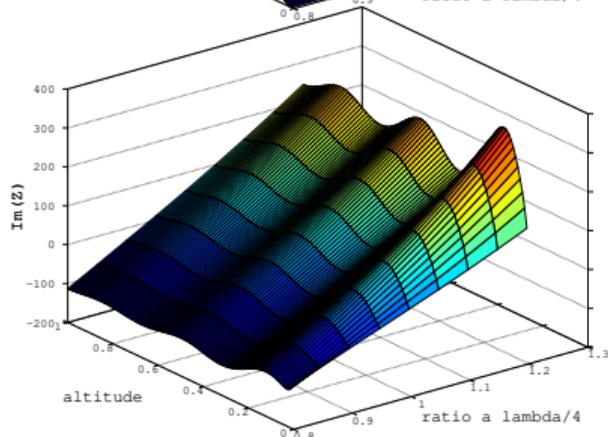
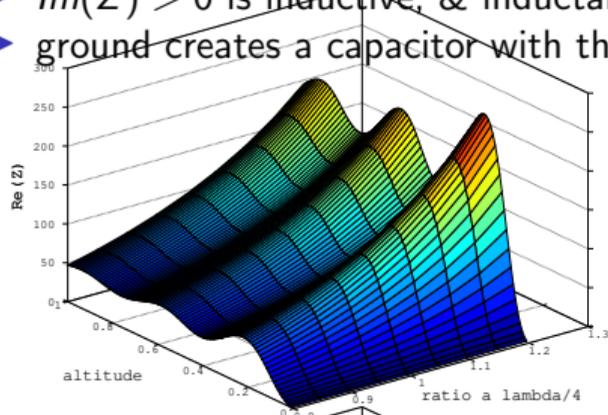
FR 0 1 0 0 1.37000E+02 1.00000E-00

RP 0 72 72 0 0.00000E+00 0.00000E+00 5.00000E+00 5.00000E+00

EN

From the ideal dipole to the real dipole

- ▶ impedance matching with a transmission line of characteristic real impedance $Z_0 \simeq 50 \Omega$
- ▶ $Im(Z) > 0$ is inductive, & inductance \propto length \Rightarrow shorten the dipole
- ▶ ground creates a capacitor with the dipole wires



```
m=1;
L=0.175
for long=0.8:0.05:1.2
  n=1;
  for alti=1e-2:0.01:1
    f=fopen('fichier.nec','w');
    fprintf(f,"CM dipole\n");
    fprintf(f,"CE\n");
    fprintf(f,"GW 3 51 -%f 0.00E-01 %f %f 0.00E-01 %f 1.00E-03\n",L*long, →
            ↪ alti,L*long,alti)
    fprintf(f,"GE 1\n");
    fprintf(f,"GN 1\n");
    fprintf(f,"EX 0 3 26 0 1.0 0.0\n");
    fprintf(f,"FR 0 1 0 0 4.34E+02 3.0\n");
    fprintf(f,"RP 0 144 144 0 0.0 0.0 2.5 2.5\n");
    fprintf(f,"EN\n");
    fclose(f);
    system('nec2c -i fichier.nec -o sort.out');
    system('grep -A2 OHMS sort.out | tail -1 > res');
    load res
    impedance_reel(n,m)=res(7);
    impedance_imag(n,m)=res(8);
  n=n+1
end
m=m+1
end
```

GNU/Octave script for generating dynamically the NEC2 configuration file and fetching result

Standing Wave Ratio calculation ²⁴

- ▶ *SWR* represents impedance matching of transmission line with load: $SWR = 1$ means there is no node/antinode, i.e. impedance matching and no standing wave pattern

- ▶ xneview source code:

```
#define R0 50.0 /* default reference impedance for SWR calculation */
double r0=R0; /* reference impedance for SWR calculation */

void calcswr(NECOutput *ne)
{
    double zr,zi,gamma;
    zr=ne->d[neco_zr];
    zi=ne->d[neco_zi];
    gamma = sqrt( ( (zr-r0)*(zr-r0) + zi*zi ) / ( (zr+r0)*(zr+r0) + zi*zi ) );
    ne->d[neco_swr] = (1+gamma)/(1-gamma);
}
```

- ▶ $\Gamma = \frac{V_r}{V_f} = \frac{Z_L - Z_0}{Z_L + Z_0} \in \mathbb{C}$ (f : forward wave, r : reflected wave)

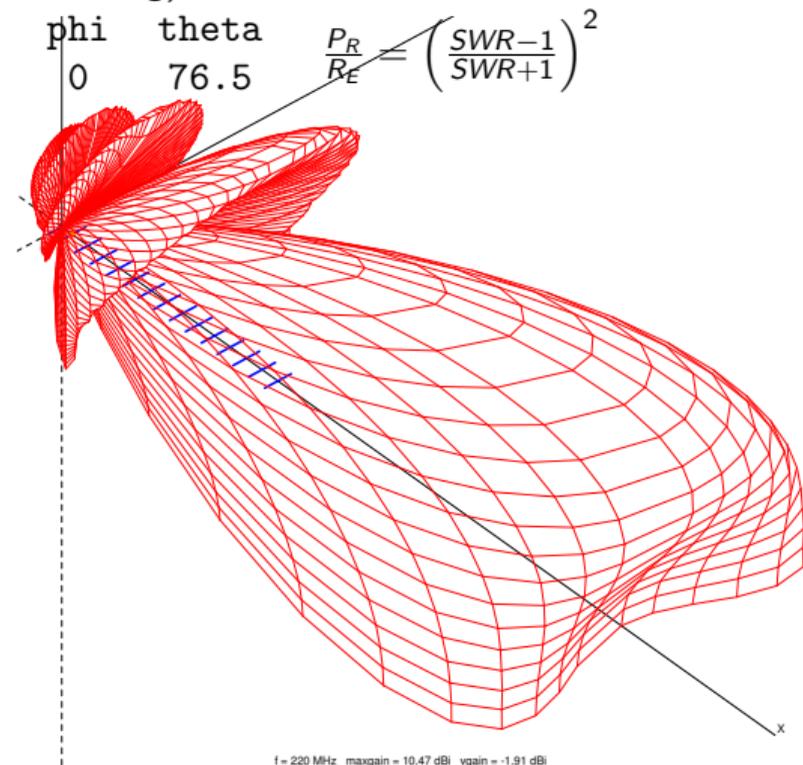
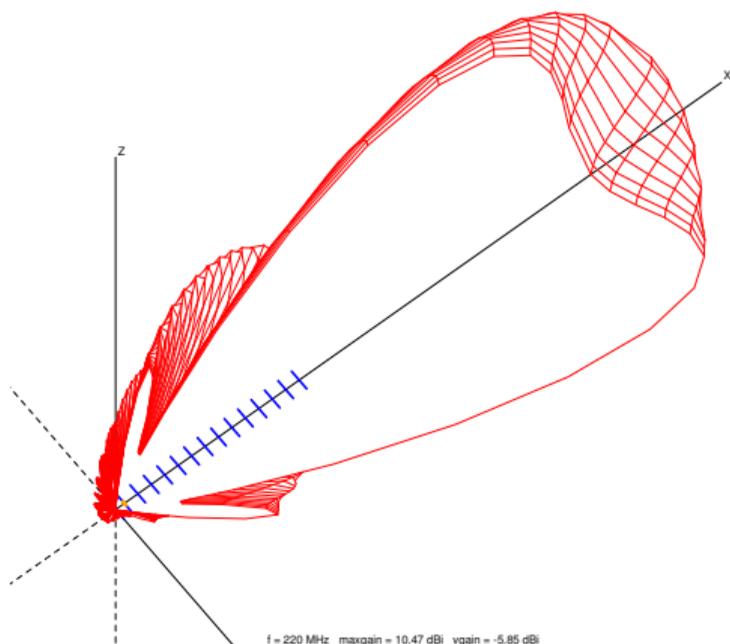
- ▶ $SWR = \frac{|V_{max}|}{|V_{min}|} = \frac{1+|\Gamma|}{1-|\Gamma|}$

²⁴https://en.wikipedia.org/wiki/Standing_wave_ratio

Yagi-Uda directional antenna

- ▶ a radiating element replaces the cable between the emitter and the receiver
- ▶ characteristic dimensions: $\lambda/2$ or, over a conducting plane, $\lambda/4$ by using mirror charges,
- ▶ impedance, directivity, polarization (NEC2, www.nec2.org)

#	freq.	Zr	Zi	SWR	gain	f/b	phi	theta	$\frac{P_R}{P_E} = \left(\frac{SWR-1}{SWR+1}\right)^2$
	220	26.1	2.2	1.9	10.47	-	0	76.5	



Optimizing antenna geometry

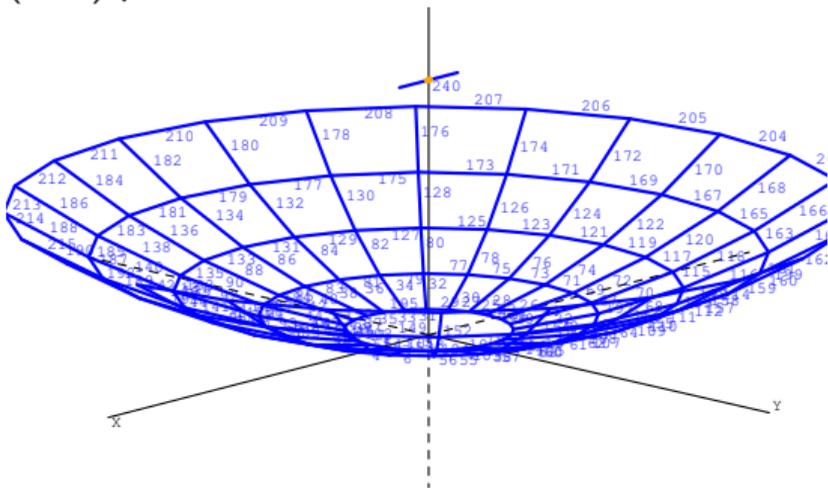
Let us assume that we are not aware of parabola characteristics ... how can we optimize the position of the feed ? ²⁵

```
function maxgain=parabole_optim(ydipole)
freq=1500; % frequency in MHz
lambda=300/freq;
p=0.3; % focal point position 1/(4*p)
dtheta=10;
dr=0.1;
n=13; % points/segment
rad=lambda/100;
r=[dr:dr:1-dr]; theta=[0:dtheta:360-dtheta]*pi/180;
x1=r*cos(theta); y1=r*sin(theta); r1=abs(x1+i*y1); z1=p*r1.^2;
theta=[dtheta:dtheta:360]*pi/180;
x2=r*cos(theta); y2=r*sin(theta); r2=abs(x2+i*y2); z2=p*r2.^2;
r=[dr+dr:dr:1]; theta=[0:dtheta:360-dtheta]*pi/180;
x3=r*cos(theta); y3=r*sin(theta); r3=abs(x3+i*y3); z3=p*r3.^2;
theta=[dtheta:dtheta:360]*pi/180;
x4=r*cos(theta); y4=r*sin(theta); r4=abs(x4+i*y4); z4=p*r4.^2;
f=fopen('entree.nec','w');
[u,v]=size(x1);
fprintf(f,"CM parabole\n");
fprintf(f,"CE\n");
inc=1;
for k=1:u-1
    for l=1:v
        fprintf(f,"SP 0 3 %f %f %f %f %f %f %f\n",x1(k,l),y1(k,l),z1(k,l),x2(k,l),y2(k,l),z2(k,l));
    ...
system('/usr/bin/nec2c -i input.nec -o output.out && cat output.out | tail -186 | head -180 | cut -c 1-46 > output.octave');
x=load('output.octave');
maxgain=max(x(:,3))
maxgain=1/maxgain;
```

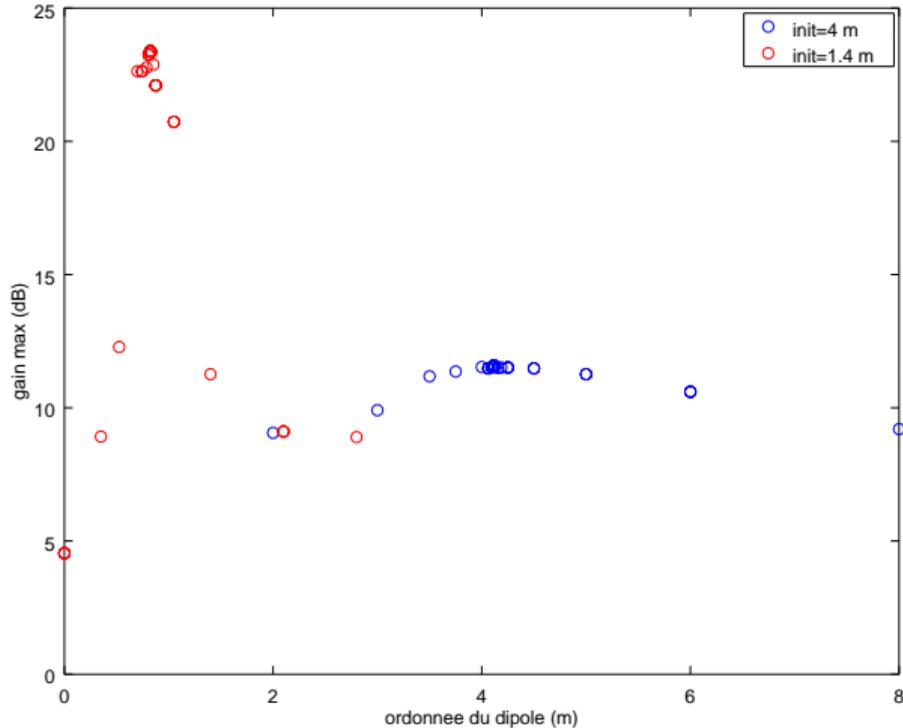
²⁵J.-M Friedt, *Modélisation et utilisation d'une parabole : application au wifi*, Opensilicium 20 pp.44–55 (2016) at http://jmfriedt.free.fr/lm_parabole.pdf

Optimizing antenna geometry

NEC2 for optimizing the radiating element (feed) position ? ^a



^aJ.-M Friedt, *Modélisation et utilisation d'une parabole : application au wifi*, Opensilicium **20** pp.44–55 (2016) at http://jmfriedt.free.fr/lm_parabole.pdf



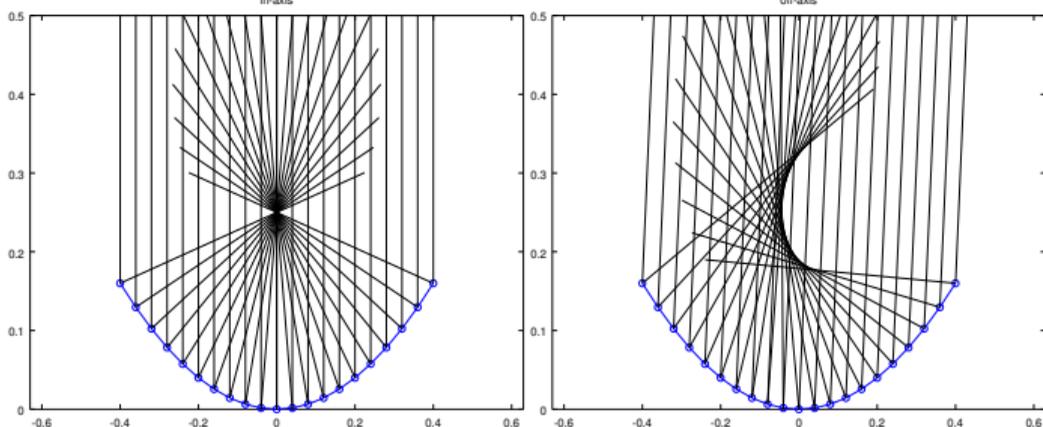
```
freq=1500; % frequency in MHz
lambda=300/freq;
p=0.3; % focal point position 1/(4*p)
dtheta=10;
dr=0.1;
n=13; % points/segment
X0=1.4; % ydipole initialization
Xopt=fminsearch('parabole_optim', X0);
printf("optim=%f theory=%f\n", Xopt, 1/(4*p))
```

Parabola antenna gain

- ▶ Parabola = large reflector focusing all parallel incoming rays to the focal point
- ▶ Dipole antenna located at focal point
- ▶ Replace the λ^2 area of the intersection of the antenna in Friis equation with parabola area
- ▶ area of parabola with diameter D (physical aperture): $A = \pi D^2/4$
- ▶ antenna gain

$$\frac{A}{\lambda^2} \cdot \underbrace{\frac{4\pi}{1}}_{\text{dB}} \sim 10 \log_{10} \left(\frac{\pi^2 D^2}{\lambda^2} \right)$$

- ▶ beamwidth = $\frac{360\lambda}{2\pi D} = \frac{57.3\lambda}{D}$ (theoretical) $\simeq \frac{70\lambda}{D}$ degrees (practical)



- ▶ offset (off-axis) dish antenna (\neq frond-feed): avoid shadowing reflector (Keeping a geostationary satellite within a $30 \text{ km radius} = 0.1^\circ \ll 5^\circ$)

NEC rules

- ▶ segment length no longer than $\lambda/10$
- ▶ radius smaller than $\lambda/100$ and $1/8$ th of the segment length
- ▶ when modelling surfaces, a segment radius selected so that the areas of the segments surrounding a region have the same area.

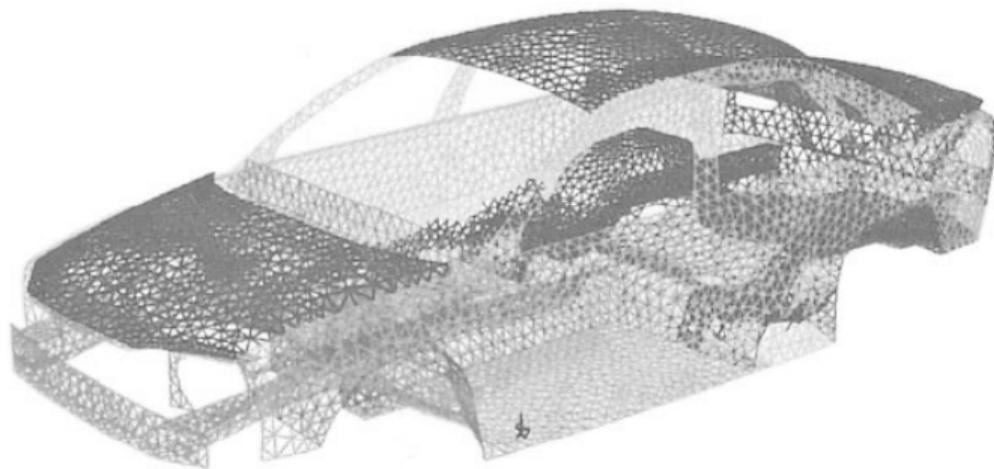


Fig. 6. NEC mesh of the considered geometry consisting of 23 449 segments used for performance comparisons.

