Digital communication

J.-M Friedt

FEMTO-ST/time & frequency department

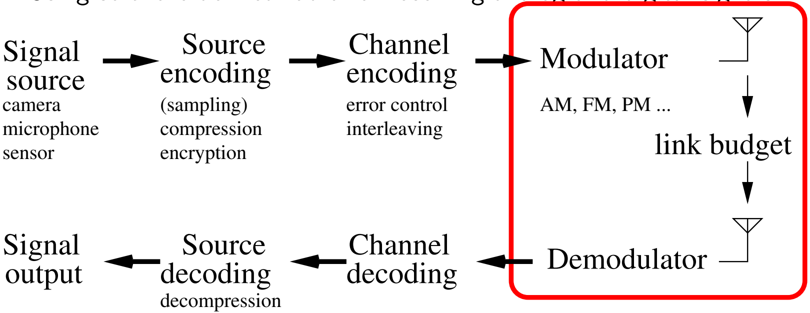
jmfriedt@femto-st.fr

slides and references at `jmfriedt.free.fr`

March 14, 2025

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



Signal source
camera
microphone
sensor

Source encoding
(sampling)
compression
encryption

Channel encoding
error control
interleaving

Modulator
AM, FM, PM ...

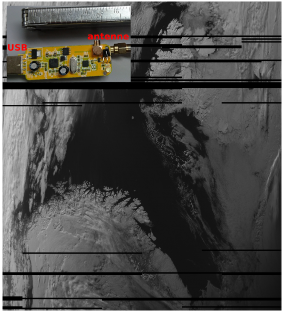
link budget

Demodulator

Channel decoding

Source decoding
decompression

Signal output



---

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

[2] J.-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

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

⇒ 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`

---

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

[2] J.-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-Hartley theorem)
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`:

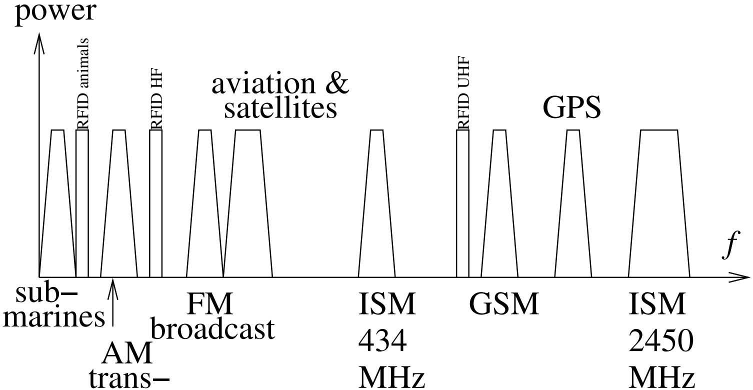
1. GNU Radio as a general purpose signal processing prototyping tool: application to AM and FM
2. PM demodulation & decoding multiple bands (analog FM and/or digital POCSAG)
3. CDMA/GPS signal characterization, correlations (GNU Radio & Octave)
4. antenna design using NEC2 and radiation pattern display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

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

Radiofreqeuncy link

- replace a wire with an electromagnetic wave
- $\lambda = c/f \Rightarrow \lambda[\mathrm{m}] = 300/f[\mathrm{MHz}]$
- 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) $\Rightarrow \lambda = 12.5$ cm
- various carriers to multiplex the radiofrequency spectrum and transmit various information in different spectral bands

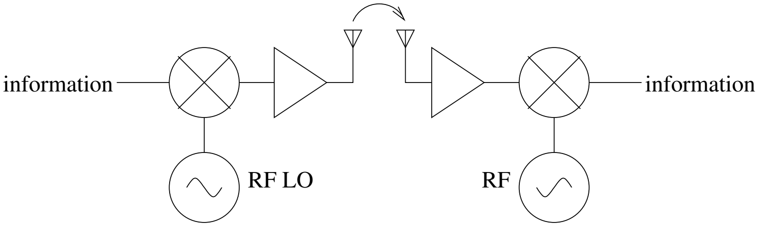
The carrier frequency is of little interest, **only the bandwidth** defines the amount of information that can be transmitted after transposition to baseband by the receiver.

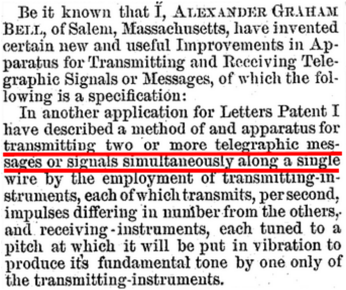


In the beginning ...

- 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:

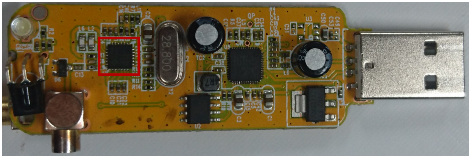
In another application for Letters Patent I have described a method of and apparatus for transmitting two or more telegraphic messages or signals simultaneously along a single wire by the employment of transmitting-instruments, each of which transmits, per second, impulses differing in number from the 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.

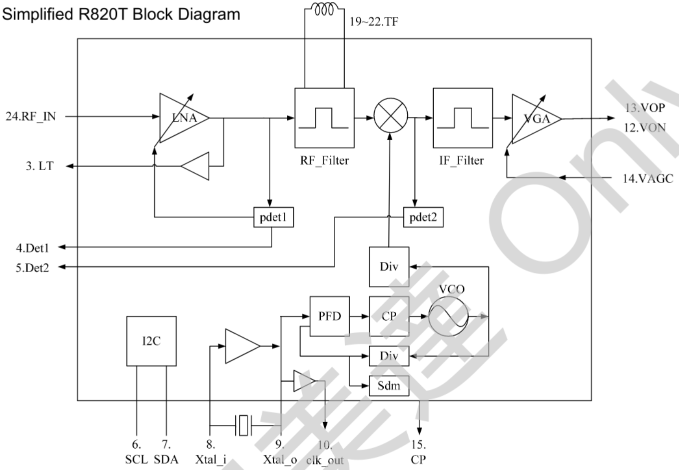
---

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

Mixer for frequency transposition

- Linear Time Invariant (LTI) system:

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

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

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

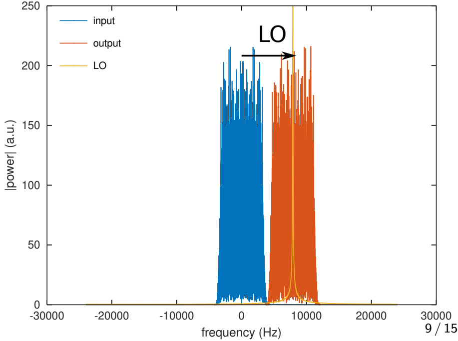
  or analyzing f each frequency at a time (Fourier analysis).