Example: A. Rubinstein, F. Rachidi, M. Rubinstein, B. Reusser, *A parallel implementation of NEC for the analysis of large structures*, IEEE Trans. on Electromagnetic Compatibility **45** (2), 177–188 (2003) with a “modern” parallelization implementation at <https://github.com/vineethajoy/Parallel-NEC>

Circular polarization

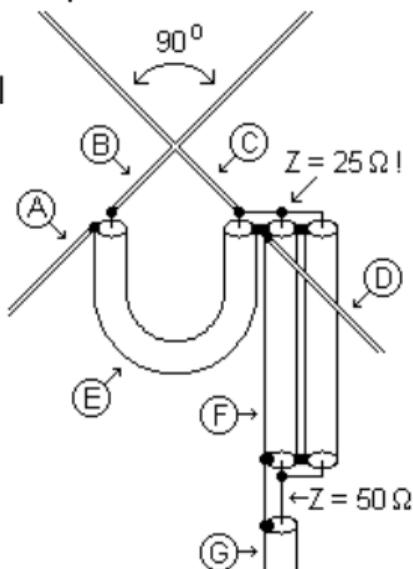
From linear to circular polarization: two crossed dipoles phase shifted by 90° .

- ▶ One of the transceiver is mobile (drone, satellite ...),
- ▶ the power exchanged by the two transceivers in linear polarization depends on relative angle ϑ as $\cos(\vartheta)^2$,
- ▶ hence, loss of signal if polarizations are orthogonal.

hans.mayer.tv/html/crossdipole137.html

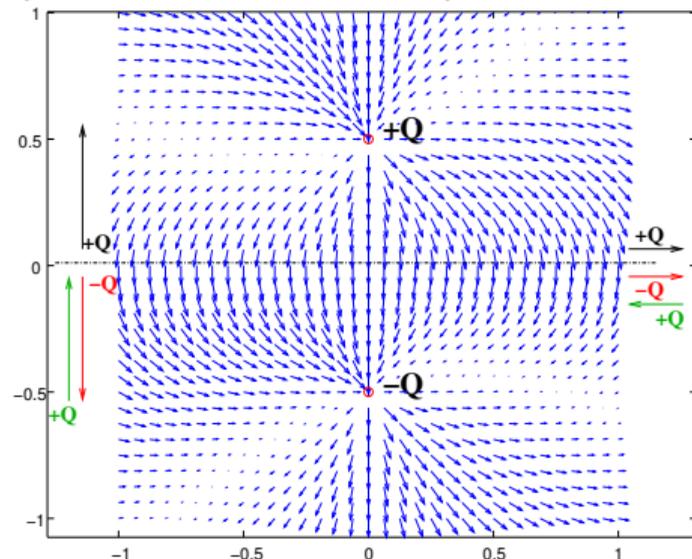
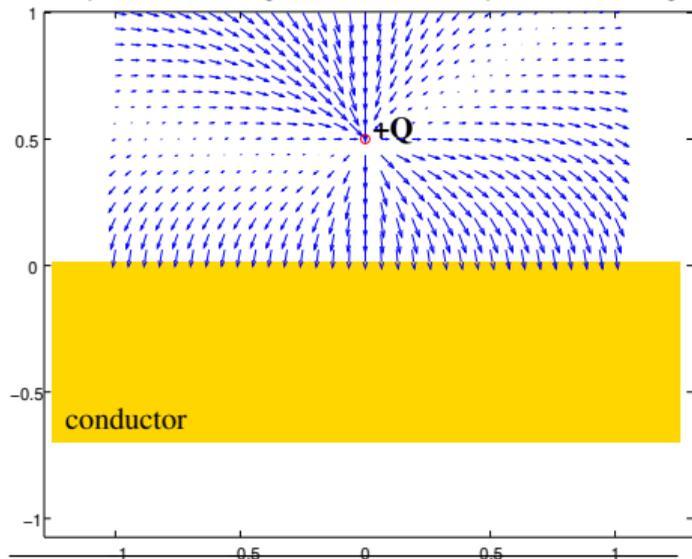
A, B, C, D: 510 mm
E: 50Ω , 361 mm
F: $75 \Omega \times 2$, 361 mm
G: 50Ω

<http://jccoppens.com/ant/qfh/img/noaa17qfh.jpg> →



The monopole

- ▶ Mirror charge principle^{26 27}: a radiating element located over a conducting plane is equivalent to this radiating element + its copy mirrored by symmetry wrt metallic plane, meeting the conditions for a **cancellation (zero) of the electric field in the metal**
- ▶ monopole = radiating element over a ground plane, taking advantage of the virtual mirror charges to convert the monopole into a dipole (radiating element in horn antennas)
- ▶ dipole = $73 + j43 \Omega \rightarrow$ monopole = $36.5 + j21.25 \Omega \Rightarrow$ shorten dipole to remove inductive component

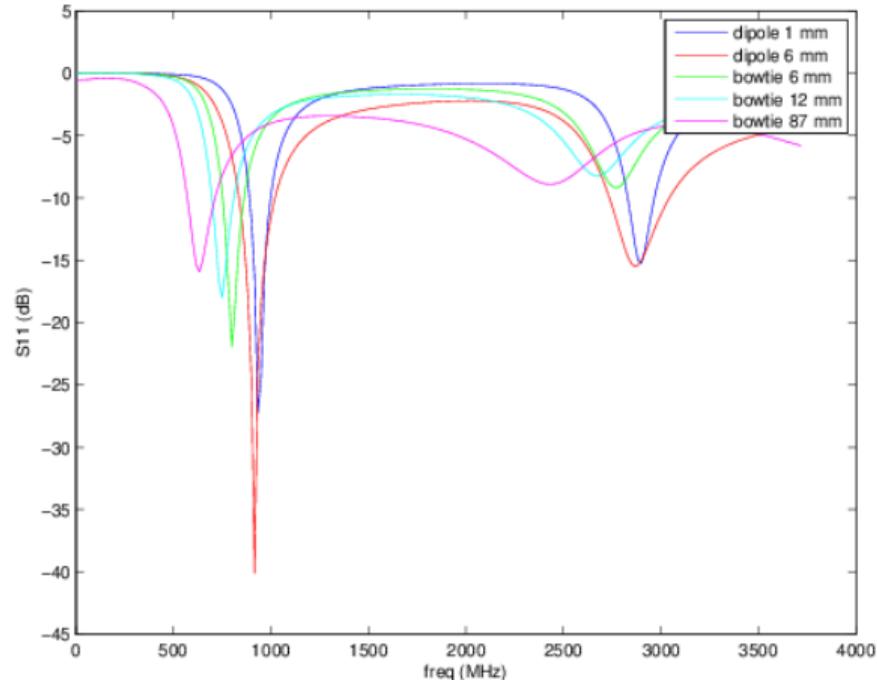


²⁶ J.D. Kraus & K.R. Carver, *Electromagnetics*, McGraw-Hill (1981), p.295

²⁷ R.P. Feynman & al., *The Feynman lectures on physics, vol.2* (1964), Chap 6-7 "The method of images" at www.feynmanlectures.caltech.edu/II_06.html

Antennas in complex media

- ▶ NEC2 can only handle wire-antennas in air
- ▶ homogeneous dielectric media: scale by $n = \sqrt{\epsilon_r}$
- ▶ heterogeneous media or radiating surfaces: FDTD (*Finite Difference–Time Domain*)
- ▶ excitation of the radiating element by a broadband pulse and system response obtained by FFT (time domain response)
- ▶ free software implementation: GPRMax 3D (www.gprmax.com/)



```
#title: Bowtie antenna
#domain: 0.10 0.050 0.200
#dx_dy_dz: 0.001 0.001 0.001
#time_window: 60e-9

#waveform: gaussian 1 1e9 mypulse
#transmission_line: z 0.05 0.025 0.100 73 mypulse

#triangle: 0.00625 0.025 0.025 0.093750 0.025 0.025 0.05 0.025 0.100 0 pec
#triangle: 0.00625 0.025 0.175 0.093750 0.025 0.175 0.05 0.025 0.101 0 pec
#plate: 0.049 0.025 0.09 0.051 0.025 0.100 pec
#plate: 0.049 0.025 0.101 0.051 0.025 0.11 pec

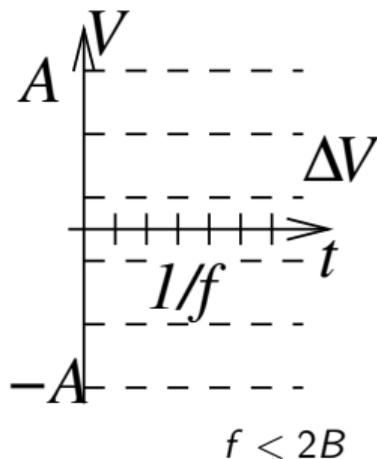
#geometry_view: 0.001 0.020 0.020 0.10 0.025 0.180 0.001 0.001 0.001 antenna_wire_dipole_fs f
```



Channel capacity ³⁰ ³¹

1. Quantized ²⁸ /noisy ²⁹ measurement of the signal carrying the information: ΔV
2. Amplitude of the received signal: A
3. Discrete levels measured: $1 + A/\Delta V$
4. Transmission bandwidth (Nyquist): $2 \cdot B$

Hartley: $C = 2B \cdot \log_2(1 + A/\Delta V)$ Shannon: $C = B \cdot \log_2(1 + SNR)$ (amplitude \rightarrow power ; quantization \rightarrow noise (SNR))
--



²⁸R.V.L. Hartley *Transmission of Information*, Bell System Technical Journal (1928)

²⁹C. E. Shannon, Communication in the presence of noise, Proc. IRE **37** (1) 10–21 (1949)

³⁰ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-450-principles-of-digital-communications-i-fall-2006/

³¹O. Rioul & J.C. Magossi, *On Shannon's Formula and Hartley's Rule: Beyond the Mathematical Coincidence*, Entropy **16** 4892–4910 (2014), & O. Rioul, *Shannon 100* (26/10/2016) at www.youtube.com/watch?v=mBrI0SaPRDc

Channel capacity

1. Quantized ²⁸ /noisy measurement of the signal carrying the information: ΔV
2. Amplitude of the received signal: A
3. Discrete levels measured: $1 + A/\Delta V$
4. Transmission bandwidth (Nyquist): $2 \cdot B$

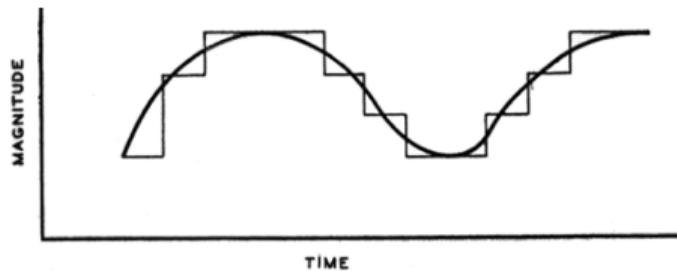


Fig. 3

A continuous curve may be thought of as the limit approached by a curve made up of successive steps, as shown in Fig. 3, when the interval between the steps is made infinitesimal. An imperfectly defined curve may then be thought of as one in which the interval between the steps is finite. The steps then represent primary selections. The number of selections in a finite time is finite. Also the change made at each step is to be thought of as limited to one of a finite number of values. This means that the number of available symbols is kept finite. If this were not the case, the curve would be

$$\text{Hartley: } C = 2B \cdot \log_2(1 + A/\Delta V)$$

$$\text{Shannon: } C = B \cdot \log_2(1 + SNR)$$

(amplitude \rightarrow power ; quantization \rightarrow noise (SNR))

²⁸R.V.L. Hartley *Transmission of Information*, Bell System Technical Journal (1928)

Channel capacity

1. Quantized /noisy²⁸ measurement of the signal carrying the information: ΔV
2. Amplitude of the received signal: A
3. Discrete levels measured: $1 + A/\Delta V$
4. Transmission bandwidth (Nyquist): $2 \cdot B$

Theorem 2: Let P be the average transmitter power, and suppose the noise is white thermal noise of power N in the band W . By sufficiently complicated encoding systems it is possible to transmit binary digits at a rate

$$C = W \log_2 \frac{P + N}{N} \quad (19)$$

with as small a frequency of errors as desired. It is not possible by any encoding method to send at a higher rate and have an arbitrarily low frequency of errors.

This shows that the rate $W \log(P + N)/N$ measures in a sharply defined way the capacity of the channel for transmitting information. It is a rather surprising result, since one would expect that reducing the frequency of errors would require reducing the rate of transmission, and that the rate must approach zero as the error frequency does. Actually, we can send at the rate C but reduce errors by using more involved encoding and longer delays at the transmitter and receiver. The transmitter will take

Hartley: $C = 2B \cdot \log_2(1 + A/\Delta V)$

Shannon: $C = B \cdot \log_2(1 + SNR)$

(amplitude \rightarrow power ; quantization \rightarrow noise (SNR))

²⁸C. E. Shannon, Communication in the presence of noise, Proc. IRE **37** (1) 10–21 (1949)

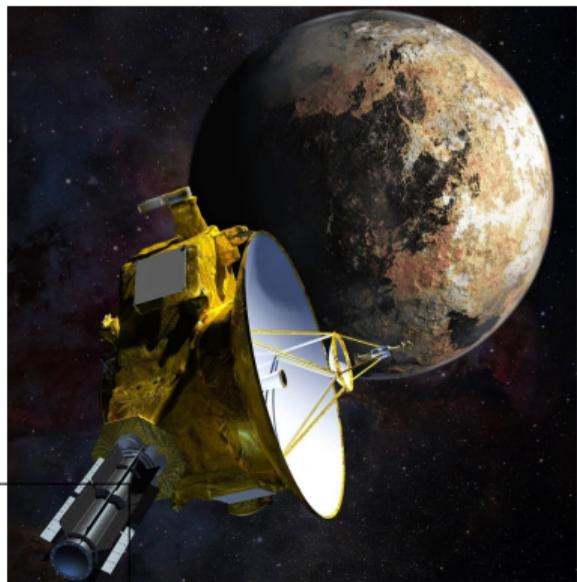
Channel capacity

1. Quantized ²⁸ /noisy ²⁹ measurement of the signal carrying the information: ΔV
2. Amplitude of the received signal: A
3. Discrete levels measured: $1 + A/\Delta V$
4. Transmission bandwidth (Nyquist): $2 \cdot B$

Hartley: $C = 2B \cdot \log_2(1 + A/\Delta V)$

Shannon: $C = B \cdot \log_2(1 + SNR)$

(amplitude \rightarrow power ; quantization \rightarrow noise (SNR))



<http://www.bbc.com/news/science-environment-30746650> : “At a distance of 5bn km and with a 15-watt transmitter, New Horizons will downlink its information at 3,000 bits per second - at best. If you can bear to recall the bad old days of dial-up internet, you’ll realise this is painfully slow.” (9 January 2015, BBC: ‘Planet’ Pluto comes into view) ³⁰

²⁸R.V.L. Hartley *Transmission of Information*, Bell System Technical Journal (1928)

²⁹C. E. Shannon, Communication in the presence of noise, Proc. IRE **37** (1) 10–21 (1949)

³⁰<http://www.mike-willis.com/Tutorial/PF13.htm> &

Channel capacity

1. Quantized /noisy measurement of the signal carrying the information: ΔV
2. Amplitude of the received signal: A
3. Discrete levels measured: $1 + A/\Delta V$
4. Transmission bandwidth (Nyquist): $2 \cdot B$

Hartley: $C = 2B \cdot \log_2(1 + A/\Delta V)$

Shannon: $C = B \cdot \log_2(1 + SNR)$

(amplitude \rightarrow power ; quantization \rightarrow noise (SNR))



<https://science.nasa.gov/learn/basics-of-space-flight/chapter18-2/>

1. $10 \log_{10}(15 \text{ W}) = 42 \text{ dBm}$ emitted @ 10 GHz
2. $FSPL = \left(\frac{4\pi \cdot d \cdot f}{c}\right)^2 = 20 \log_{10}(d) + 20 \log_{10}(f) - 147.55 \text{ dB}$ in SI units, so $FSPL=306 \text{ dB}$
3. $P_R = -306 + 42 + 77 = -187 \text{ dBm}$ for a 77 dB gain parabola (70 m diameter dish –
 $G = A/\lambda^2 \rightarrow 10 \cdot \log_{10}(\pi D/\lambda)^2$
4. Sky background noise: $k_B T = 1.38 \cdot 10^{-23} \cdot 20 \text{ K} = -186 \text{ dBm/Hz}$
5. For small SNR, $\log(1 + SNR) \simeq SNR \Rightarrow C \simeq \frac{B \times P_R}{B \times k_B \cdot T} = \frac{P_R}{k_B \cdot T} \forall B$
6. $-187 - (-186) = -1 \text{ dB} \rightarrow 10^{-1/10} \simeq 0.8 \text{ b/s}$

Channel capacity

1. Quantized /noisy measurement of the signal carrying the information: ΔV
2. Amplitude of the received signal: A
3. Discrete levels measured: $1 + A/\Delta V$
4. Transmission bandwidth (Nyquist): $2 \cdot B$

Hartley: $C = 2B \cdot \log_2(1 + A/\Delta V)$

Shannon: $C = B \cdot \log_2(1 + SNR)$

(amplitude \rightarrow power ; quantization \rightarrow noise (SNR))



<https://science.nasa.gov/learn/basics-of-space-flight/chapter18-2/>

1. $10 \log_{10}(15 \text{ W}) = 42 \text{ dBm}$ emitted @ 10 GHz
2. $FSPL = \left(\frac{4\pi \cdot d \cdot f}{c}\right)^2 = 20 \log_{10}(d) + 20 \log_{10}(f) - 147.55 \text{ dB}$ in SI units, so $FSPL=306 \text{ dB}$
3. $P_R = -306 + 42 + 77 = -187 \text{ dBm}$ for a 77 dB gain parabola (70 m diameter dish –
 $G = A/\lambda^2 \rightarrow 10 \cdot \log_{10}(\pi D/\lambda)^2$
4. Sky background noise: $k_B T = 1.38 \cdot 10^{-23} \cdot 20 \text{ K} = -186 \text{ dBm/Hz}$
5. For small SNR, $\log(1 + SNR) \simeq SNR \Rightarrow C \simeq \frac{B \times P_R}{B \times k_B \cdot T} = \frac{P_R}{k_B \cdot T} \forall B$
6. **Additional information: HGA is 2.1 m parabola=47 dBi gain @ 10 GHz**
 $\Rightarrow C = -1 + 47 = 46 \text{ dB} \Rightarrow 10^{46/10} = 40000 \text{ bps} (\geq 3000 \text{ bps})$

Asymptotic limits ²⁸

SNR dependent channel capacity with bandwidth:

$$\text{thermal noise} = k_B T \times B \Rightarrow \text{SNR} = \frac{P_E}{k_B T \times B}$$

High SNR ($\gg 1$, PLC or DSL)

$$C = B \log_2(1 + \text{SNR}) \simeq B \times \log_2(\text{SNR})$$

doubling $B=B'$ yields

$$C' = B' \log_2(\text{SNR}) = B' \log_2\left(\frac{P_E}{k_B T B'}\right) = 2B \log_2\left(\frac{P_E}{2k_B T B}\right)$$

$$= 2B \left(\log_2\left(\frac{P_E}{k_B T B}\right) - 1 \right)$$

$$\simeq 2B \log_2\left(\frac{P_E}{k_B T B}\right) \text{ since } \text{SNR} \gg 1$$

$$\Rightarrow C' \simeq 2C$$

Doubling B doubles C

bandwidth limited regime

Power-line communication (PLC), digital subscriber line (DSL)