- 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:

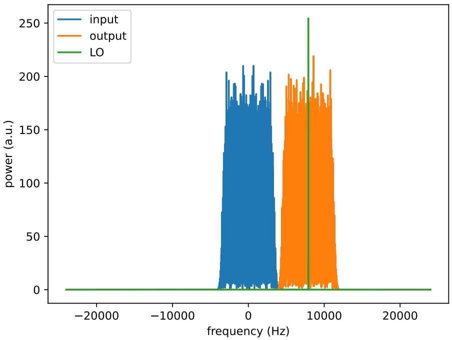
$$f\left(A\exp(j\omega_1 t)+B\exp(j\omega_2 t)\right)=Af(\exp(j\omega_1 t))+Bf(\exp(j\omega_2 t))$$

- **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)')
```



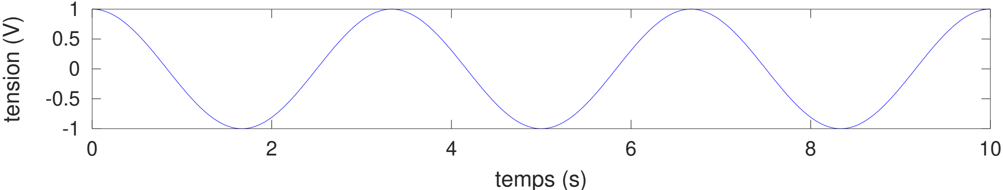
Discrete time signal processing

Discrete levels ⇒ limited dynamic range   Discrete time ⇒ 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 ⇒ $f_s/2$ is defined (Nyquist frequency) as 1 and express all frequencies f as normalized frequency

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

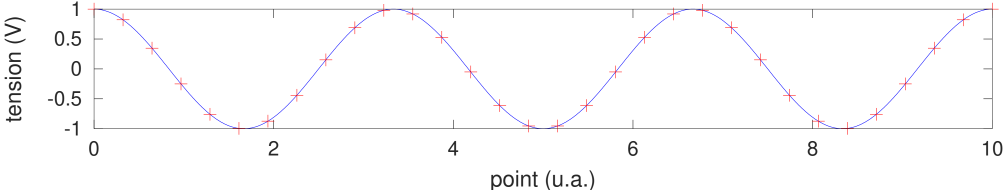
  - sampling N points ⇒ discrete Frequency transform on N samples: spectral resolution f_s/N



Discrete time signal processing

Discrete levels ⇒ limited dynamic range Discrete time ⇒ 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 ⇒ $f_s/2$ 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 ⇒ discrete Frequency transform on N samples: spectral resolution f_s/N



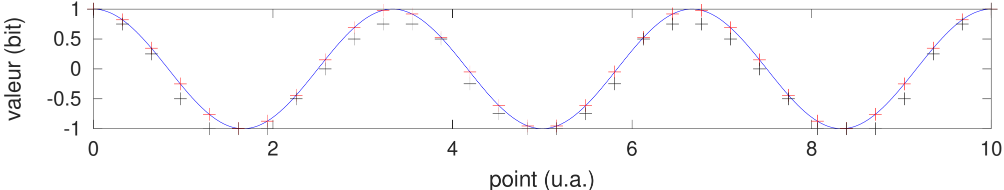
Discrete time signal processing

Discrete levels ⇒ limited dynamic range Discrete time ⇒ 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 ⇒ $f_s/2$ 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 ⇒ discrete Frequency transform on N samples: spectral resolution f_s/N



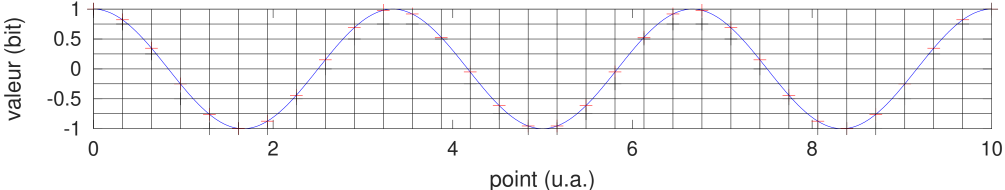
Discrete time signal processing

Discrete levels ⇒ limited dynamic range   Discrete time ⇒ 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 ⇒ $f_s/2$ is defined (Nyquist frequency) as 1 and express all frequencies f as normalized frequency

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

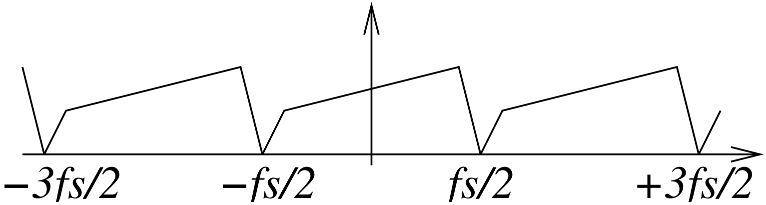
    - sampling N points ⇒ discrete Frequency transform on N samples: spectral resolution f_s/N



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);`

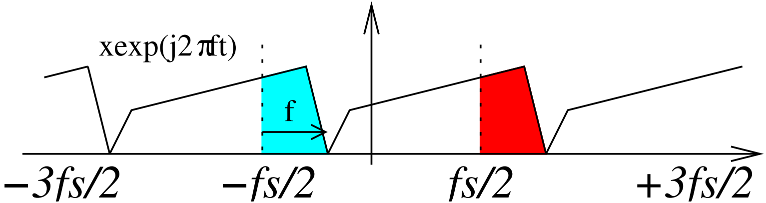


Remember: $\exp(j\omega_1 t) \cdot \exp(j\omega_2 t) = \exp(j(\omega_1 + \omega_2)t)$

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:

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

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

- **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:

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

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

- 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:

$$\underbrace{A_1\sin(\omega_1 t+\varphi)}_{\text{received signal}}\cdot\underbrace{A_2\cos(\omega_2 t)}_{\text{local reference}}\propto A_1A_2\left[\sin(\omega_1+\omega_2+\varphi)+\sin(\omega_1-\omega_2+\varphi)\right]$$

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

- 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^o as reference signal: cos → sin

$$A_1\sin(\omega_1 t+\varphi)\cdot A_2\sin(\omega_2 t)\propto A_1A_2\left[\cos(\omega_1+\omega_2+\varphi)+\cos(\omega_1-\omega_2+\varphi)\right]$$

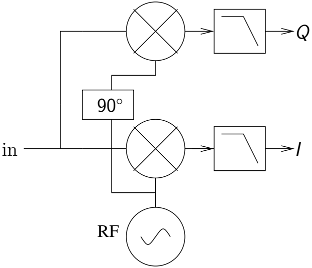
$$\propto A_1\cdot A_2\cdot\cos(\varphi)\text{ if }\omega_1=\omega_2$$

- **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:

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



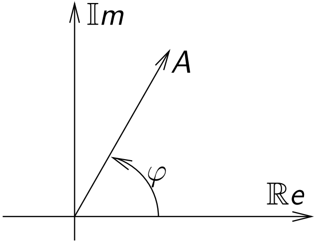
$$A_1 \sin(\omega_1 t + \varphi) \cdot A_2 \sin(\omega_2 t) \propto A_1 \cdot A_2 \cdot \cos(\varphi) \text{ if } \omega_1 = \omega_2$$

- 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

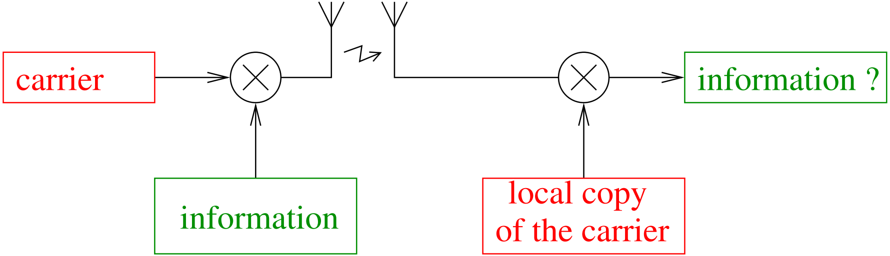
- $\exp(j \cdot \omega t) = \cos(\omega t) + j \cdot \sin(\omega t)$ where $j^2 = -1$