Low SNR ($\ll 1$, space link or DVB-T):

$$B \log_2(1 + \text{SNR}) \simeq B \times \text{SNR}$$

$$= B \times \frac{P_E}{k_B T B} = \frac{P_E}{k_B T}$$

No impact of B , only P_E defines C

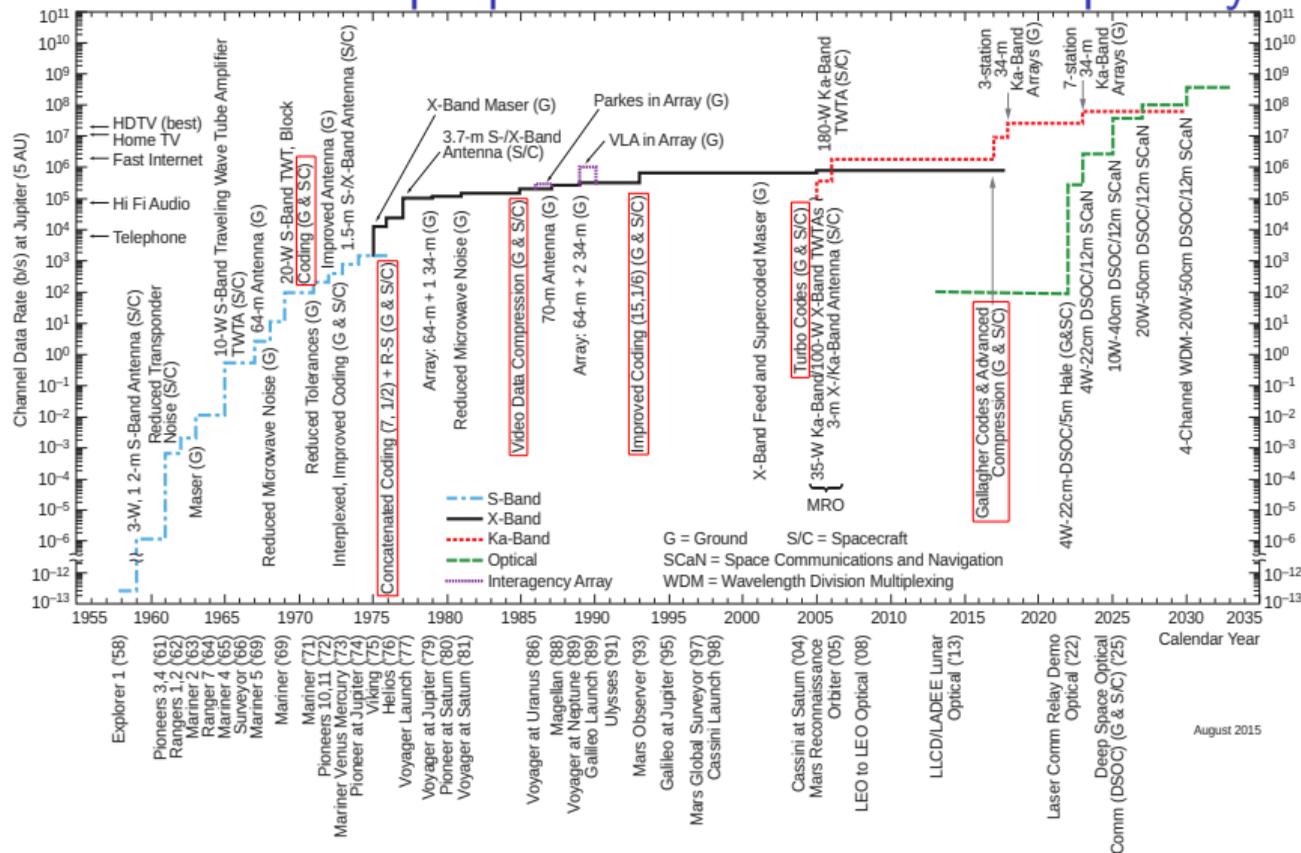
power limited regime

Spatial communications, deep space or LEO satellites

²⁸J.-M Friedt, H. Boeglen, *Décodage d'images numériques issues de satellites météorologiques en orbite basse : le protocole LRPT de Meteor-M2 (partie 3/3)*, GNU/Linux Magazine France **228** (Jul/Aug 2019) at

http://jmfriedt.free.fr/glmf_meteor3.pdf

Evolution of Deep Space Communications Capability²⁹



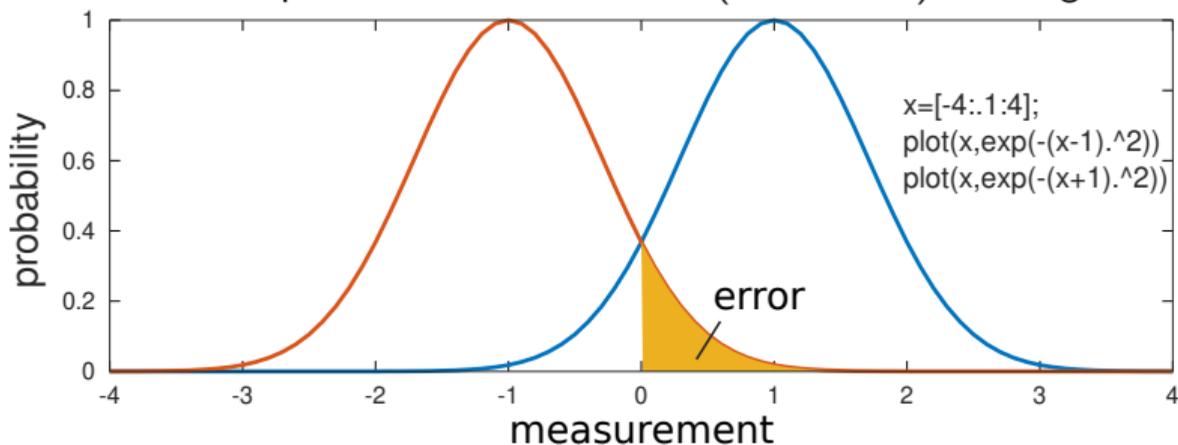
Name	Frequency range
VHF	30–300 MHz
UHF	300–3000 MHz
L	1–2 GHz
S	2–4 GHz
C	4–8 GHz
X	8–12 GHz
Ku	12–18 GHz
K	18–27 GHz
Ka	27–40 GHz
E	60–90 GHz

Growth of normalized deep space communications capacity. (Note: major jumps in capability result from frequency band changes.)

²⁹<https://descanso.jpl.nasa.gov/performmetrics/profileDSCC.html>

Bit Error Rate (BER)

Gaussian noise spreads the measurements (here BPSK) and might select the wrong bit state



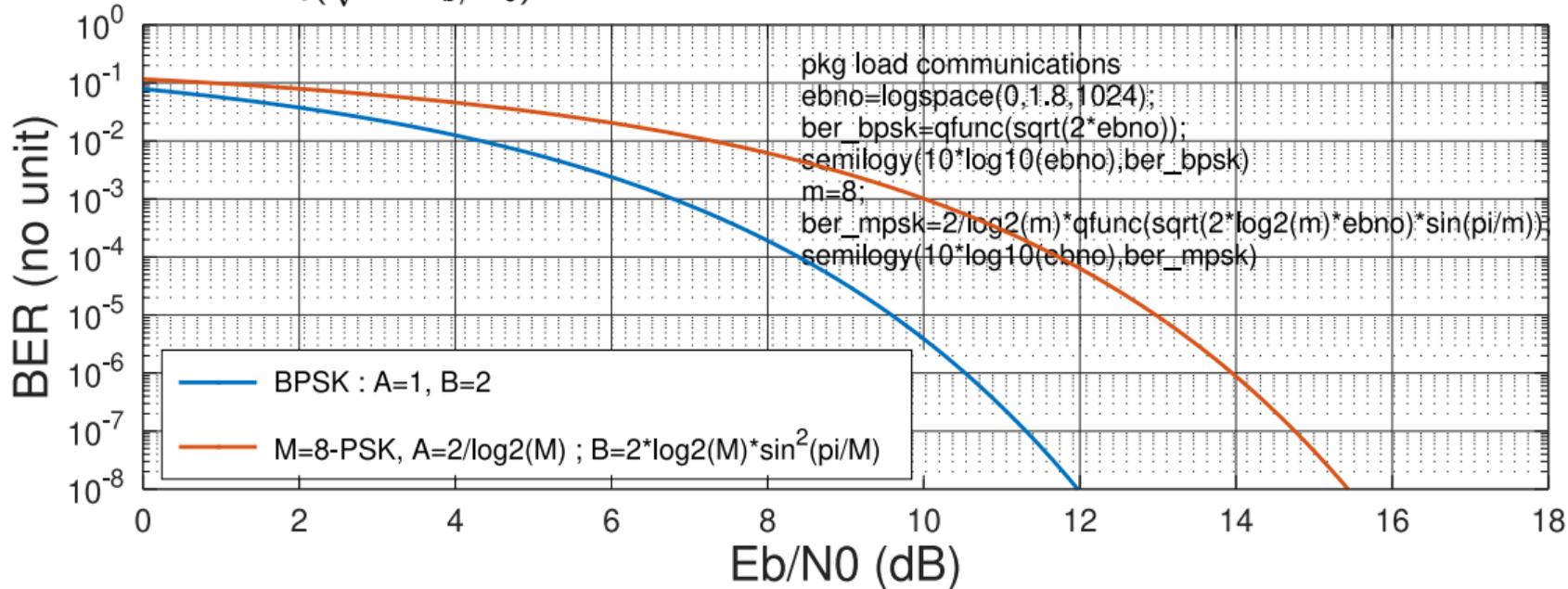
$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) \cdot du \text{ (yellow part)}$$

GNU/Octave communication toolbox `qfunc()` or `erfc()` functions: $Q(x) = \frac{1}{2}\text{erfc}(x/\sqrt{2})$

Abscissa: energy in each symbol to noise power ratio E_b/N_0

Bit Error Rate (BER) ³⁰

Plot $BER = A \times Q(\sqrt{B \cdot E_b/N_0})$



Example: if $BER < 10^{-6}$ (1 bit in 1 Mb), then $E_b/N_0 > 10.5$ dB (E_b/N_0 related to SNR by spectral efficiency (equal to 2 in case of QPSK))

³⁰J.-M Friedt, H. Boeglen, *Décodage d'images numériques issues de satellites météorologiques en orbite basse : le protocole LRPT de Meteor-M2 (partie 3/3)*, GNU/Linux Magazine France **228** (Jul/Aug 2019) at http://jmfriedt.free.fr/glmf_meteor3.pdf

On the need for a good antenna ... Galileo

The 4.8-meter (16-foot) wide, umbrella-like high-gain antenna is mounted at the top of the spacecraft. When unfurled, the antenna's hosiery-like wire mesh stretches over 18 umbrella ribs to form a large parabolic dish. Galileo was to have used this dish to radio its scientific data from Jupiter. This high-performance, X-band antenna was designed to transmit data back to Earth at rates of up to 134,400 bits of digital information per second (the equivalent of about one imaging frame each minute).

[...]

The Low-Gain Antenna

The difference between Galileo sending its data to Earth using the high-gain antenna and the low-gain is like the difference between the concentrated light from a spotlight versus the light emitted diffusely from a bare bulb. If unfurled, the high-gain would transmit data back to Deep Space Network (DSN) collecting antennas in a narrowly focused beam. The low-gain antenna transmits in a comparatively unfocused broadcast, and only a tiny fraction of the signal actually reaches DSN receivers. Because the received signal is 10,000 times fainter, data must be sent at a lower rate to ensure that the contents are clearly understood.

[...]

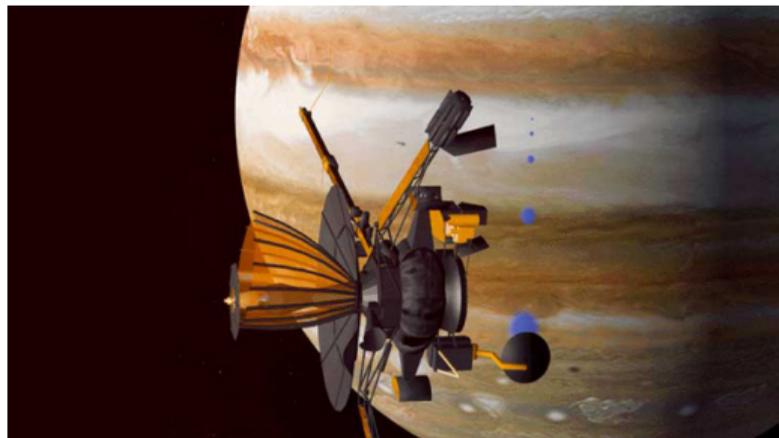
Without any new enhancements, the low-gain antenna's data transmission rate at Jupiter would be limited to only 8-16 bits per second (bps), compared to the high-gain's 134,400 bps.

31 32 Analysis:

- ▶ X-band = 10 GHz $\Rightarrow G = 10 \times \log_{10} \left(\left(\pi \frac{D}{\lambda} \right)^2 \right) = 54 \text{ dBi} = 250000$
- ▶ lost antenna gain: $134000 / 250000 \simeq 0.5 \text{ bps} \rightarrow 2.5 \text{ bps with } 7 \text{ dBi LGA}$ (<https://www2.jpl.nasa.gov/galileo/faqgen.html>)

³¹https://www2.jpl.nasa.gov/galileo/hga_fact.html

³²https://descanso.jpl.nasa.gov/monograph/series13/DeepCommo_Chapter4--141029.pdf



On the need for a good antenna ... Galileo

The 4.8-meter (16-foot) wide, umbrella-like high-gain antenna is mounted at the top of the spacecraft. When unfurled, the antenna's hosiery-like wire mesh stretches over 18 umbrella ribs to form a large parabolic dish. Galileo was to have used this dish to radio its scientific data from Jupiter. This high-performance, X-band antenna was designed to transmit data back to Earth at rates of up to 134,400 bits of digital information per second (the equivalent of about one imaging frame each minute).

[...]

The Low-Gain Antenna

The difference between Galileo sending its data to Earth using the high-gain antenna and the low-gain is like the difference between the concentrated light from a spotlight versus the light emitted diffusely from a bare bulb. If unfurled, the high-gain would transmit data back to Deep Space Network (DSN) collecting antennas in a narrowly focused beam. The low-gain antenna transmits in a comparatively unfocused broadcast, and only a tiny fraction of the signal actually reaches DSN receivers. Because the received signal is 10,000 times fainter, data must be sent at a lower rate to ensure that the contents are clearly understood.

[...]

Without any new enhancements, the low-gain antenna's data transmission rate at Jupiter would be limited to only 8-16 bits per second (bps), compared to the high-gain's 134,400 bps.

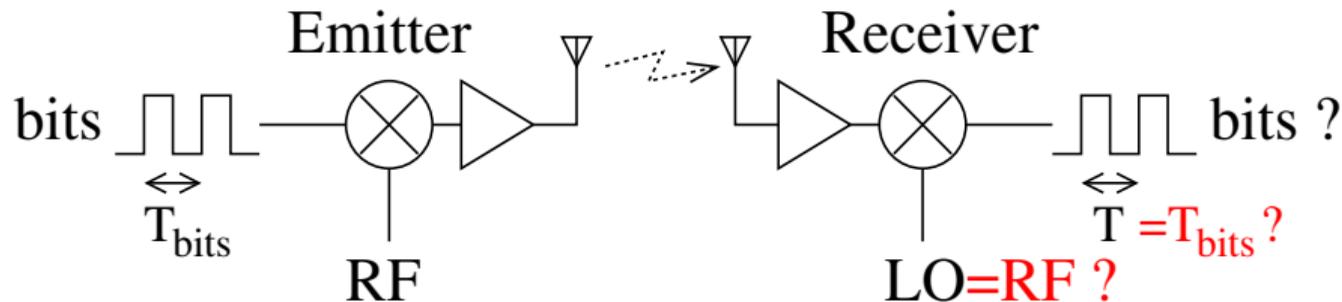
31 32 Analysis:

- ▶ X-band = 10 GHz $\Rightarrow G = 10 \times \log_{10} \left(\left(\pi \frac{D}{\lambda} \right)^2 \right) = 54 \text{ dBi} = 250000$
- ▶ lost antenna gain: $134000/250000 \simeq 0.5 \text{ bps} \rightarrow 2.5 \text{ bps}$ with 7 dBi LGA (<https://www2.jpl.nasa.gov/galileo/faqgen.html>)
- ▶ **S-band** low gain antenna: 2-4 GHz
- ▶ $P_R = G_1 G_2 \left(\frac{\lambda}{4\pi d} \right)^2 = G_1 \frac{A}{\lambda^2} \left(\frac{\lambda}{4\pi d} \right)^2 = G_1 \left(\frac{A}{4\pi d} \right)^2$, $\forall \lambda$ for parabolic receiver with area A



Digital stream synchronization

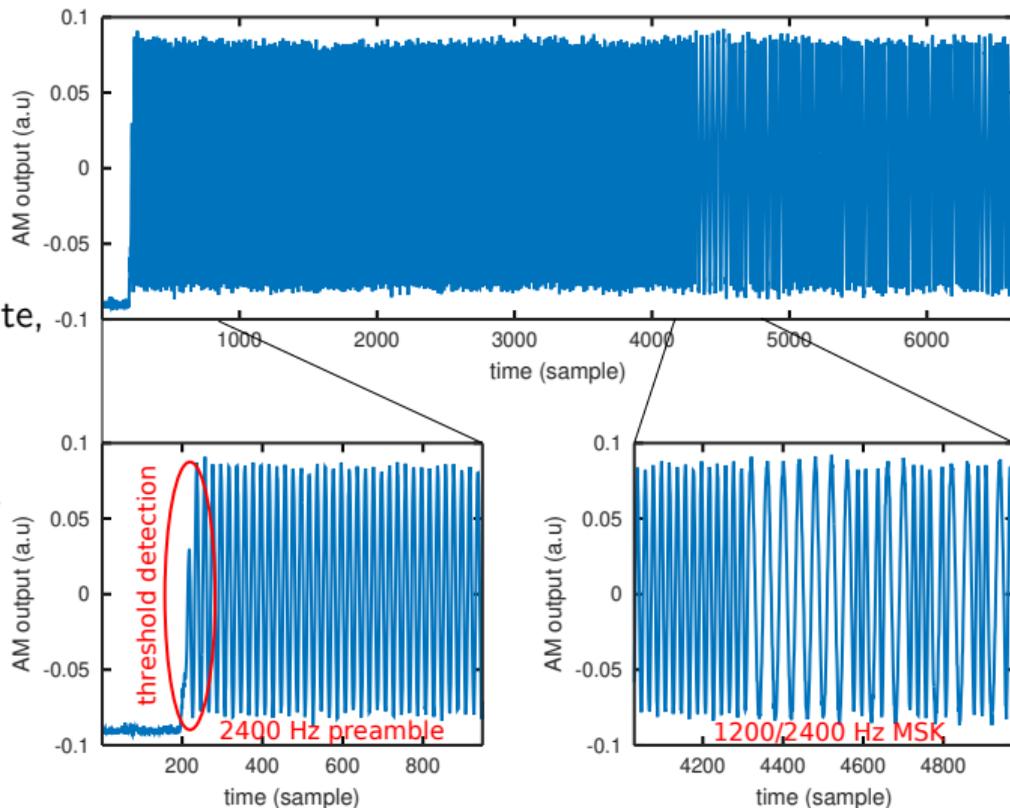
1. Synchronizing LO on RF: demodulation
2. Nominal data rate generated by a remote oscillator on the emitter \Rightarrow synchronization of the sampling times on the receiver
3. Startup sequence ("start bit"): ACARS 16-char long pre-key
4. Feedback on bit transitions (**clock recovery**)