- trigonometric identities: $\cos(a) = \frac{1}{2}\left(\exp(j \cdot a) + \exp(-j \cdot a)\right)$ and $\sin(a) = \frac{1}{2j}\left(\exp(j \cdot a) - \exp(-j \cdot a)\right)$
- on the complex plane (*phasor diagram*),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



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

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)

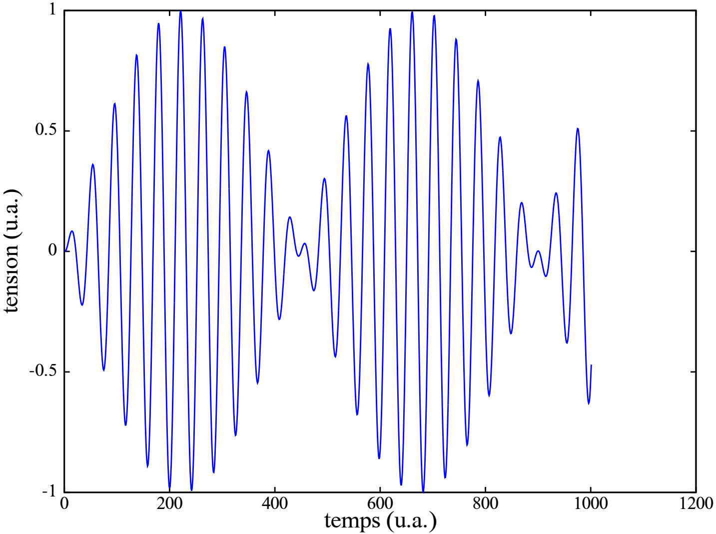


Non-linear processes (multiplication) are the only ones able to sum frequencies.

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]



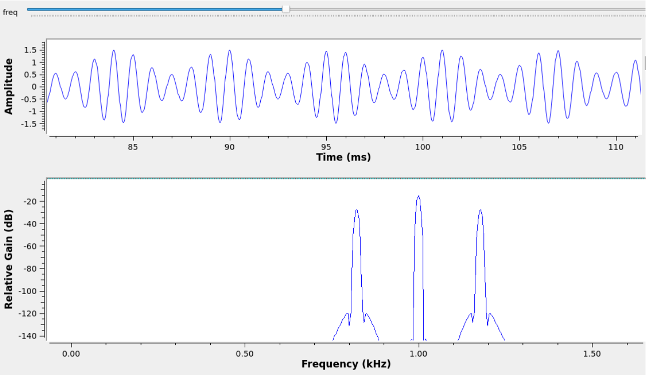
---

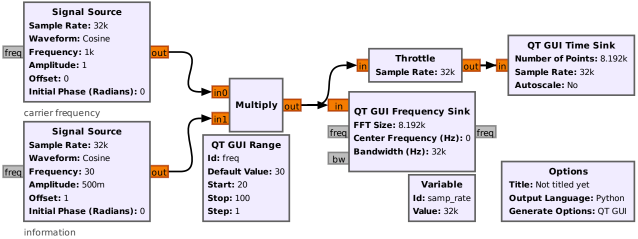
[a] T. Rondeau, *GNU Radio for Exploring Signals*, FOSDEM 2016, at `https://archive.fosdem.org/2016/schedule/event/gnu_radio/`

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]





Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 30
Amplitude: 500m
Offset: 1
Initial Phase (Radians): 0

Multiply

QT GUI Range
Id: freq
Default Value: 30
Start: 20
Stop: 100
Step: 1

Throttle
Sample Rate: 32k

QT GUI Time Sink
Number of Points: 8.192k
Sample Rate: 32k
Autoscale: No

QT GUI Frequency Sink
FFT Size: 8.192k
Center Frequency (Hz): 0
Bandwidth (Hz): 32k

Variable
Id: samp_rate
Value: 32k

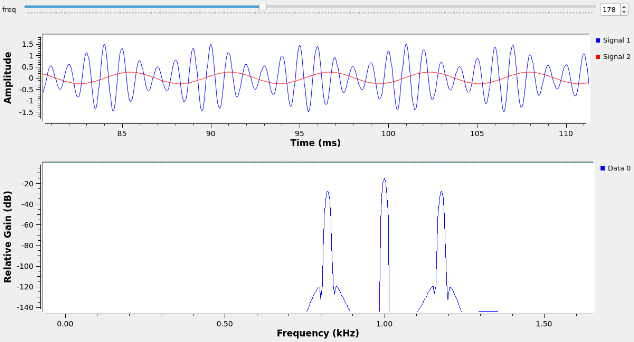
Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

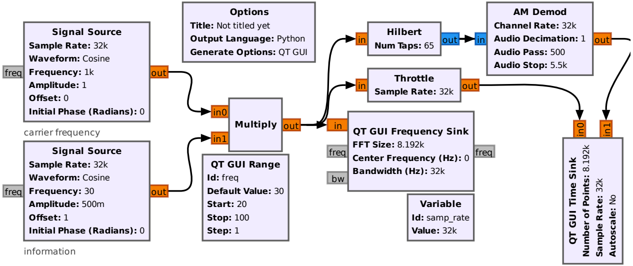
Throttle due to lack of hardware timing peripheral in the flowgraph

---

[a] T. Rondeau, *GNU Radio for Exploring Signals*, FOSDEM 2016, at `https://archive.fosdem.org/2016/schedule/event/gnu_radio/`

AM demodulation





Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

carrier frequency

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 30
Amplitude: 500m
Offset: 1
Initial Phase (Radians): 0

information

Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Multiply

QT GUI Range
Id: freq
Default Value: 30
Start: 20
Stop: 100
Step: 1

Hilbert
Num Taps: 65

Throttle
Sample Rate: 32k

QT GUI Frequency Sink
FFT Size: 8.192k
Center Frequency (Hz): 0
Bandwidth (Hz): 32k

Variable
Id: samp_rate
Value: 32k

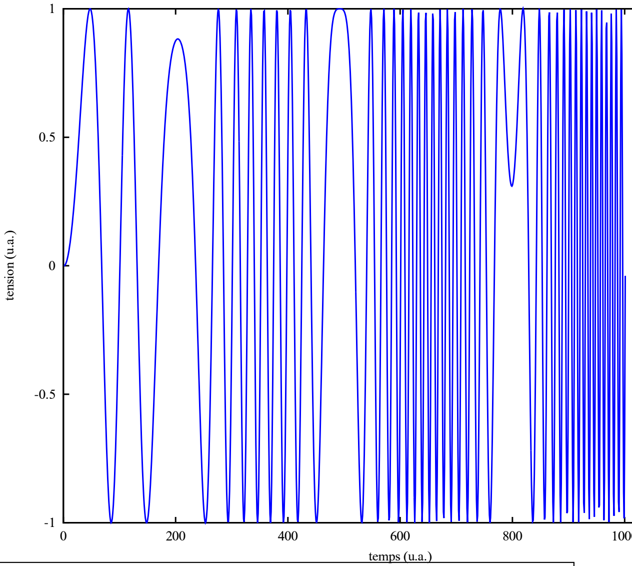
AM Demod
Channel Rate: 32k
Audio Decimation: 1
Audio Pass: 500
Audio Stop: 5.5k

QT GUI Time Sink
Number of Points: 8.192k
Sample Rate: 32k
Autoscale: No

- 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

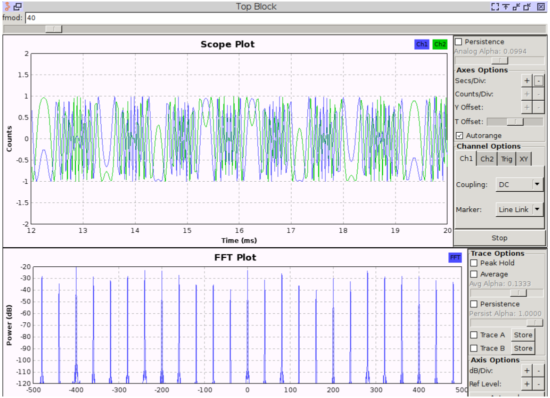
- information encoded on the *frequency*
- FSK (Frequency Shift Keying): two frequencies encode the two possible data states (0, 1)
- the output of the frequency demodulation is a voltage dependent on the carrier frequency offset
- $$y(t) = \sin\left(2\pi \cdot f_c \cdot t + 2\pi f_\Delta \underbrace{\int_0^t x(\tau)d\tau}_{phase \rightarrow freq}\right)$$
- 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

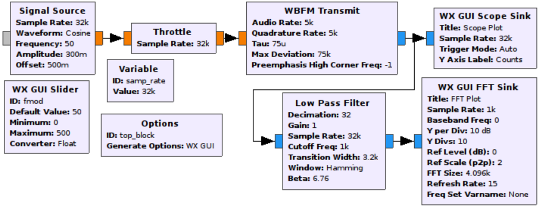


Carson: $B \simeq 2 \cdot (f_\Delta + f_m)$, f_Δ=deviation, f_m highest spectral component of the signal

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





Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 50
Amplitude: 300m
Offset: 500m

Throttle
Sample Rate: 32k

WBFM Transmit
Audio Rate: 5k
Quadrature Rate: 5k
Tau: 75u
Max Deviation: 75k
Preemphasis High Corner Freq: -1

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 32k
Trigger Mode: Auto
Y Axis Label: Counts

Variable
ID: samp_rate
Value: 32k

WX GUI Slider
ID: fmod
Default Value: 50
Minimum: 0
Maximum: 500
Converter: Float

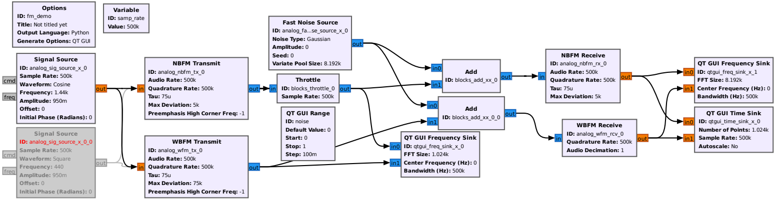
Options
ID: top_block
Generate Options: WX GUI

Low Pass Filter
Decimation: 32
Gain: 1
Sample Rate: 32k
Cutoff Freq: 1k
Transition Width: 3.2k
Window: Hamming
Beta: 6.76

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 1k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 4.096k
Refresh Rate: 15
Freq Set Varname: None

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

FM demodulation



Options
ID: fm_demo
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 500k

Signal Source
ID: analog_sig_source_x_0
Sample Rate: 500k
Waveform: Cosine
Frequency: 1.44k
Amplitude: 950m
Offset: 0
Initial Phase (Radians): 0

Signal Source
ID: analog_sig_source_x_0_0
Sample Rate: 500k
Waveform: Square
Frequency: 440
Amplitude: 950m
Offset: 0
Initial Phase (Radians): 0

NBFM Transmit
ID: analog_nbfm_tx_0
Audio Rate: 500k
Quadrature Rate: 500k
Tau: 75u
Max Deviation: 5k
Preemphasis High Corner Freq: -1

WBFM Transmit
ID: analog_wfm_tx_0
Audio Rate: 500k
Quadrature Rate: 500k
Tau: 75u
Max Deviation: 75k
Preemphasis High Corner Freq: -1

Fast Noise Source
ID: analog_fa...se_source_x_0
Noise Type: Gaussian
Amplitude: 0
Seed: 0
Variate Pool Size: 8.192k

Throttle
ID: blocks_throttle_0
Sample Rate: 500k

QT GUI Range
ID: noise
Default Value: 0
Start: 0
Stop: 1
Step: 100m

Add
ID: blocks_add_xx_0

Add
ID: blocks_add_xx_0_0

QT GUI Frequency Sink
ID: qtgui_freq_sink_x_0
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 500k

NBFM Receive
ID: analog_nbfm_rx_0
Audio Rate: 500k
Quadrature Rate: 500k
Tau: 75u
Max Deviation: 5k

WBFM Receive
ID: analog_wfm_rcv_0
Quadrature Rate: 500k
Audio Decimation: 1

QT GUI Frequency Sink
ID: qtgui_freq_sink_x_1
FFT Size: 8.192k
Center Frequency (Hz): 0
Bandwidth (Hz): 500k

QT GUI Time Sink
ID: qtgui_time_sink_x_0
Number of Points: 1.024k
Sample Rate: 500k
Autoscale: No

↑ **Observe the bandwidth occupied by WBFM and NFM**

- $f_\Delta \gg f_m$: 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: baseband I/Q signal is

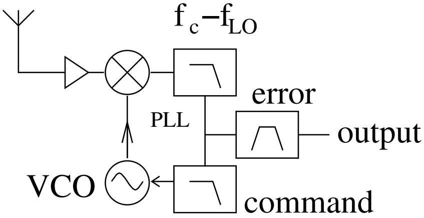
  $$X(t) = A\cdot\exp(j\Delta\omega\cdot t + j\varphi(t)) \text{ with } \varphi(t) = D\int_{-\infty}^{t} m(\tau)d\tau$$

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

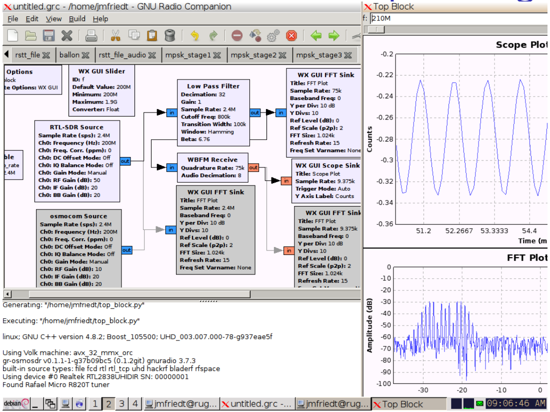
which is multiplied with $\frac{1}{T_s}$: *quadrature FM demodulator* implementation of GNURadio [5]



---

[5] D. Bederov, *Arithmetic based implementation of a quadrature FM Demodulator*, at `https://fosdem.org/2015/schedule/event/sdr_arithmetic/`

FM demodulation: example of digital modem



Options
te Options: WX GUI

WX GUI Slider
ID: f
Default Value: 200M
Minimum: 200M
Maximum: 1.9G
Converter: Float

RTL-SDR Source
Sample Rate (sps): 2.4M
Ch0: Frequency (Hz): 200M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 50
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

osmocom Source
Sample Rate (sps): 2.4M
Ch0: Frequency (Hz): 200M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Low Pass Filter
Decimation: 32
Gain: 1
Sample Rate: 2.4M
Cutoff Freq: 800k
Transition Width: 100k
Window: Hamming
Beta: 6.76

WBFM Receive
Quadrature Rate: 75k
Audio Decimation: 8

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 2.4M
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

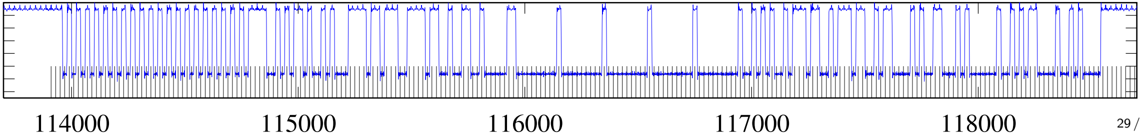
WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 75k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 9.375k
Trigger Mode: Auto
Y Axis Label: Counts

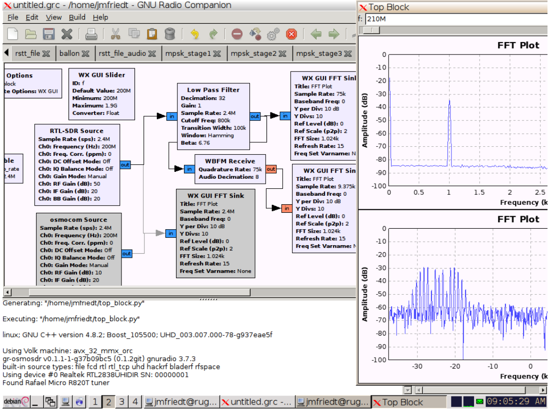
WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 9.375k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15