ACARS and gr-acars ³³

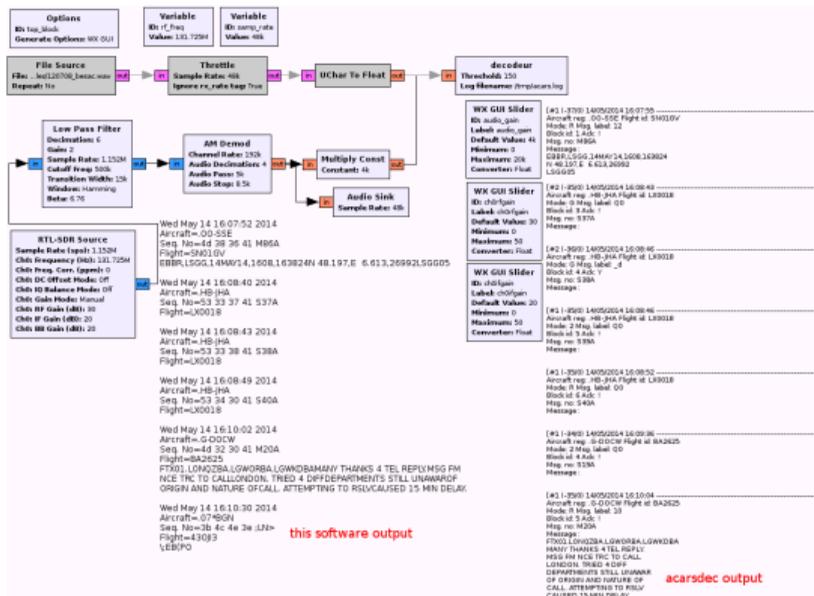
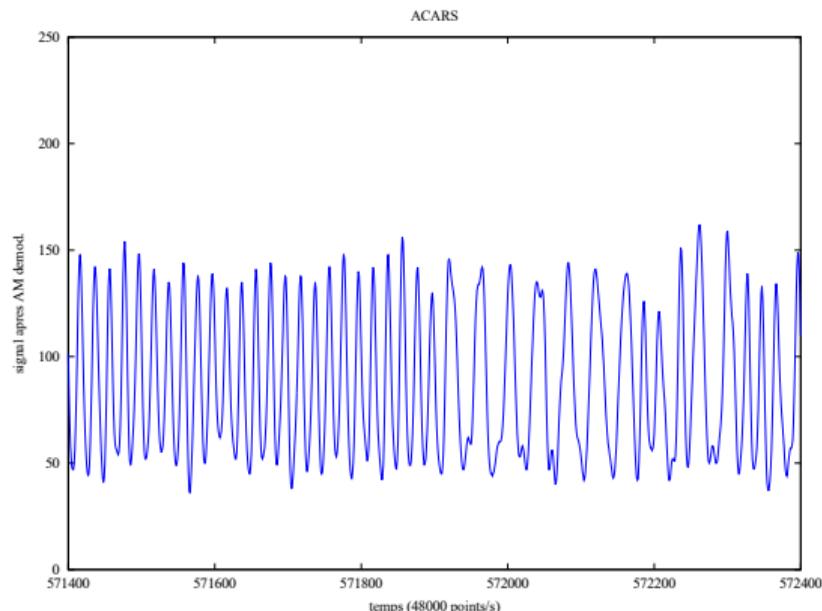
Aircraft Communications Addressing and Reporting System:

- ▶ Protocol developed in the 70s for communicating between plane pilots and ground
- ▶ Telemetry and free text
- ▶ As with all aeronautical protocols, AM mod.
- ▶ Information is MSK modulated at 2400 bps: 0=half 1200 Hz period, 1=full 2400 Hz period
- ▶ Differential encoding (0=keep bit state, 1=change bit state)
- ▶ Header: synchronization using 16 bytes with ones only (2400 Hz)
- ▶ Three channels in European: primary (131.725 MHz), secondary (131.525 MHz), additional (131.825 MHz, 131.850 MHz ?)



Digital mode demodulation: ACARS (prototyping)

How to convert the raw signal to a useful information ³⁴ ?



³⁴J.-M Friedt, G. Goavec-Mérou, *La réception radiofréquence définie par logiciel (Software Defined Radio – SDR)*, GNU/Linux Magazine France **153** (Octobre 2012), pp.4-33, available at http://jmfriedt.free.fr/lm_sdr.pdf

Digital mode demodulation: ACARS (prototyping)

Description of the encoding: www.radioscanner.ru/files/download/file4094/acars.pdf

VHF-3 Transceiver

(a) The MU encodes the digital data for transmission as a series of 1200 Hz and 2400 Hz tones at 2400 baud (future ACARS systems may operate with 2400/4800 Hz at 4800 baud). A 1200 Hz tone indicates a bit change from the previous bit (0 to 1 or 1 to 0) while a 2400 Hz tone indicates no bit change (0 to 0 or 1 to 1).



Convolution with two filters (1200 & 2400 Hz centered):

$$\int_0^1 \sin(2\pi t) \cdot \sin(\pi t) \propto \int_0^1 (\cos(3\pi t) - \cos(\pi t)) \\ = (\sin(3\pi) - \sin(0)) - (\sin(\pi) - \sin(0)) = 0$$

but

$$\int_0^1 \sin(2\pi t) \cdot \sin(2\pi t) \propto \int_0^1 (\cos(4\pi t) - \cos(0)) \\ = (\sin(4\pi) - \sin(0)) + (1) \neq 0$$

Bit decoding

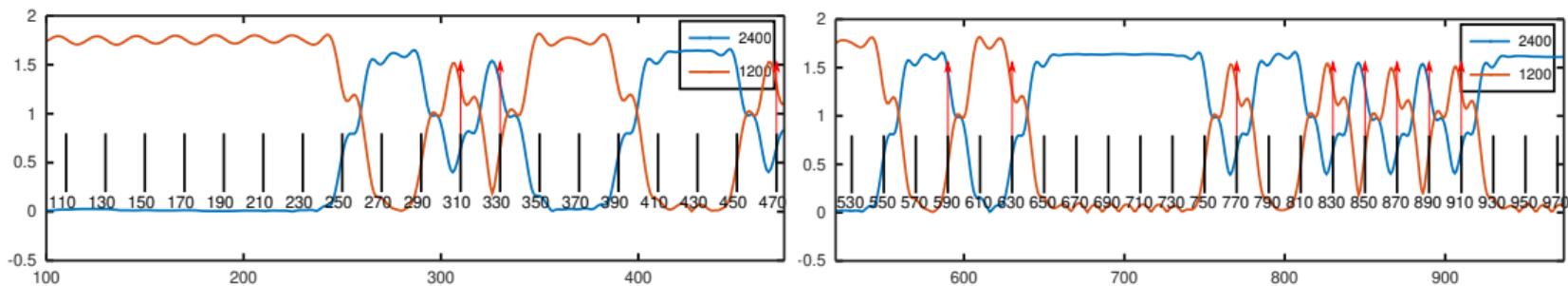
- ▶ Two tones define the two bit states: convolution³⁵ with one period = bandpass filter

```
fs=48000;
bps=2400;
f2400=exp(j*2*pi*2400*[0:1/fs:1/bps*2]'); f2400=f2400(1:end-1);
f1200=exp(j*2*pi*1200*[0:1/fs:1/bps*2]'); f1200=f1200(1:end);
% s2400=abs(conv(y, f2400)); s2400=s2400(200:end); % convolution
% s1200=abs(conv(y, f1200)); s1200=s1200(200:end);
% b=firls(256,[0 3400 4500 fs/2]*2/fs,[1 1 0 0]); % low pass filter taps
% s2400=filter(b,1,s2400); s2400=s2400(200:end); % filter (remove 4800 Hz)
% s1200=filter(b,1,s1200); s1200=s1200(200:end);
sf=(fft(y));
f2400f=(fft(f2400, length(sf)));
f1200f=(fft(f1200, length(sf)));
sf2400f=sf.*f2400f; % convolution
sf1200f=sf.*f1200f;
df=fs/length(sf); % FFT bin width (in Hz)
fcut=floor(3500/df); % cutoff bin position
sf2400f(fcut:end-fcut)=0; % low pass filter/Matlab FFT convention (0 left, fs/2 middle)
sf1200f(fcut:end-fcut)=0;
s2400=abs(ifft(sf2400f)); % back to time domain
s1200=abs(ifft(sf1200f));
plot(s2400, '-'); hold on; plot(s1200, '-'); legend('2400', '1200')
```

³⁵ $FFT(x \circledast y) = FFT(x) \cdot FFT(y)$

Bit decoding

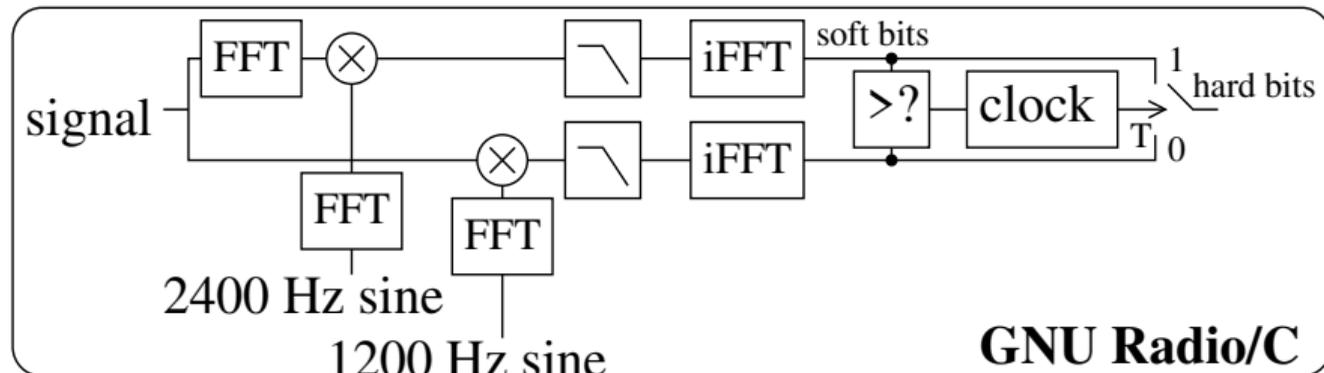
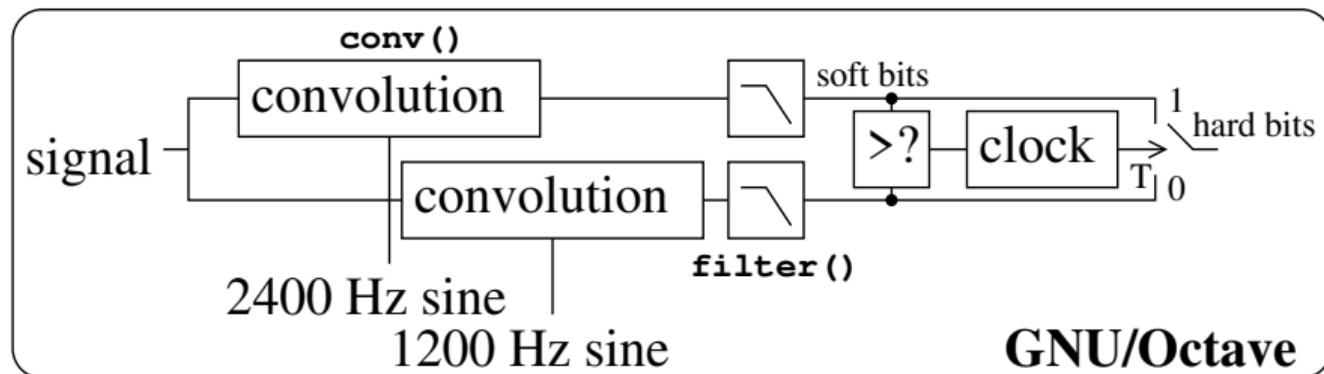
- ▶ Detect message as **rise in the standard deviation** of the received signal to avoid continuously running FFTs (might need refinement)
- ▶ Two tones define the two bit states: convolution with one period = bandpass filter
- ▶ Check every bit period the **most probable bit state**: soft bit → hard bit by comparison
- ▶ Sampling @ 48 kS/s and 2400 bps message: check state every 20 samples.



Soft-bit sequences following convolution with 2400 Hz (blue) and 1200 (red) bandpass filters. Red arrows: **poor sampling** time due to initial poor threshold detection.

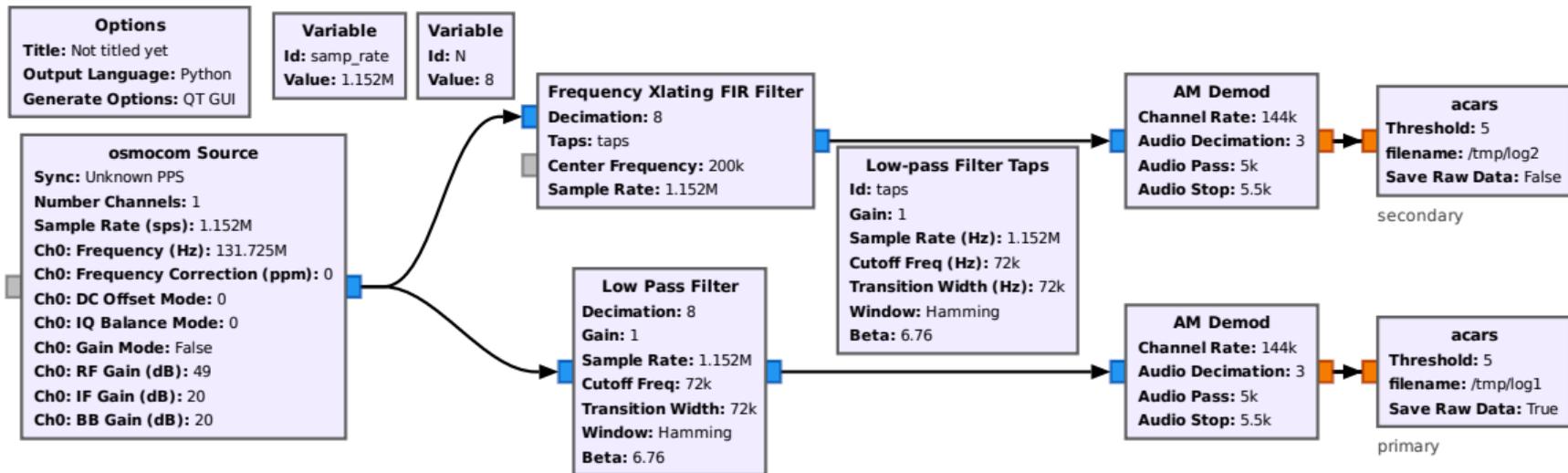
GNU Octave to GNU Radio

- ▶ Convolution (Octave/Matlab `conv()`) as FFT product
- ▶ Filtering as FFT weighing



Bit decoding: FFT implementation

FFTW → GNU Radio FFT



The FFTW initialization is **not** thread safe:

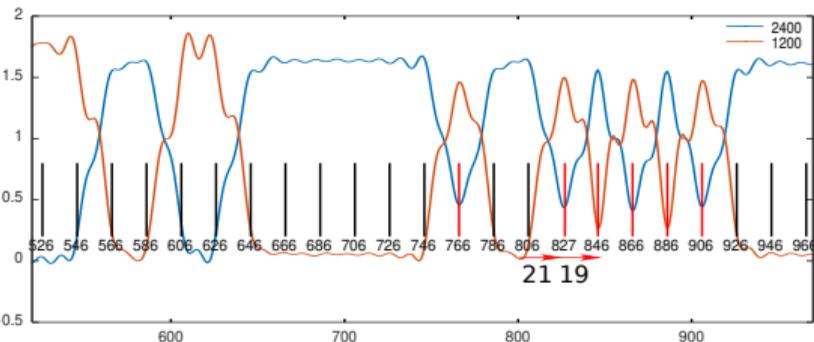
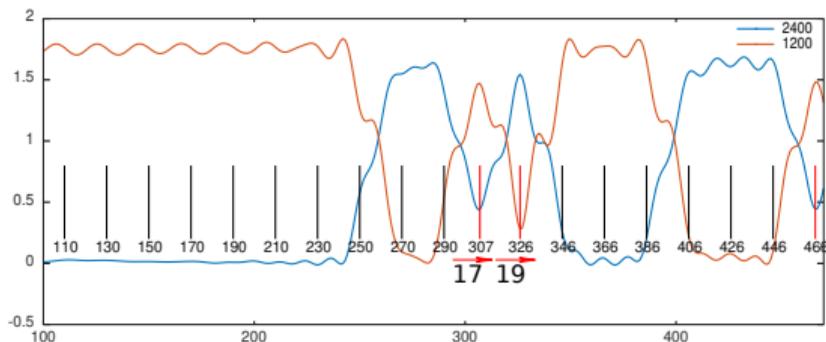
http://www.fftw.org/fftw3_doc/Thread-safety.html *planner routines share data (e.g. wisdom and trigonometric tables) between calls and plans ... Neither strategy works, however, in the following situation. The “application” is structured as a set of “plugins” which are unaware of each other, and for whatever reason the “plugins” cannot coordinate on grabbing the lock. (This is not a technical problem, but an organizational one ...)*

Clock synchronization

- ▶ Initial rise in preamble poorly defined (eg. Automatic Gain Control)
- ▶ Challenge in accurately finding the beginning of the sentence (sync. word → payload)
- ▶ Differential: any synchronization loss will last until end of sentence
- ▶ Challenge to keep long sentences consistent

⇒ need for **clock synchronization**

- ▶ **Proposed algorithm:** identify unique bit state (1 framed with 0s or 0 framed with 1s) and lock on signal maximum



Black vertical lines indicate when successive hard bits are expected to be the same, **red vertical lines indicate when a hard bit is framed with two different hard-bit states and are hence used for clock synchronization**

Results: single receiver, multiple channels

Structure du message ³⁶:

bits:	128	16	16	8	8	56	8	16	8	8	32	48	???	8	128	8
name	pre-key	bit sync	char sync	SOH	mode	addr	(n)ack	label	block ID	STX	Seq No.	Flight No.	Text	ETX	CRC	BCS
Eg.	010101 ...	0x2b 0x2a	0x16 0x16	0x01	0x45	.A7-...				02				03		

(2) Message format

The format of the messages exchanged between the ground and the aircraft is as follows (in the transmission order):

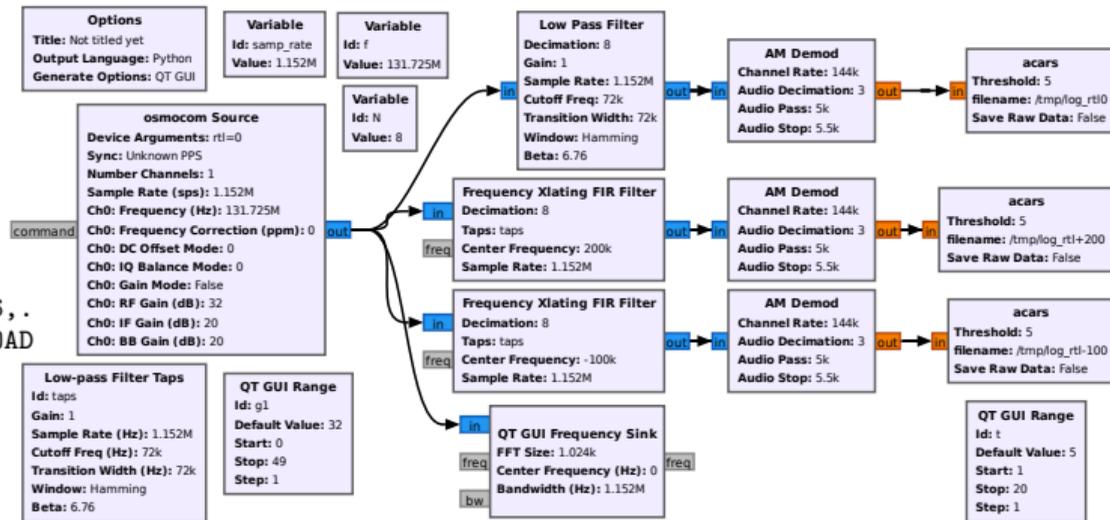
- Pre-key (16 characters)
- Bit synchro (2 characters)
- Character synchro (2 characters)
- Start of heading (1 character)
- Mode (1 character)
- Address (7 characters)

Example (hex and ASCII) ...

```
2b 2a 16 16 01 45 2e 41 37 2d ...
+*E.A7-BAKB02L82AQR3256/LSAG.AFN/FMHQTR3256,
A7rBAK,06A078,144616/FPON47427E007102,1/FCOADS
S,01/FCOATC,01247A
```

... interpreted as:

```
Aircraft=.A7-BAK
STX
Seq. No.=4c 38 32 41 L82A
Flight=QR3256
/LSAG.AFN/FMHQTR3256, .A7-BAK,06A078,144616/FPON47427E007102,1/FCOADS,01/FCOATC,01247AETX
```



³⁶http://www.scancat.com/Code-30_html_Source/acars.html

Results: single receiver, multiple channels

Fri May 1 17:53:22 2020
Aircraft=.HB-JMI
Seq. No=53 34 36 41 S46A
Flight=LX0018
ETX

Fri May 1 17:53:27 2020
Aircraft=.HB-JMI
Seq. No=53 34 37 41 S47A
Flight=LX0018
ETX

Fri May 1 19:32:18 2020
Aircraft=.B-5921
Seq. No=44 35 37 41 D57A
Flight=MU7080

#DFBR04/A33004,1,1C1,.B-5921,
20MAY01,17.19.51,LSGG,ZSPD,
CES7080 ,4000,013C2,016,
05.0,000000,D1333RR06COMU12,
000,012,012C3,015.0,01962,
0.252,111,1.53,111,10,0,01,
111,1.61,XC4,015.0,01859,
0.252,111C5,!:

Fri May 1 19:32:23 2020
Aircraft=.B-5921
Seq. No=44 35 37 42 D57B
Flight=MU7080
#DFB042335,07008,12610,
01345,00104,00016,00000,
00802,06066C6,042188,09517,
32847,01911,00104,00016,
00000,00802,16066C7,01,00,
00,00,00,00,01,00C8,01,00,
00,00,00,00,01,00N1,1.458,
1.459,1.458,1.611,082.08,

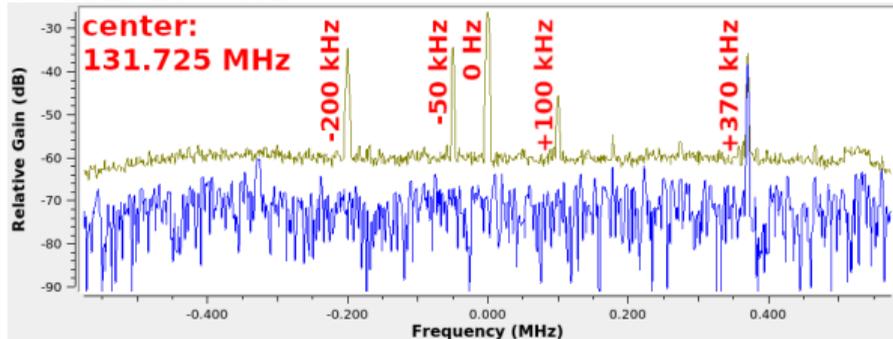
0,d

Fri May 1 19:32:30 2020
Aircraft=.B-5921
Seq. No=44 35 37 43 D57C
Flight=MU7080
#DFB92.6,093.9,0704,
21.152N2,1.459,1.459,1.458,
1.611,082.17,092.1,093.8,
0725,21.121S1,08895,14.35,
015.0,097.7,241.4,418.2,
546.4,08.9,078.0,078.0S2,
08886,14.36,015.0,097.3,
240.5,414.4,546.9,09.6,
077.9,077.F

Fri May 1 19:32:54 2020
Aircraft=.B-5921
Seq. No=44 35 37 47 D57G
Flight=MU7080
#DFB7,0487,0428T4,023,45,
151,073,097,119.8,022.6,
047.0,047,0488,0424V3,
0.15,0.05,0.09,042,0.25,
008.3V4,0.15,0.08,0.02,
191,0.19,008.0X1,1.477,
012.5,03050,0.237,2,20X2,
1.527,004.5,10025,0.395,0,
80X3,1.

Fri May 1 19:33:00 2020
Aircraft=.B-5921
Seq. No=44 35 37 48 D57H
Flight=MU7080
#DFB.567,-03.5,20011,0.660,
0,80X4,1.584,-12.7,25031,
0.731,0,80:ETX

← 131.725 MHz



131.825 MHz

Fri May 1 12:09:16 2020
Aircraft=..N22UB
Seq. No=53 30 32 41 S02A
Flight=ZD0001
ETX

Fri May 1 14:32:07 2020
Aircraft=.LX-YCV
Seq. No=53 32 30 41 S20A
Flight=CV7944
ETX

Fri May 1 17:52:30 2020
Aircraft=.F-GUOC
Seq. No=44 38 31 41 D81A
Flight=AF6746
#DFBOP DATAFUEL1552 33997
2.33 -0.3 6200 0 6500

12700 0.816 0.814 0.816 73600
72360TRUE TRUE -11 669 789
ETX

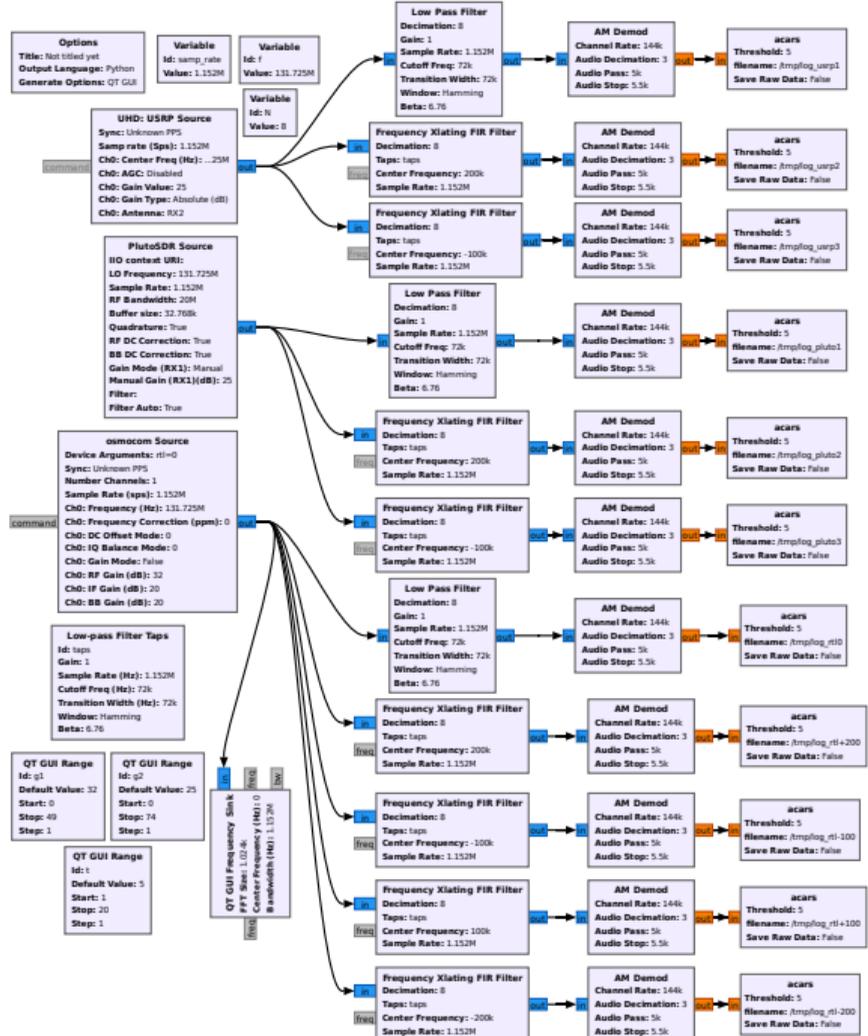
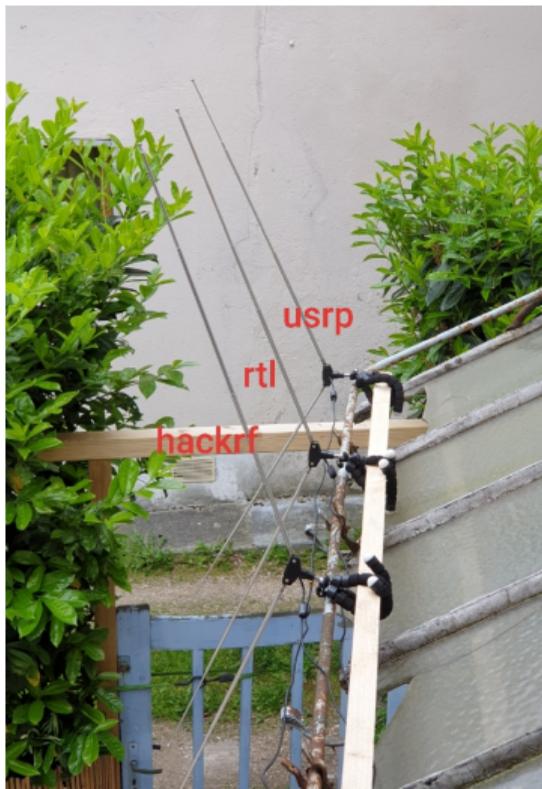
Sat May 2 16:46:21 2020
Aircraft=.A7-BAK
Seq. No=4c 38 32 41 L82A
Flight=QR3256
/LSAG.AFN/FMHQTR3256,
.A7-BAK,06A078,
144616/FPON47427E007102,1/FCOADS,
01/FCOATC,01247AETX

Sun May 3 07:34:07 2020
Aircraft=.LX-YCV
Seq. No=53 31 38 41 S18A
Flight=CV062M
ETX

Results: multiple receivers

Three sources (RTL-SDR, UHD for B200 and PlutoSDR) all connected to three ACARS channels

Each source connected to a different antenna



Results: multiple receivers

```
Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=05 4a 7f 44 J??

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
STX
Seq. Nos=43 30 37 41 C07#
Flight=AF6741
#CFBPLF 1 04#WY20 1155 AFF-GUOB AFB6741 KORD/LPG 3115-BOG-00#-17 R 0404 04#WY20FIE 343#9141 1123 04#WY20MSG 3442021 1 1123 0
4#WY20 IC 1 1 PLDB RADIO ALTIMETER TRANSCIVER (LEFT)NOMMSG 2719010 1 11

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=00 5e 7f 41 ??#

Wed Apr 22 18:24:24 1970
Aircraft:_F-GSOT
Seq. Nos=0 70 7f 00 p??

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=0c 25 7f 00 X??

Wed Apr 22 18:24:24 1970
Aircraft:_F-GSOT
Seq. No=74 5b 7f 00 t(?)

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=61 1b 7f 00 a??

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=43 30 37 44 C07#
Flight=AF6741
#CFBQUANTITY PROCESSOR UNITMSG 2557233 L 0404 04#WY20 ES H PLDB CHS MASTER CONTROL PANELMSG 2557023 L 0404 04#WY20 ES H PLDB CH
S MASTER CONTROL PANELMSG 2557070 L 0404 04#WY20 ES HDB CHS MASTER CONT#y

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=39 13 7f 44 9??#D

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=43 30 37 45 C07#
Flight=AF6741
#CFBROL PANELMSG 2557256 L 0404 04#WY20 ES H PLDB CHS MASTER CONTROL PANELEDRET#

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=40 30 7f 45 80??E

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=0 32 5b 53 02#S
Flight=CGLFFP
04#901N00233EV136975/ETX
"rtl-100.txt" [Modified] 392 lines --000--
317.1 962

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=05 4a 7f 06 J??

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
STX
Seq. Nos=43 30 37 41 C07#
Flight=AF6741
#CFBPLF 1 04#WY20 1155 AFF-GUOB AFB6741 KORD/LPG 3115-BOG-00#-17 R 0404 04#WY20FIE 343#9141 1123 04#WY20MSG 3442021 1 1123 0
4#WY20 IC 1 1 PLDB RADIO ALTIMETER TRANSCIVER (LEFT)NOMMSG 2719010 1 11

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=00 5e 7f 53 ??#S

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=43 30 37 42 C07#
Flight=AF6741
#CFBBS 04#WY20 DC H PLDB PRIMARY FLIGHT COMPUTER (RIGHT)DB PRIMARY FLIGHT COMPUTER (LEFT)DB PRIMARY FLIGHT COMPUTER (CENTER)MSG
2371711 A 0404 04#WY20 ES H PLDB DCH# IN LEFT AIRMSG 2371710 A 0404 0#

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=43 30 37 43 C07#
Flight=AF6741
#CFB#WY20 ES H PLDB DCH# IN RIGHT AIRMSG 2375001 A 0404 04#WY20 ES H PLDB FID# IN LEFT AIRMSG 2553026 L 0404 04#WY20 ES M PL
DB CHS MASTER CONTROL PANELMSG 2520050 A 0404 04#WY20 ES H PLDB FUEL #s

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=61 1b 7f 43 a??C

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=43 30 37 44 C07#
Flight=AF6741
#CFBQUANTITY PROCESSOR UNITMSG 2557233 L 0404 04#WY20 ES H PLDB CHS MASTER CONTROL PANELMSG 2557023 L 0404 04#WY20 ES H PLDB CH
S MASTER CONTROL PANELMSG 2557070 L 0404 04#WY20 ES HDB CHS MASTER CONT#y

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=39 13 7f 44 9??#D

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=43 30 37 45 C07#
Flight=AF6741
#CFBROL PANELMSG 2557256 L 0404 04#WY20 ES H PLDB CHS MASTER CONTROL PANELEDRET#

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. No=40 30 7f 45 80??E

Wed Apr 22 18:24:24 1970
Aircraft:_F-GUOB
Seq. Nos=0 32 5b 53 02#S
Flight=CGLFFP
DBDLHT_E#10A0BT #ETX
"uhd-100.txt" [Modified] 117 lines --1000--
117.1 Bot
```

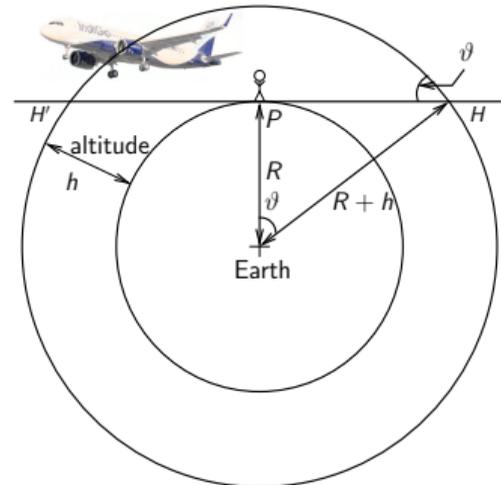
RTL-SDR (131.825 MHz)

B200 (131.825 MHz)

Red: identical messages by RTL-SDR and B200 ; Green: nearly identical messages by RTL-SDR and B200

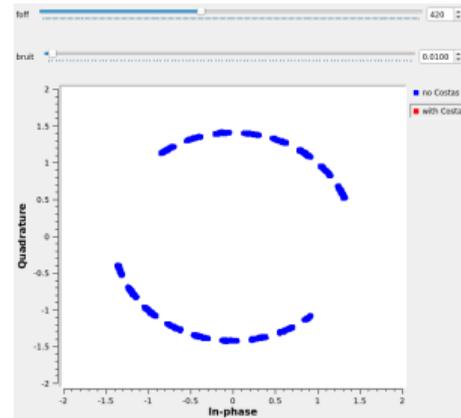
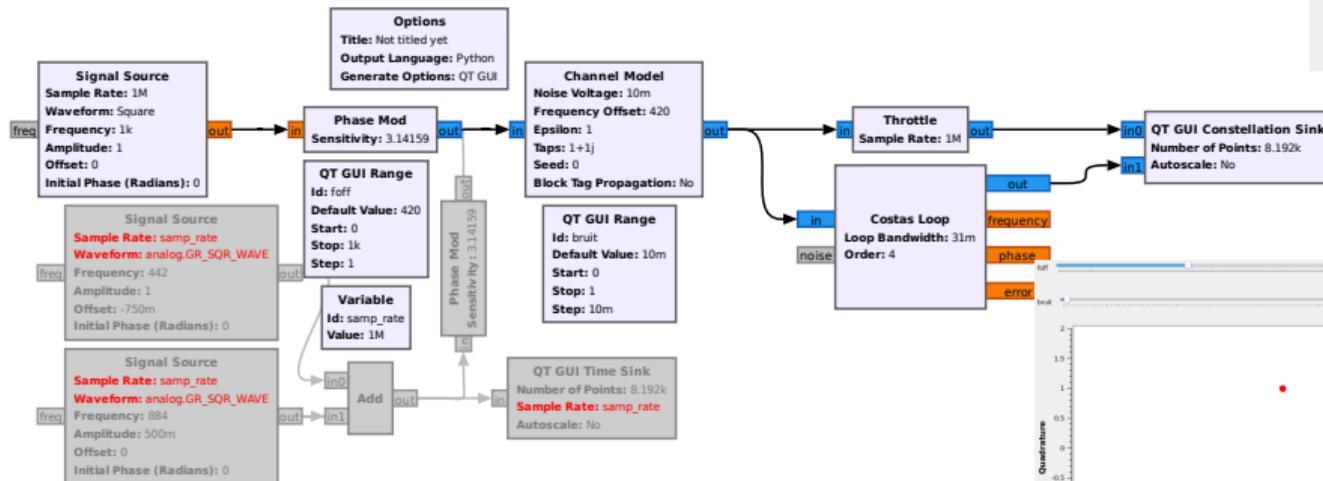
Communication range

An emitter flying at altitude h over the Earth with radius $R = 6400$ km can be detected (line of sight) at range $d = R \cdot \arccos\left(\frac{R}{R+h}\right)$
 \Rightarrow if $h = 10$ km: $d \simeq 350$ km (Paris !)