```
Generating: "/home/jmfriedt/top_block.py"

Executing: "/home/jmfriedt/top_block.py"

linux; GNU C++ version 4.8.2; Boost_105500; UHD_003.007.000-78-g937eae5f

Using Volk machine: avx_32_mmx_orc
gr-osmosdr v0.1.1-1-g37b09bc5 (0.1.2git) gnuradio 3.7.3
built-in source types: file fcd rtl rtl_tcp uhd hackrf bladerf rfspace
Using device #0 Realtek RTL2838UHIDIR SN: 00000001
Found Rafael Micro R820T 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 digital modem



Options
lock
te Options: WX GUI

WX GUI Slider
ID: f
Default Value: 200M
Minimum: 200M
Maximum: 1.9G
Converter: Float

Low Pass Filter
Decimation: 32
Gain: 1
Sample Rate: 2.4M
Cutoff Freq: 800k
Transition Width: 100k
Window: Hamming
Beta: 6.76

WX GUI FFT Sin
Title: FFT Plot
Sample Rate: 75k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: N

RTL-SDR Source
Sample Rate (sps): 2.4M
Ch0: Frequency (Hz): 200M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 50
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

ble
_rate
2.4M

WBFM Receive
Quadrature Rate: 75k
Audio Decimation: 8

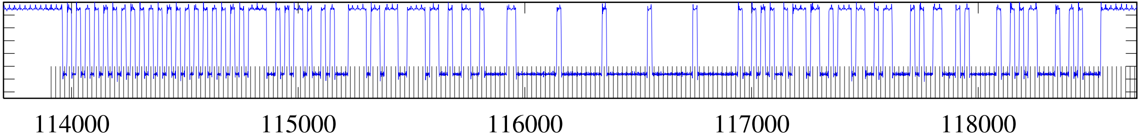
WX GUI FFT Sin
Title: FFT Plot
Sample Rate: 9.375k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: N