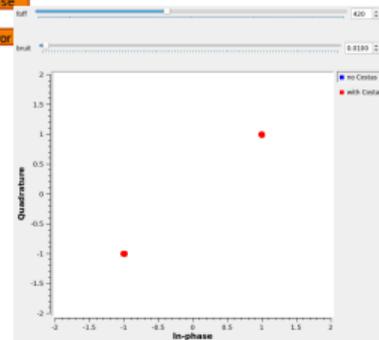


Constellation diagram

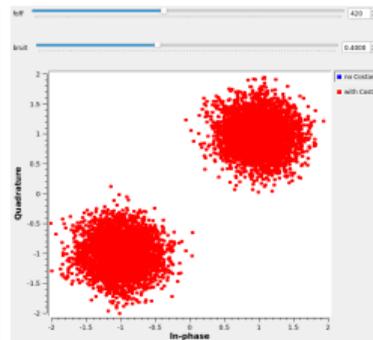
- ▶ I/Q coefficients represented in the **complex plane** (e.g. BPSK, QPSK ³⁷)
- ▶ AM = distance from origin, PM = angle with respect to X-axis
- ▶ Visual clue of symbol separation in a **noisy communication channel**
- ▶ Importance of **synchronizing** bit-generator (TX) and sampling (RX) clocks: rotation of the constellation if synchronization is not achieved



BPSK: no Costas



Costas



Costas+noise

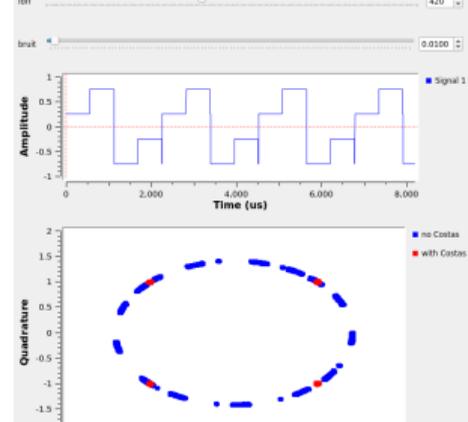
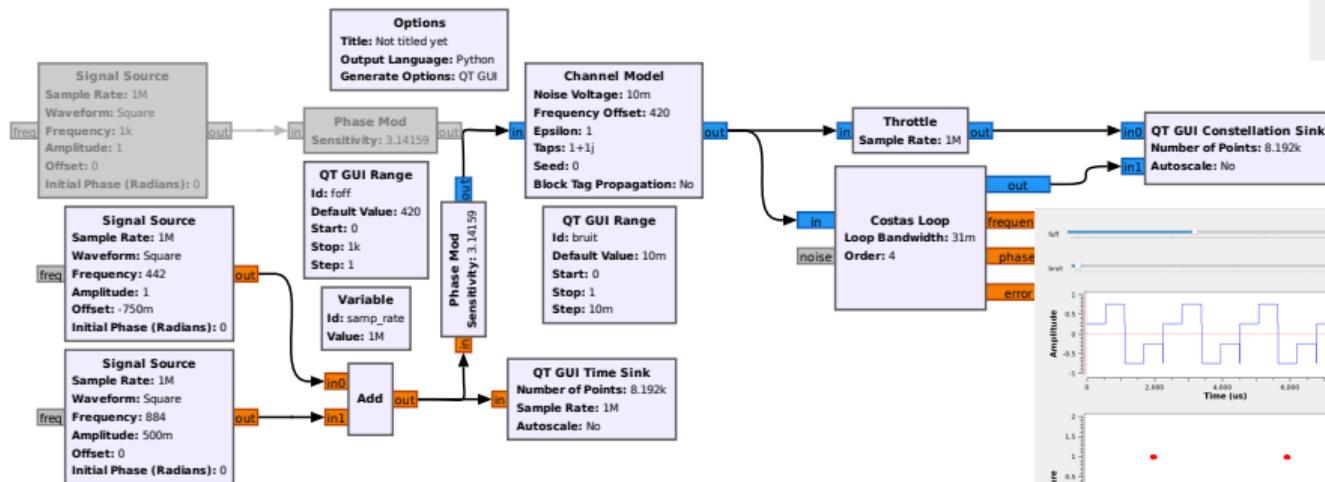
³⁸Balint's 4th tutorial on YouTube

https://www.youtube.com/watch?v=JMEyN_1vaiE &

demo: slides/constellation/constellation_analogique.grc

Constellation diagram

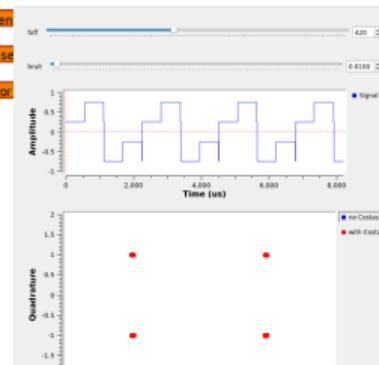
- ▶ I/Q coefficients represented in the **complex plane** (e.g. BPSK, QPSK ³⁹)
- ▶ AM = distance from origin, PM = angle with respect to X-axis
- ▶ Visual clue of symbol separation in a **noisy communication channel**
- ▶ Importance of **synchronizing** bit-generator (TX) and sampling (RX) clocks: rotation of the constellation if synchronization is not achieved



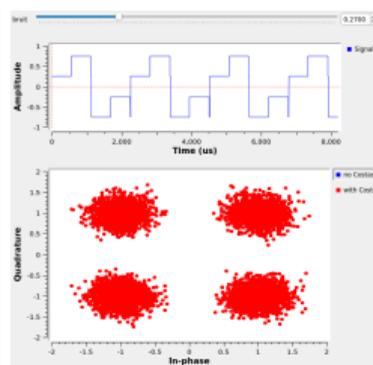
QPSK: no Costas

⁴⁰Balint's 4th tutorial on YouTube

https://www.youtube.com/watch?v=JMEyN_lvaiE



Costas



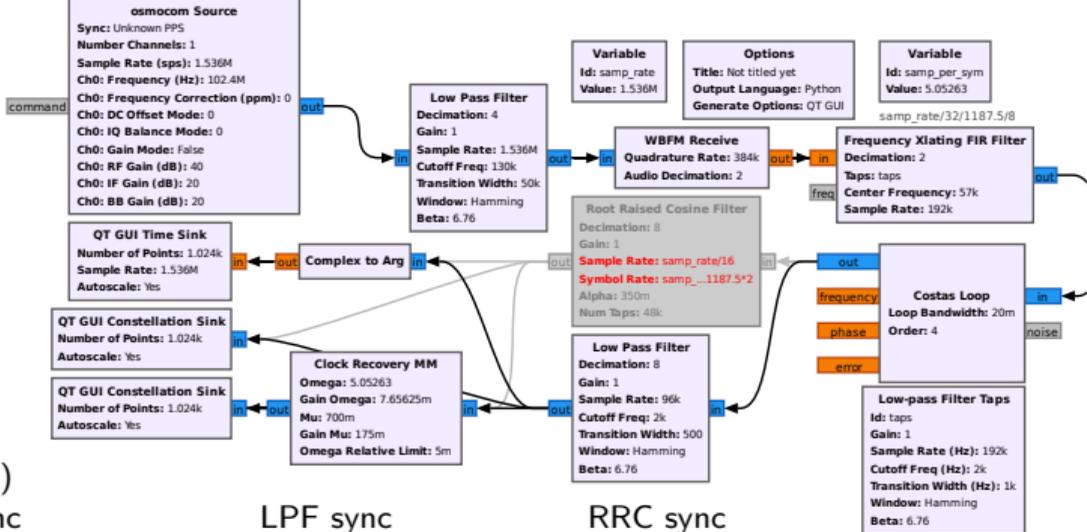
Costas+noise

Data stream synchro. (RDS)

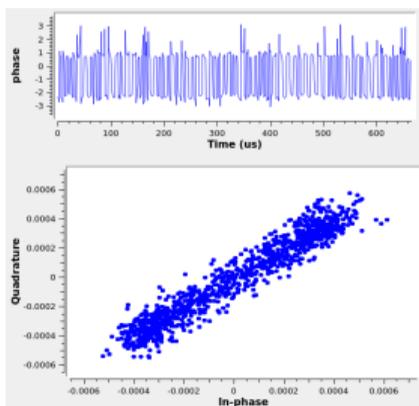
Practical demo on Radio Campus, 102.4 MHz filter digital stream

RDS: 57 kHz sub-carrier, 1187.5 bps, differential Manchester BPSK.

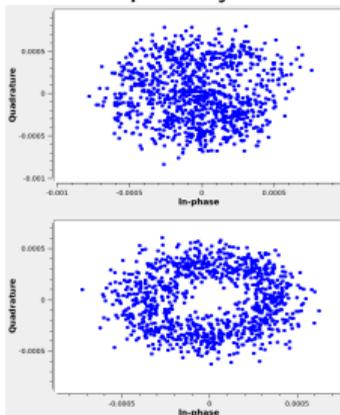
1. carrier recovery (Costas)
 2. clock synchronization (filter + MM)
 3. soft to hard bit threshold selection
 4. digital bitstream decoding (T. Lavarenne)
- Python script at jmfriedt.free.fr/rds.py



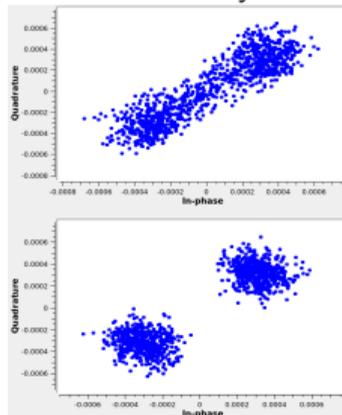
No sync



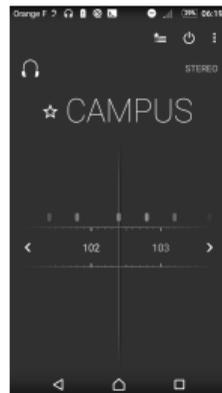
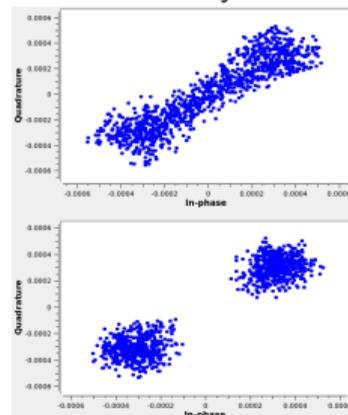
poor sync



LPF sync



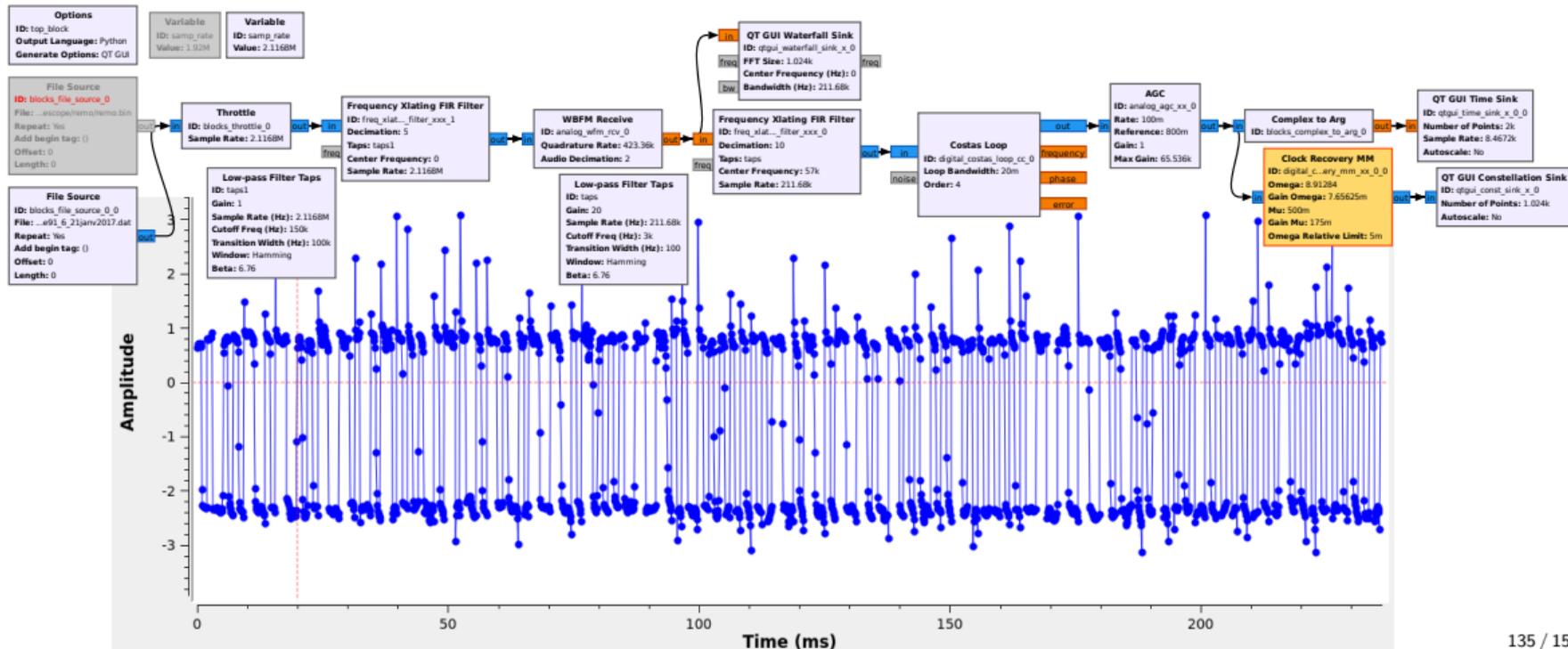
RRC sync



Access code for RDS: header in the continuous stream

- ▶ BPSK so two possible states (0 or 1): correlation is $+1$ or -1 but $abs(xcorr) = +1$ always
- ▶ Since $atan(Q/I) = \{0; \pi\}[2\pi]$ can jump between π and $-\pi$, use $Re(\varphi = 0) = 1$ and $Re(\varphi = \pi = -\pi) = -1$ while $Im(\varphi = \{0, \pi\}) = 0$

Question: in a continuous datastream (phases), how to identify the **beginning of sentences** ?



RDS access code: stream header

2.4 Synchronisation of blocks and groups

The blocks within each group are identified by the offset words A, B, C or C' and D added to blocks 1, 2, 3, and 4 respectively in each group (see annex A).

The beginnings and ends of the data blocks may be recognized in the receiver decoder by using the fact that the error-checking decoder will, with a high level of confidence, detect block synchronisation slip as well as additive errors. This system of block synchronisation is made reliable by the addition of the offset words (which also serve to identify the blocks within the group). These offset words destroy the cyclic property of the basic code so that in the modified code, cyclic shifts of codewords do not give rise to other codewords [6, 7].

Further explanation of a technique for extracting the block synchronisation information at the receiver is given in annex C.

Page 66
EN 50067:1998

ANNEX C (informative)

Implementation of group and block synchronisation using the modified shortened cyclic code

C.1 Theory

C.1.1 Acquisition of group and block synchronisation

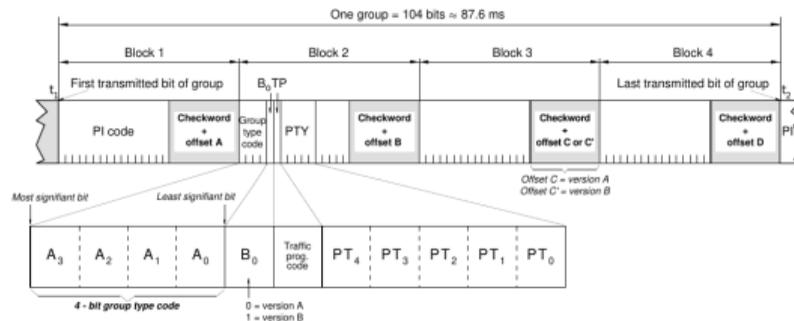
To acquire group and block synchronisation at the receiver (for example when the receiver is first switched on, or on tuning to a new station, or after a prolonged signal-fade) the syndrome \bar{s} must be calculated for each received 26-bit sequence. That is, on every data-clock pulse the syndrome of the currently stored 26-bit sequence (with the most recently received data bit at one end and the bit received 26 clock pulses ago at the other) is calculated on every clock pulse.

This bit-by-bit check is done continuously until two syndromes corresponding to valid offset words, and in a valid sequence for a group i.e. [A, B, C (or C'), D] are found $n \times 26$ bits apart (where $n = 1, 2, 3$, etc.). When this is achieved, the decoder is synchronised and the offset words which are added to the parity bits at the transmitter are subtracted at the receiver before the syndrome calculation for error correction/ detection is done (see annex B).

← Documented solution⁴¹: use the error correction code on all successive bits received until the consistent CRC is reached ...

↓ ... but all sentences start with the PI code unique⁴² to each radio station (convert to Manchester):

Page 13
EN 50067:1998



Notes to figure 9:

1. Group type code = 4 bits (see 3.1)
2. B₀ = version code = 1 bit (see 3.1)
3. PI code = Programme Identification code = 16 bits (see 3.2.1.1 and annex D)
4. TP = Traffic Programme Identification code = 1 bit (see 3.2.1.3)
5. PTY = Programme Type code = 5 bits (see 3.2.1.2 and annex F)
6. Checkword + offset "N" = 10 bits added to provide error protection and block and group synchronization

⁴² Specification of the radio data system (RDS) for VHF/FM sound broadcasting in the frequency range from 87.5 to 108.0 MHz, European Standard EN 50067, April 1998

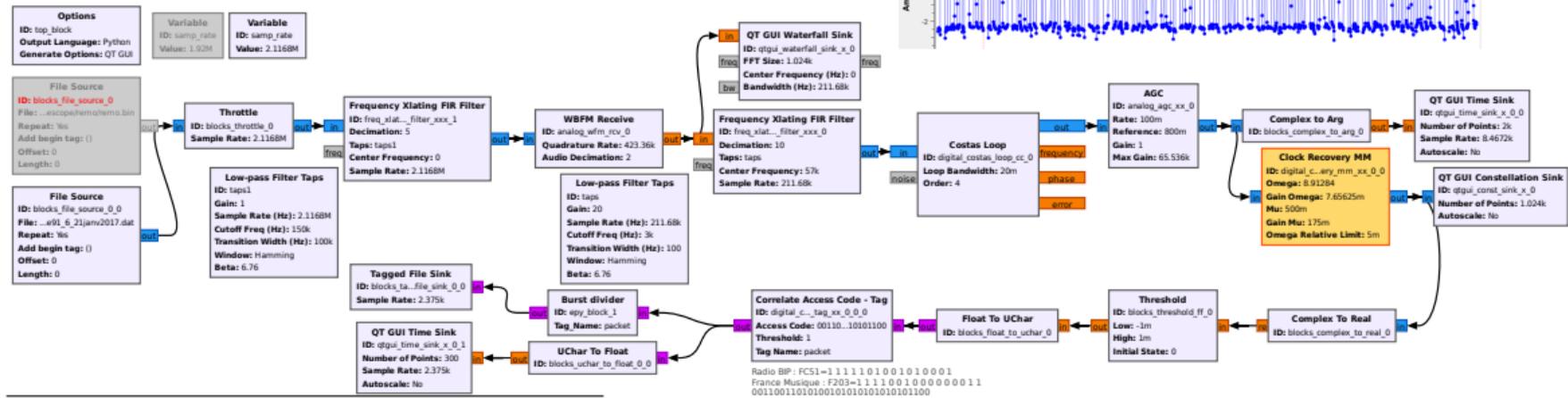
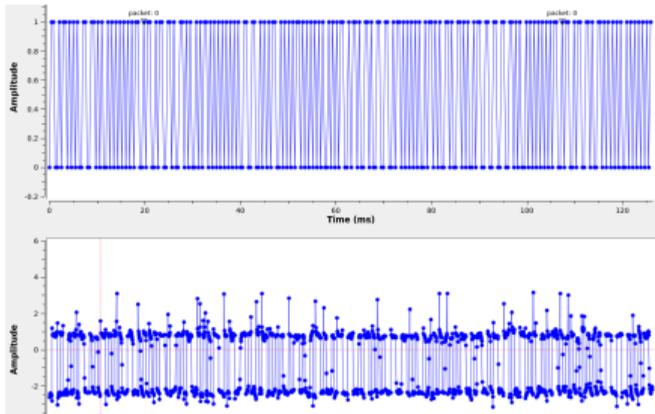
⁴²List of Access Codes: https://www.csa.fr/maradiofm/radiords_tableau, 0xFC3A for Radio Campus

RDS access code: stream header

PI is documented as “Program Identification”: differential Manchester encoding to maximize transitions (0 → 00 or 11, and 1 → 01 or 10) so that searching (**correlation**) this encoded pattern in the bit sequence must lead to the sentence beginning ⁴³

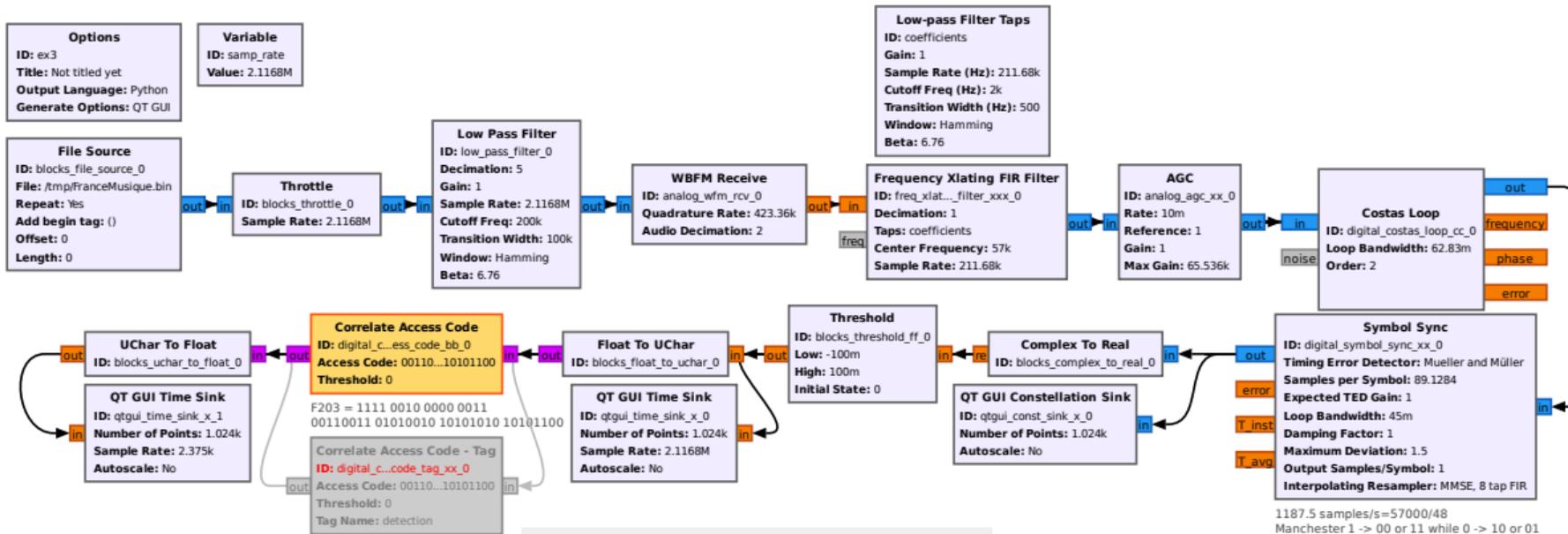


Radio Bip	FC51	BIP 96.9	Dijon	03/10/2011
Radio Blackbox	FD45	BLACKBOX	Bordeaux	04/03/2008
Radio Bleue	FE34	BLEUE	Polynésie	04/03/2008



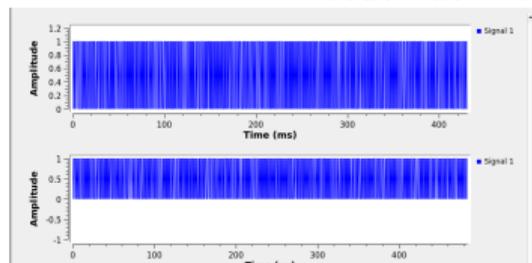
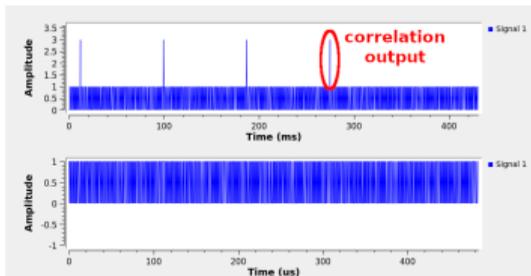
GNU Radio ≥ 3.9 : Symbol Sync block

Reference sequence must be properly encoded (convolutional, differential Manchester ...)



PI=F203 = 1111 0010 0000
0011 becomes 00110011 01010010
10101010 10101100

Left right sequence, right wrong sequence



1187.5 samples/s=57000/48
Manchester 1 -> 00 or 11 while 0 -> 10 or 01

Sentinel1 (CCSDS)



Sentinel-1 SAR Space Packet Protocol Data Unit

Doc. No.: S1-IF-ASD-PL-0007
Issue: 13
Date: 22.06.2015
Page 17 of 85

Octet Offset	bit							
	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
12	Sync Marker							
13								
14								
15								
16	Data Take ID							
17								
18								
19								
20	ECC Number							
21	n/a	Test Mode			Rx Channel ID			
22	Instrument Configuration ID							
23								
24								
25								

Table 3.2-3: Fixed Auxiliary Data Service Field

- ▶ Sentinel1: spaceborne RADAR launched by the European Space Agency (ESA) whose raw data are available at <https://scihub.copernicus.eu/dhus/>
- ▶ Each 25 second long data sequence file is 512 MB I,Q datastream

Question: how to find the beginning of sentences in the byte stream of the downloaded file?

3.2.2.1 Sync Marker

Description:	The Sync Marker represents a bit pattern to support (re-)synchronisation of packet data on Space Packet layer level (e.g. in case of corruptions or disruptions in a continuous stream of Space Packets)				
Performance:	constant value during the mission				
Short Name:	SYNC				
Code Name:	SYNC _{code}				
Code Properties	Start Position:	End Position:	Size of Code	Data Type	Applicable Range of Code:
	Octet 12, bit 0	Octet 15, bit 7	32 bit	unsigned int.	one static bit pattern
Interpretation:	SYNC = 352E F853 _{HEX}				

Search for synchronization word:

```
jeFriedt@rugged:/tmp$ xxd s1b-1u-raw-s-vv-20210216t083028-20210216t083100-025629-030def.dat | head -5
00000000: 0c1c eb95 4b15 4d56 4536 6e01 352e f853  ....K.MVE6n.5..S
00000010: 061b de80 0800 0000 0001 11c5 d300 03ab  ....E0/..
00000020: 9500 03b6 b80c 1f00 0808 8645 302f 0007  ...Uc...6JlA...
00000030: af09 0055 6300 11a8 0036 4a7c 6181 0600  ..J....(...d8.0+
00000040: 0a2e 5d00 03dc 0528 bc9a b264 3817 302b  ..J....(...d8.0+
```

Sentinel1 (CCSDS)

3.1 Packet Primary Header

The Packet Primary Header format is shown in Table 3.1-1 with the parameters described in Table 3.1-2.

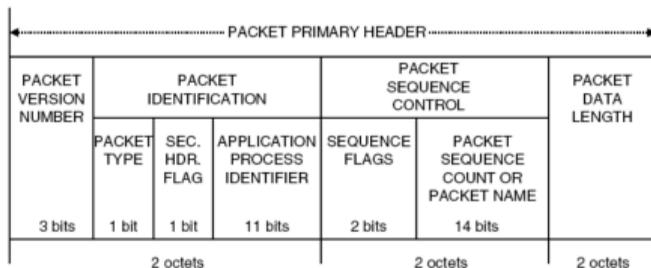


Table 3.1-1: Format of Packet Primary Header



Sentinel-1 SAR Space Packet Protocol Data Unit

Doc. No.: S1-IF-ASD-PL-0007
 Issue: 13
 Date: 22.06.2015
 Page: 14 of 85

Parameter	Length	Value	Comment	
Packet Version Number	3 bits	000 _{BIN} ¹		
Packet Type	1 bit	0 _{BIN}		
Secondary Header Flag	1 bit	1 _{BIN}	Secondary Header is present	
Application Process Identifier	PID	7 bis	100 0001 _{BIN} (65 _{DEC})	Process ID
	PCAT	4 bits	1100 _{BIN} (12 _{DEC})	Packet Category
Sequence Flags	2 bits	11 _{BIN}	user data are unsegmented	
Packet Sequence Count	14 bits	actual count of space packet (modulo 16384)	<ul style="list-style-type: none"> starts with "0" at start of measurement counts all packets output by the Instrument to the platform is an ambiguous count, that wraps to "0" after "16383" 	
Packet Data Length	16 bits	61 to 65533	number of octets in packet data field -1	

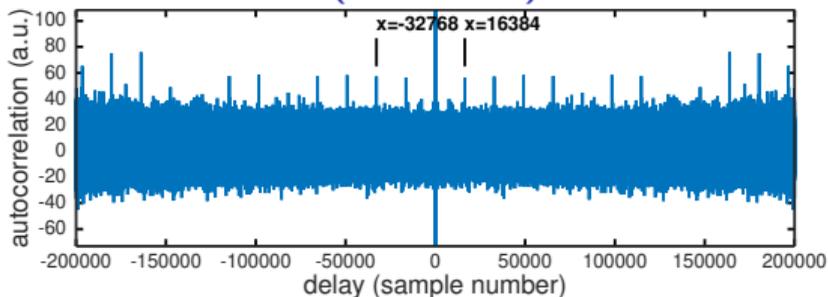
Once the synchronization word has been identified, the consistency of the header and data payload in each packet is verified:

$$000\ 0\ 1\ 100\ 0001\ 11 = 0000\ 1100\ 0001\ 11 = 0C1C$$

Pages refer to "Sentinel1 SAR Space Packet Protocol Data Unit S1-IF-ASD-PL-0007"

The packet counter is indeed incremented from one packet to the next: consistency of the decoding process

Meteor M2N (CCSDS): QPSK encoding



2.5.2.1 Bit Domain Frame Synchronization

The Consultative Committee for Space Data Systems has adopted the 32-bit ASM shown in Figure 4 for synchronization in the bit domain. The pattern is represented in hexadecimal as 1ACFFC1D but any pattern having a length of 8 to 64 bits such as the Inter-range Instrumentation Group (IRIG) patterns can be accommodated.

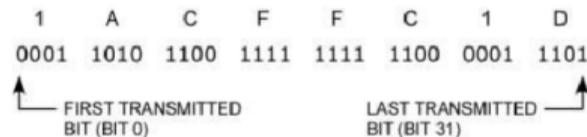
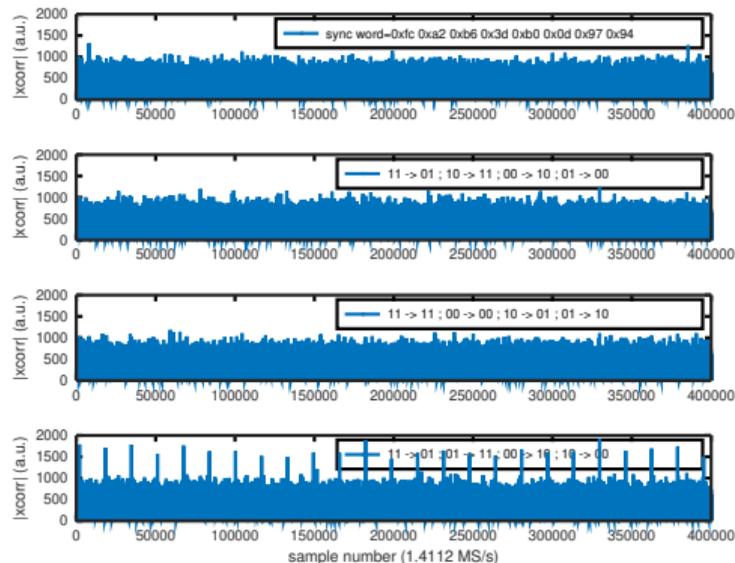


Figure 4. CCSDS Recommended 32-bit Attached Synchronization Marker



QPSK encodes two bits in each transmitted symbol ($0^\circ = 00$, $90^\circ = 01$, $180^\circ = 11$, $270^\circ = 10$)

Convolutional *Forward Error Correction* – FEC: the synchronization word must be encoded accordingly to be found in the received bit sequence ⁴⁴

⁴⁴J.-M Friedt, *Décodage d'images numériques issues de satellites météorologiques en orbite basse : le protocole LRPT de Meteor-M2*, GNU/Linux Magazine France 226 (1/3 Mai 2019), (2/3 Juin 2019)

General approach to digital mode demodulation

```
bin>65520 acm=140  
bin>12018.220753 bin>0092  
0a 16 01 32 2a 48 02 2f 4a 5a 54 15 48 32 2f 02 04 31 30 04 44 53 33 39 41 5a 30 47 20 20 20 20 64 34 37 33 30 37 45  
30 36 30 36 37 32 38 35 34 44 33 38 30 32 34 30 34 34 39 47 20 20 20 20 4e 34 37 33 31 37 45 30 36 30 30 31 32 39  
38 36 44 5a 31 30 32 34 30 30 39 47 20 20 20 20 4a 54 48 4e 4e 4e 3a 2b  
>>> Date
```

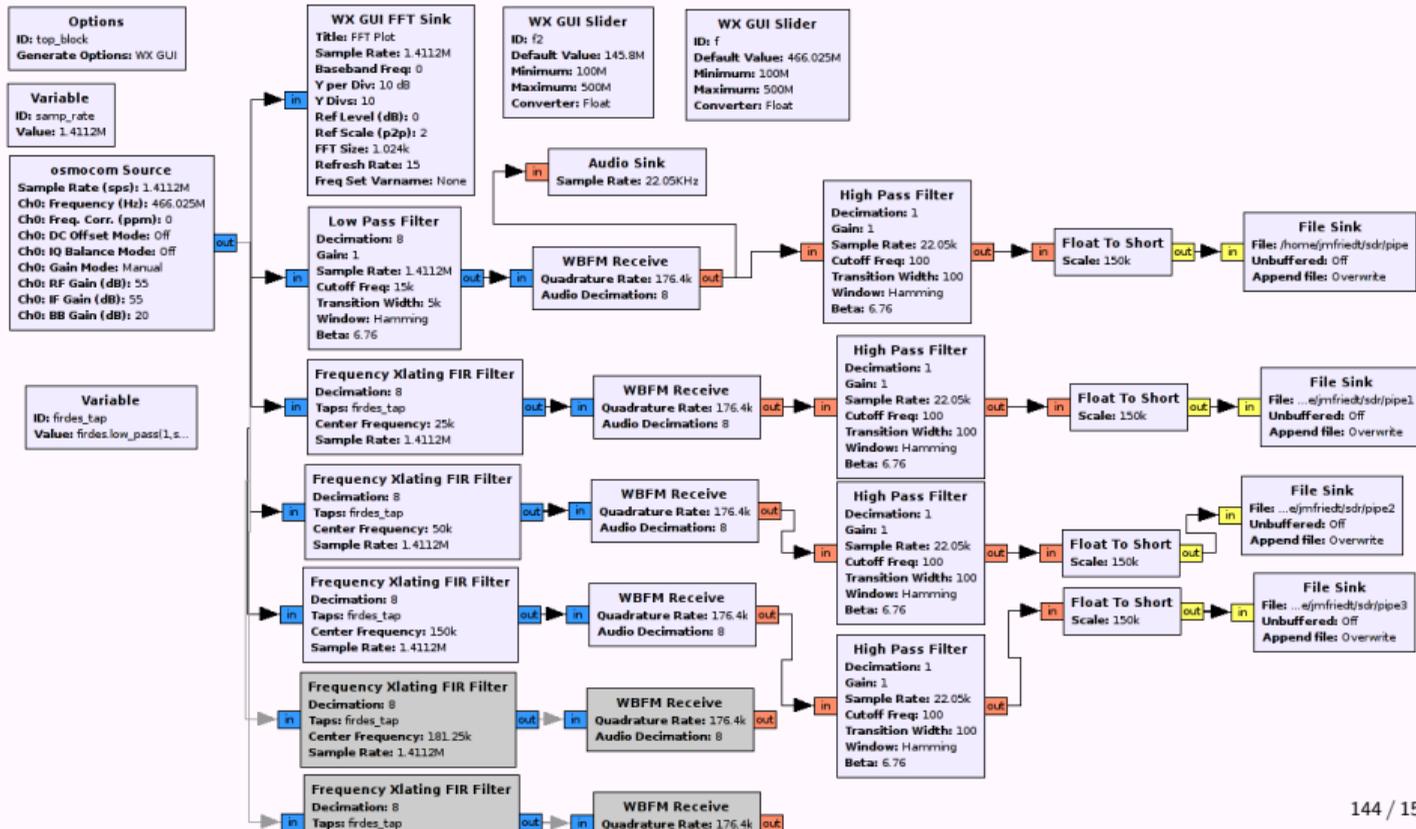
```
bin>65520 acm=140  
bin>12018.220753 bin>0092  
0a 16 01 32 2a 48 02 2f 4a 5a 54 15 48 32 2f 02 04 31 30 04 44 53 33 39 41 5a 30 47 20 20 20 20 64 34 37 33 30 37 45  
30 36 30 36 37 32 38 35 34 44 33 38 30 32 34 30 34 34 39 47 20 20 20 20 4e 34 37 33 31 37 45 30 36 30 30 31 32 39  
38 36 44 5a 31 30 32 34 30 30 39 47 20 20 20 20 4a 54 48 4e 4e 4e 3a 2b  
>>> Date
```