osmocom Source
Sample Rate (sps): 2.4M
Ch0: Frequency (Hz): 200M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 2.4M
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

```
Generating: "/home/jmfriedt/top_block.py"

Executing: "/home/jmfriedt/top_block.py"

linux; GNU C++ version 4.8.2; Boost_105500; UHD_003.007.000-78-g937eae5f

Using Volk machine: avx_32_mmx_orc
gr-osmosdr v0.1.1-1-g37b09bc5 (0.1.2git) gnuradio 3.7.3
built-in source types: file fcd rtl rtl_tcp uhd hackrf bladerf rfspace
Using device #0 Realtek RTL2838UHIDIR SN: 00000001
Found Rafael Micro R820T 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)

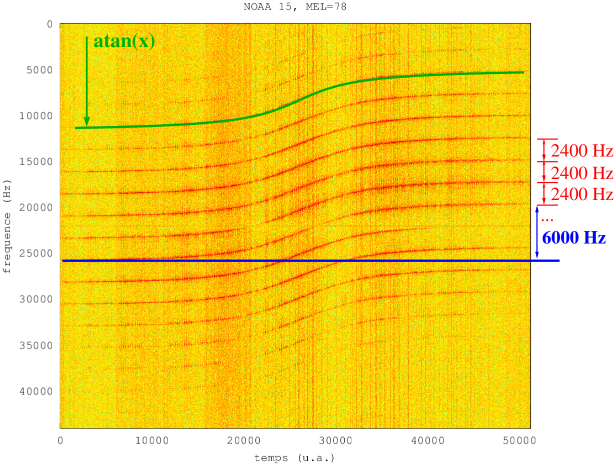
The carrier acts as a reference with respect to which sidebands carry the information

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)

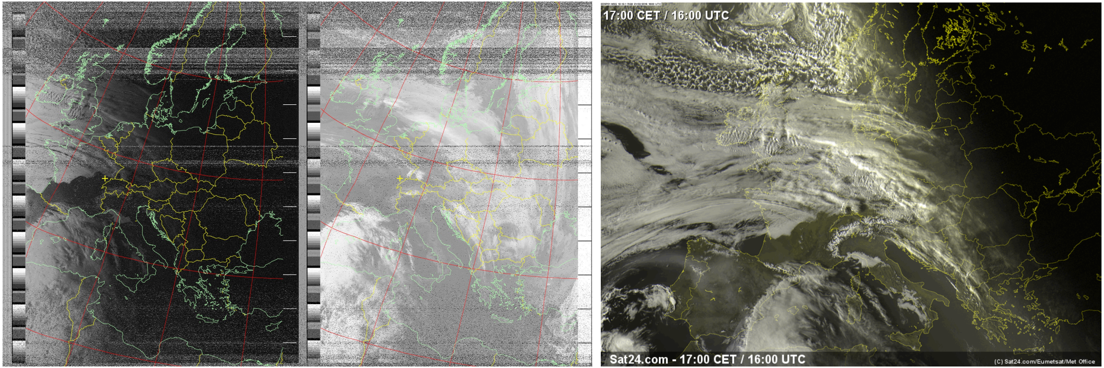
The carrier acts as a reference with respect to which sidebands carry the information



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)

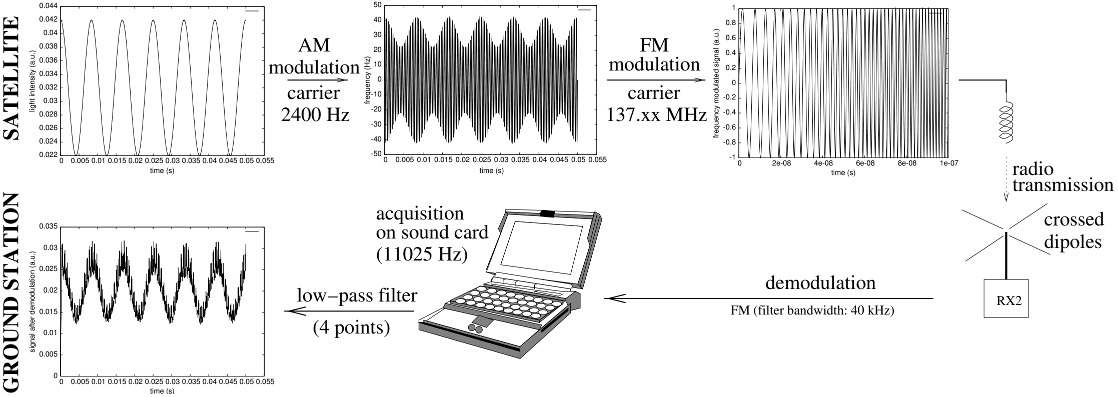
The carrier acts as a reference with respect to which sidebands carry the information



Without AFSK: FM is positively shifted (rising orbit Doppler shift), null and negatively shifted (setting orbit Doppler shift)⇒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 ($\simeq$ 137 MHz) [6] (APT).



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

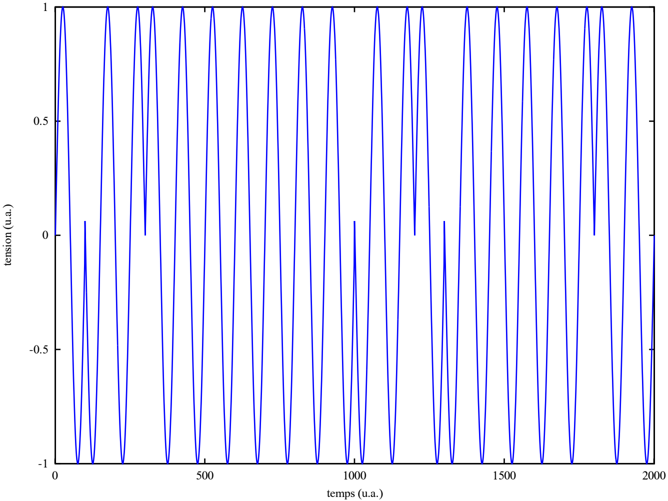
---

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

[7] Polar 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 demodulation[8].

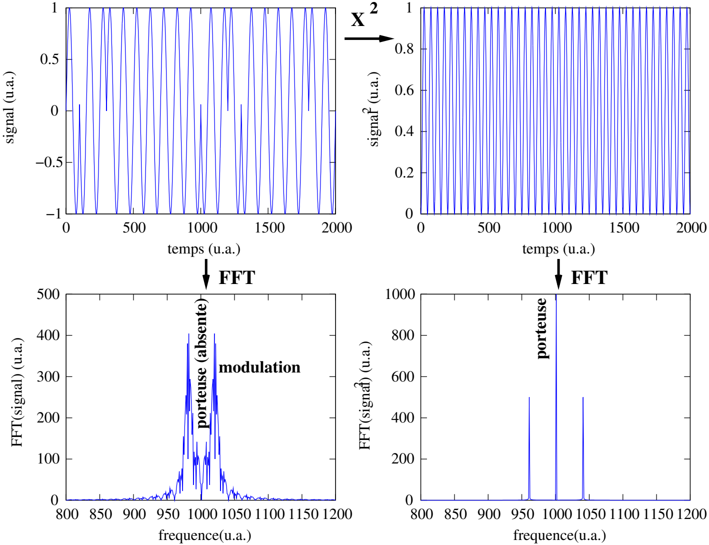
---

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

Phase demodulation

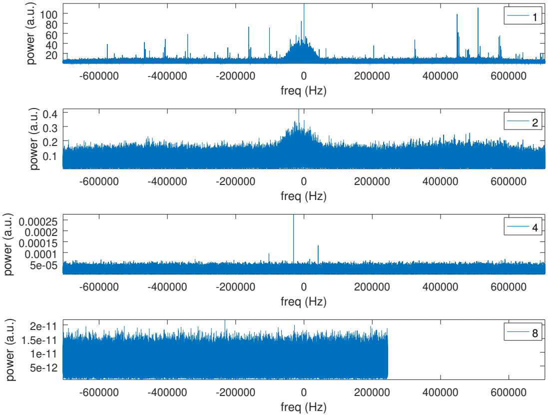
Requires accurate reproduction of the unmodulated carrier

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

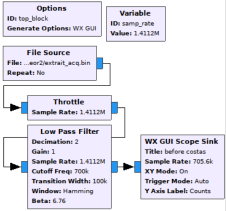