Beyond using available processing blocks: use GNURadio's basic blocks (demodulation) and add custom functionalities (**assuming** external tools are not already available):

1. record demodulated as (binary audio) baseband file
2. prototype signal processing algorithms in an interpreted language (GNU/Octave, Python/NumPy)
3. convert functional algorithms to C(++): **accumulate** enough samples and process the usable fraction (rotating FIFO buffer)
4. comply with gnuradio-companion architecture and benefit from GNU Radio scheduler

Hardware → software

- ▶ VCO → NCO ($\sin(2 \cdot \pi \cdot f \cdot t) \rightarrow \sin(\varphi)$ where $\varphi = 2 \cdot \pi \cdot f \cdot t[2\pi]$)
- ▶ convolution: $\sum x(k) \cdot pattern(n - k) = iFT(FT(x) \cdot FT(pattern))$
- ▶ mixer → \times
- ▶ filter → FIR
- ▶ rectifier → $abs()$

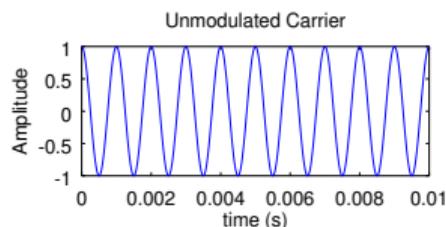
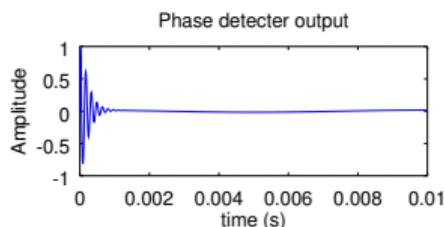
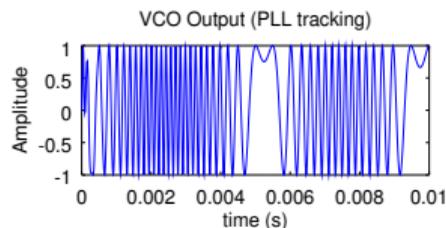
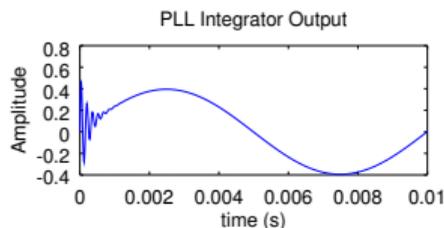
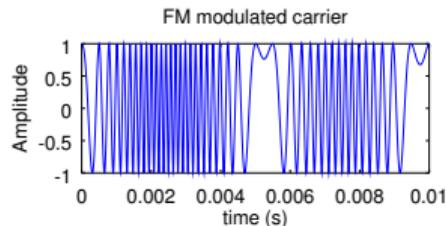
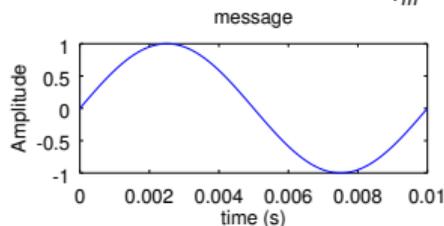


SDR:

- ▶ stable,
- ▶ flexible,
- ▶ reconfigurable

FM demodulation (PLL): software implementation

- ▶ $y(t) = \sin\left(2\pi \cdot f_c \cdot t + 2\pi f_\Delta \int_0^t x(\tau) d\tau\right)$
- ▶ if $x(\tau) = \cos(2\pi \cdot f_m t)$: $y(t) = \sin\left(2\pi \cdot f_c \cdot t + \frac{f_\Delta}{f_m} \cos(2\pi f_m t)\right)$
- ▶ modulation index $kf = \frac{f_\Delta}{f_m}$: $kf \ll 1 \Rightarrow$ NFM, $kf > 1 \Rightarrow$ WBFM



Software implementation of PLL^a

```
f=1000; % Carrier frequency
fs=100000; % Sample frequency
N=1001; % Number of samples
t=[0:1/fs:(N/fs-1/fs)];

%Create the message signal
f1=100; % Modulating frequency
msg=sin(2*pi*f1*t);

kf=.0628; % Modulation index
S=exp(j*(2*pi*f*t+2*pi*kf*cumsum(msg)));
C=exp(j*(2*pi*f*t)); % Unmodulated carrier

kp=0.15; ki=0.1; % Loop P & I constants
phi(1)=30; e(1)=0; phd(1)=0; vco(1)=0;

for n=2:length(S) % PLL implementation
    vco(n)=conj(exp(j*(2*pi*n*f/fs+phi(n-1))));
    phd(n)=imag(S(n)*vco(n)); % VCO x S input
    e(n)=e(n-1)+(kp+ki)*phd(n)-ki*phd(n-1); % PI
    phi(n)=phi(n-1)+e(n); % Update VCO
end;
```

^a http://fr.mathworks.com/matlabcentral/fileexchange/24167-simple-pll-demostration/content/simple_PLL.m

FM demodulation: SDR specific implementation without PLL

Baseband (RF mixer output) I/Q signal is

$$X(t) = A \cdot \exp(j\Delta\omega \cdot t + j\varphi(t)) \text{ with } \varphi(t) = D \int_{-\infty}^t m(\tau) d\tau$$

then

$$X_n \cdot X_{n+1}^* = A \cdot \exp\left(j\Delta\omega \cdot T_s + jD \int_{n \cdot T_s}^{(n+1) \cdot T_s} m(\tau) d\tau\right)$$

and

$$\int_{n \cdot T_s}^{(n+1) \cdot T_s} m(\tau) d\tau \simeq T_s \cdot m(n \cdot T_s) \text{ (rectangle)}$$

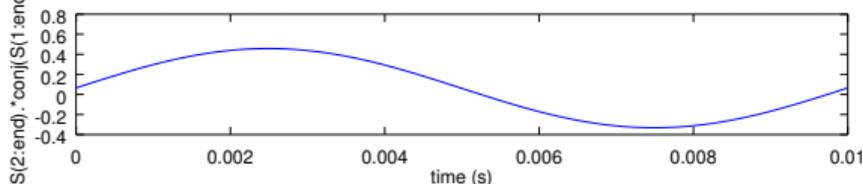
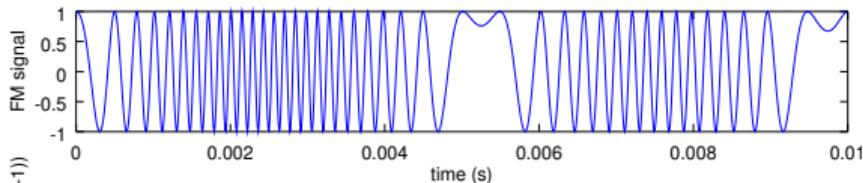
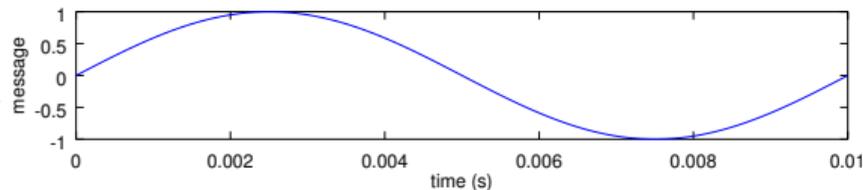
so that

$$\arg(X_n \cdot X_{n+1}^*) = \underbrace{\Delta\omega \cdot T_s}_{\text{offset}} + \underbrace{D \cdot T_s}_{\text{gain}} \cdot m(n \cdot T_s)$$

which is multiplied with $\frac{1}{T_s}$

This is the *quadrature FM demodulator* implementation of GNURadio ^a

^aD. Bederov, *Arithmetic based implementation of a quadrature FM Demodulator*, at https://fosdem.org/2015/schedule/event/sdr_arithmetic/



```
f=1000; fs=100000; N=1001;
t=[0:1/fs:(N/fs-1/fs)];
f1=100; % Modulating frequency
msg=sin(2*pi*f1*t);
kf=.0628; % Modulation index
S=exp(j*(2*pi*f*t+2*pi*kf*cumsum(msg)));
plot(angle(S(2:end)).*conj(S(1:end-1)))
```

- ▶ FM demodulation is achieved with a single line of code !
- ▶ Issue with computing the angle function which requires **trigonometric functions**

$$\leftarrow \Rightarrow \frac{dX_n}{dt} \cdot X_n^* = jA^2 (2\pi\Delta f + D \cdot m(nT_s)) .$$

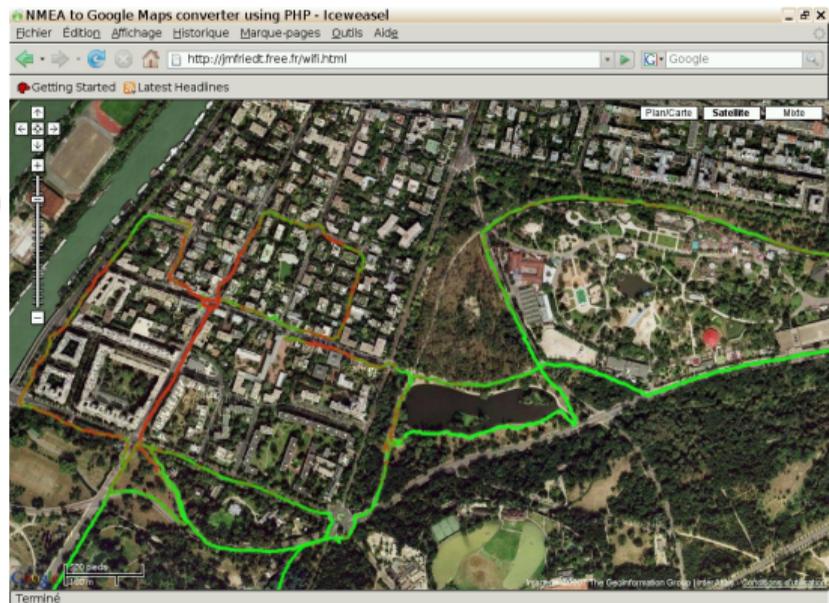
Limitations of the software approach

- ▶ limitation on the sampling rate (measurement bandwidth)
- ▶ sound card ≤ 192 kS/s (16 bits), DVB-T ≤ 2.4 MS/s (8 bits), fast ADC (250–310 MS/s)
- ▶ processing bandwidth (RAM to CPU transfer, processing by the CPU)
- ▶ measurement dynamic range (8 bits=48 dB, 10 bits=60 dB, 16 bits=96 dB): a signal below the quantization resolution cannot be retrieved during post-processing

but once these limitations are identified, flexibility in prototyping and implementing radiofrequency processing algorithms.

Radiofrequency communication

- ▶ high power consumption \Rightarrow consider power saving strategies \Rightarrow data storage and compression
- ▶ lower reliability than wired links
- ▶ various technologies depending on the objectives: radiomodems, wifi, bluetooth/zigbee, LoRa
- ▶ selecting the right technology depends on the purpose. Consider redundancy (local storage).
- ▶ variable performances whether the system is static or mobile
- ▶ the lower the carrier frequency, the longer the communication range but the larger the antenna
- ▶ security ? wireless removes the OSI layer 0 (hardware security)



Emission regulations

CC1101

Low-Power Sub-1 GHz RF Transceiver

Applications

- Ultra low-power wireless applications operating in the 315/433/868/915 MHz ISM/SRD bands
- Wireless alarm and security systems
- Industrial monitoring and control
- Wireless sensor networks
- AMR – Automatic Meter Reading
- Home and building automation
- Wireless MBUS

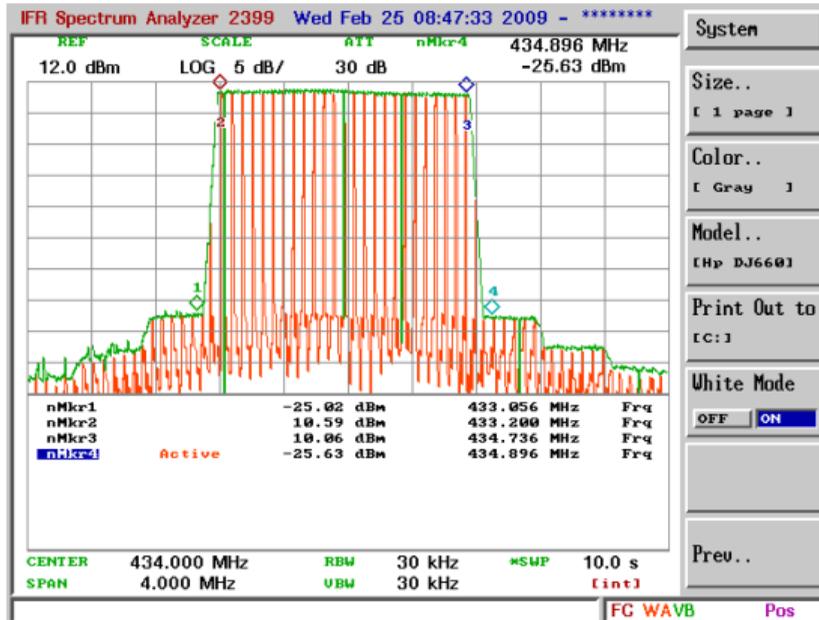
Product Description

CC1101 is a low-cost sub-1 GHz transceiver designed for very low-power wireless applications. The circuit is mainly intended for the ISM (Industrial, Scientific and Medical) and SRD (Short Range Device) frequency bands at 315, 433, 868, and 915 MHz, but can easily be programmed for operation at other frequencies in the 300-348 MHz, 387-464 MHz

microcontroller and a few additional passive components.

The *CC1190* 850-950 MHz range extender [21] can be used with *CC1101* in long range applications for improved sensitivity and higher output power.

- ...
- Suited for systems targeting compliance with EN 300 220 (Europe) and FCC CFR Part 15 (US)
 - Suited for systems targeting compliance with the Wireless MBUS standard EN 13757-4:2005



Being able to emit at 915 MHz does not mean we are *allowed* to do so in Europe (for example).

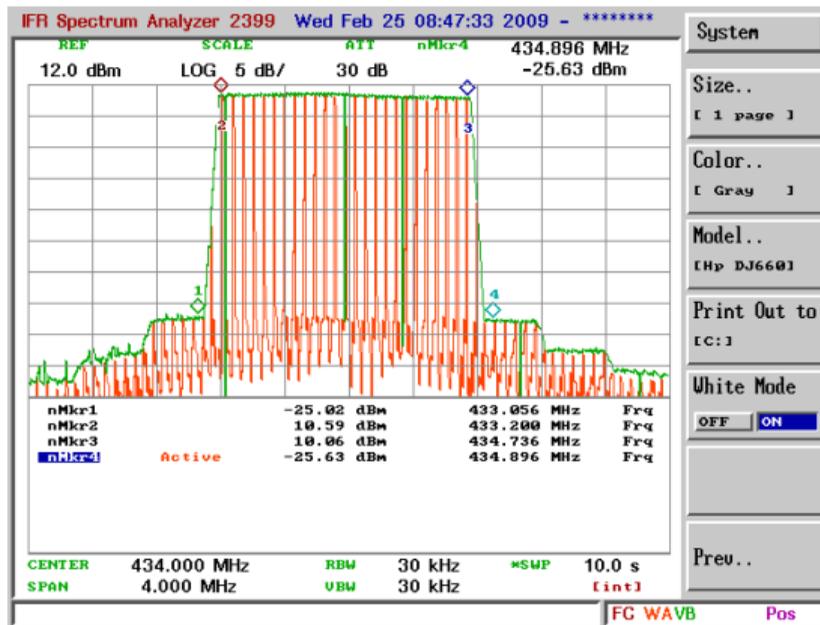
Emission regulations

Include

- ▶ the allocated frequency band (ISM: 13.56, 433.9 and 2440 MHz),
- ▶ **radiated** power,
- ▶ duty cycle, type of signal⁴⁷ ...

+

- ▶ sensitivity to supply voltage/noise (mains supply voltage noise)
- ▶ avoid disturbing other circuits (noise generated on the mains supply)
- ▶ avoid unwanted radiated signal (frequency band)
- ▶ be resistant to jamming signals (provide the expected functionality even if another emitter complying with regulations is emitting)
- ▶ ESD (4-8 kV) on case and connectors



Being *able* to emit at 915 MHz does not mean we are *allowed* to do so in Europe (for example).

⁴⁷ETSI EN 300 220, available at

Practical applications

- ▶ Getting familiar with GNURadio thanks to GNURadio Companion
- ▶ Commercial broadcast FM
 - ▶ 200 kHz-wide channels, 88-108 MHz carrier
- ▶ POCSAG reception
 - ▶ FSK \pm 4.5 kHz
 - ▶ 25 kHz-wide channels
 - ▶ 1200 or 2400 bauds
 - ▶ multiple channel decoding
- ▶ using an external tool for decoding sentences (`multimon-ng`)
- ▶ GPS decoding

Bibliography 1/2

1. T.P. Zieliński, *Starting Digital Signal Processing in Telecommunication Engineering, A Laboratory-based Course*, Springer (2020)
2. T. Collins & al., *Software-Defined Radio for Engineers*, (2018) at <https://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html>
3. S.W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing, 2nd Ed* (1999) at <https://www.dspguide.com/pdfbook.htm>
4. T. McDermott, *Wireless Digital Communications : Design and Theory*, Tucson Amateur Packet Radio Corporation – TAPR (1997)
5. J.G. Proakis, D.K. Manolakis, *Digital Signal Processing*, Prentice Hall (2006)
6. R.G. Lyons, *Understanding Digital Signal Processing*, Prentice Hall (2004)
7. A.V. Oppenheim, R.W. Schaffer, *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/
8. K. Borre, D.M. Akos, N. Bertelsen, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*, Birkhäuser (2007)
9. E.D. Kaplan, C. Hegarty, *Understanding GPS: Principles and Applications, 2nd Ed.*, Artech House (2005)
10. C.A. Balanis, *Antenna Theory, Analysis and Design*, Wiley Interscience (2005)

Bibliography 2/2

On the 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 02/2021⁴⁸)
5. *Learning DSP Illustrated* at <https://dspillustrations.com/pages/index.html> (accessed 02/2021)

⁴⁸presented at FOSDEM2021 at https://fosdem.org/2021/schedule/event/fsr_pysdr_guide_to_sdr_and_dsp_using_python/

POES satellites

<https://www.ospo.noaa.gov/Operations/POES/status.html>

SECTION 1: POLAR APT/LRPT REPORT { 18th January 2016.

Satellite Frequency (MHz) Status Image Quality

NOAA 15 137.620 (APT) on good
NOAA 18 137.9125 (APT) on good
NOAA 19 137.100 (APT) on good

2. NOAA 14 was decommissioned on 23rd May 2007.
3. NOAA 12 was decommissioned on 10th August 2007.
4. NOAA 17 was decommissioned on 10th April 2013.
5. NOAA 16 was decommissioned on 9th June 2014.

Pass prevision: wxtoimg, predict (Sattrack is obsolete and requires correcting a Y2K bug when compiled)