```
x=read_complex_binary('extrait_acq.bin');
fs=11025*64*2; % 1.4112 MHz
N=300e3;
f=linspace(-fs/2,fs/2,N);
for m=0:3
 subplot(4,1,m+1);
 plot(f,fftshift(abs(fft(x(1:N).^(2^m)))));
end
```

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



Options
ID: top_block
Generate Options: WX GUI

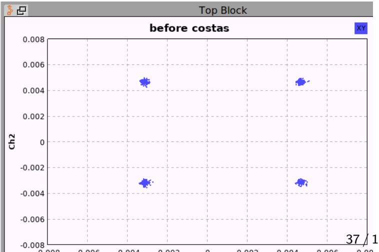
Variable
ID: samp_rate
Value: 1.4112M

File Source
File: ...eor2/extrait_acq.bin
Repeat: No

Throttle
Sample Rate: 1.4112M

Low Pass Filter
Decimation: 2
Gain: 1
Sample Rate: 1.4112M
Cutoff Freq: 700k
Transition Width: 100k
Window: Hamming
Beta: 6.76

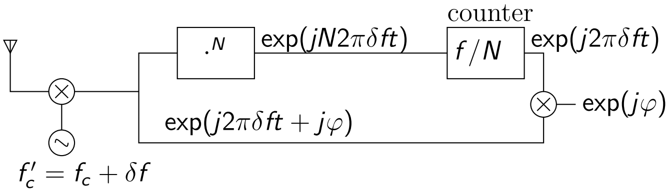
WX GUI Scope Sink
Title: before costas
Sample Rate: 705.6k
XY Mode: On
Trigger Mode: Auto
Y Axis Label: Counts



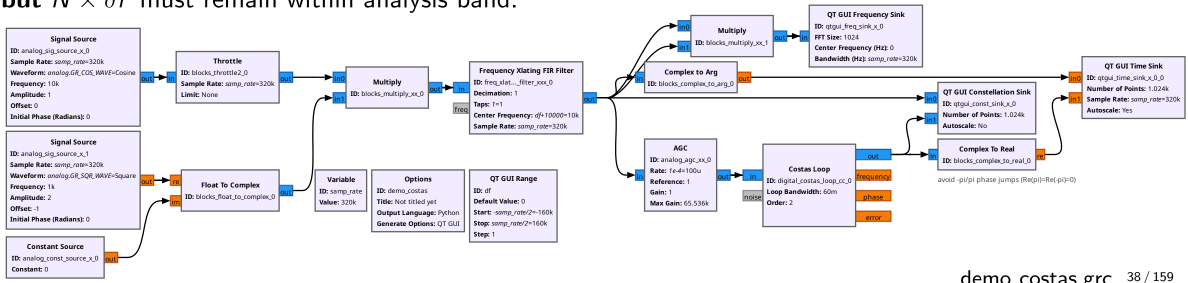
Phase demodulation: carrier recovery

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.



Signal Source
ID: analog_sig_source_x_0
Sample Rate: *samp_rate*=320k
Waveform: *analog.GR_COS_WAVE*=Cosine
Frequency: 10k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

Signal Source
ID: analog_sig_source_x_1
Sample Rate: *samp_rate*=320k
Waveform: *analog.GR_SQR_WAVE*=Square
Frequency: 1k
Amplitude: 2
Offset: -1
Initial Phase (Radians): 0

Constant Source
ID: analog_const_source_x_0
Constant: 0

Throttle
ID: blocks_throttle2_0
Sample Rate: *samp_rate*=320k
Limit: None

Float To Complex
ID: blocks_float_to_complex_0

Multiply
ID: blocks_multiply_xx_0

Variable
ID: samp_rate
Value: 320k

Options
ID: demo_costas
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Frequency Xlating FIR Filter
ID: freq_xlat..._filter_xxx_0
Decimation: 1
Taps: *1*=1
Center Frequency: *df+10000*=10k
Sample Rate: *samp_rate*=320k

QT GUI Range
ID: df
Default Value: 0
Start: *-samp_rate/2*=-160k
Stop: *samp_rate/2*=160k
Step: 1

Multiply
ID: blocks_multiply_xx_1

QT GUI Frequency Sink
ID: qtgui_freq_sink_x_0
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): *samp_rate*=320k

Complex to Arg
ID: blocks_complex_to_arg_0

QT GUI Time Sink
ID: qtgui_time_sink_x_0_0
Number of Points: 1.024k
Sample Rate: *samp_rate*=320k
Autoscale: Yes

QT GUI Constellation Sink
ID: qtgui_const_sink_x_0
Number of Points: 1.024k
Autoscale: No

AGC
ID: analog_agc_xx_0
Rate: *1e-4*=100u
Reference: 1
Gain: 1
Max Gain: 65.536k

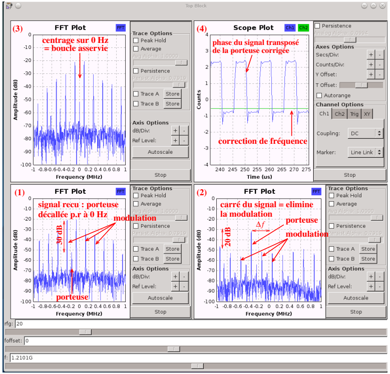
Costas Loop
ID: digital_costas_loop_cc_0
Loop Bandwidth: 60m
Order: 2

Complex To Real
ID: blocks_complex_to_real_0

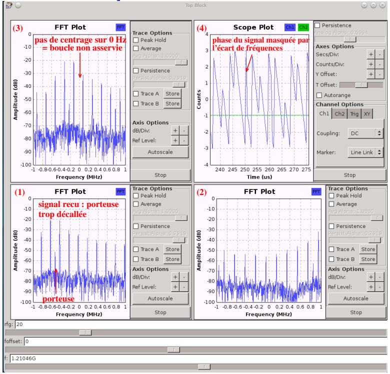
avoid -pi/pi phase jumps (Re(pi)=Re(-pi)=0)

demo_costas.grc

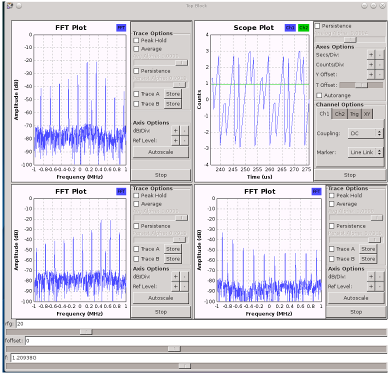
Phase demodulation: Costas loop



rfg: 20
foffset: 0
f: 1.2101G



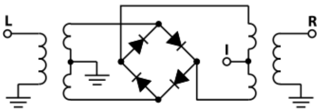
rfg: 20
foffset: 0
f: 1.21046G



rfg: 20
foffset: 0
f: 1.20938G

Experimental demonstration with a square wave modulating a BPSK signal (mixer):
Carrier offset: $\Delta\omega$

Electrical Schematic



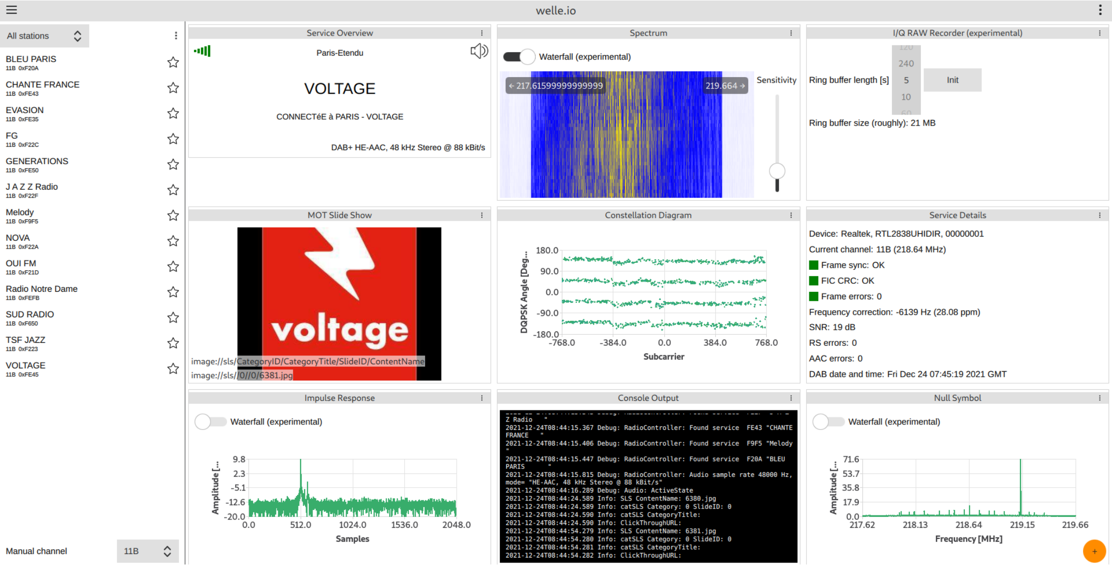
Modulation: $\varphi = [0;\pi]$

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

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



BLEU PARIS
11B 0xF20A
CHANTE FRANCE
11B 0xFE43
EVASION
11B 0xFE35
FG
11B 0xF22C
GENERATIONS
11B 0xFE50
J A Z Z Radio
11B 0xF22F
Melody
11B 0xF9F5
NOVA
11B 0xF22A
OUI FM
11B 0xF21D
Radio Notre Dame
11B 0xFEFB
SUD RADIO
11B 0xF650
TSF JAZZ
11B 0xF223
VOLTAGE
11B 0xFE45

Paris-Etendu

VOLTAGE

CONNECTéE à PARIS - VOLTAGE

DAB+ HE-AAC, 48 kHz Stereo @ 88 kBit/s

Ring buffer size (roughly): 21 MB

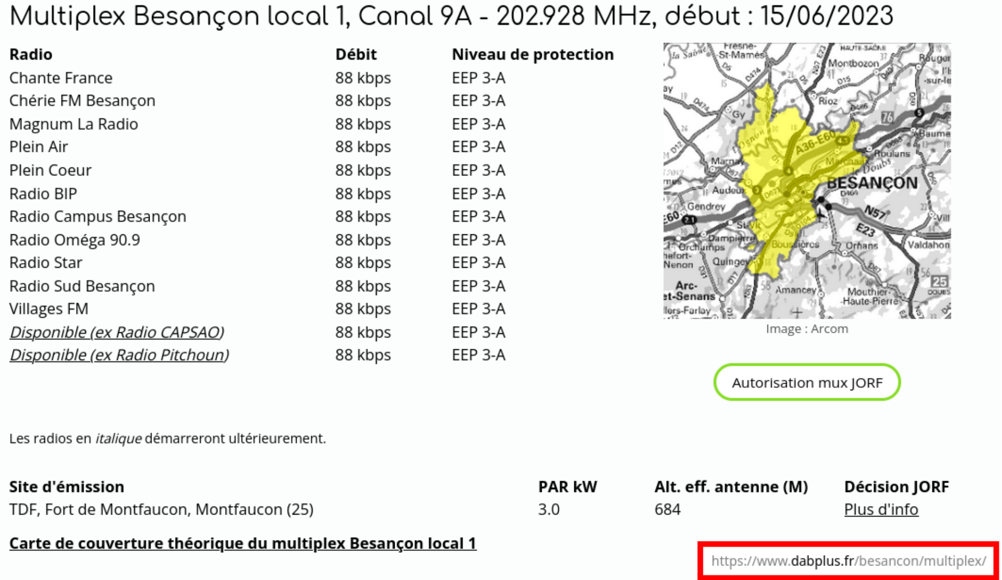
image://sls/CategoryID/CategoryTitle/SlideID/ContentName
image://sls/0//0/6381.jpg

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

```
Z Radio   "
2021-12-24T08:44:15.367 Debug: RadioController: Found service  FE43 "CHANTE
FRANCE   "
2021-12-24T08:44:15.406 Debug: RadioController: Found service  F9F5 "Melody
"
2021-12-24T08:44:15.447 Debug: RadioController: Found service  F20A "BLEU
PARIS     "
2021-12-24T08:44:15.815 Debug: RadioController: Audio sample rate 48000 Hz,
mode= "HE-AAC, 48 kHz Stereo @ 88 kBit/s"
2021-12-24T08:44:16.289 Debug: Audio: ActiveState
2021-12-24T08:44:24.589 Info: SLS ContentName: 6380.jpg
2021-12-24T08:44:24.589 Info: catSLS Category: 0 SlideID: 0
2021-12-24T08:44:24.590 Info: catSLS CategoryTitle:
2021-12-24T08:44:24.590 Info: ClickThroughURL:
2021-12-24T08:44:54.279 Info: SLS ContentName: 6381.jpg
2021-12-24T08:44:54.280 Info: catSLS Category: 0 SlideID: 0
2021-12-24T08:44:54.281 Info: catSLS CategoryTitle:
2021-12-24T08:44:54.282 Info: ClickThroughURL:
```

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
Disponible (ex Radio CAPSAO)	88 kbps	EEP 3-A
Disponible (ex Radio Pitchoun)	88 kbps	EEP 3-A

Image : Arcom

Les radios en *italique* démarreront ultérieurement.

Site d'émission	**PAR kW**	**Alt. eff. antenne (M)**	**Décision JORF**
TDF, Fort de Montfaucon, Montfaucon (25)	3.0	684	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 demodulation[9].

---

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

Radiofrequency enabled microcontroller internals [10]

See Texas Instrument's SimpleLink MCU products:

Figure 4-2 shows the CC3120R hardware overview.

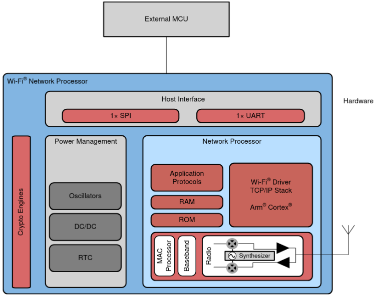
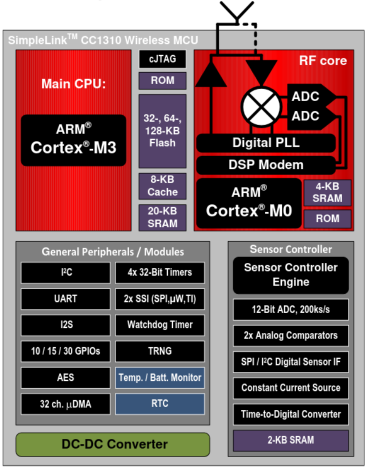


Figure 4-2. CC3120R Hardware Overview

- Carrier frequency range is determined by analog VCO tuning range
- Communication bandwidth is determined by ADC and DAC sampling rate

Figure 1-1 shows a block diagram for the CC1310 device.



Copyright © 2016, Texas Instruments Incorporated

Figure 1-1. CC1310 Block Diagram

Circuit Description

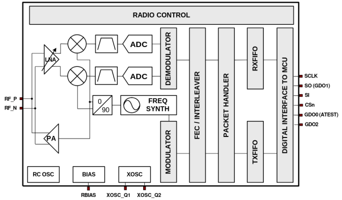
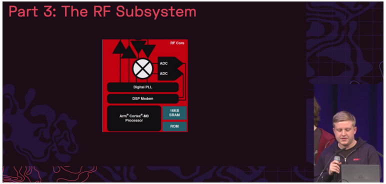


Figure 9: *CC1101* Simplified Block Diagram



Part 3: The RF Subsystem

---

[10] A. Batori & R. Pafford, *Beyond BLE: Cracking Open the Black-Box of RF Microcontrollers*, 38C3 (2024) at `https://media.ccc.de/v/38c3-beyond-ble-cracking-open-the-black-box-of-rf-microcontrollers`

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

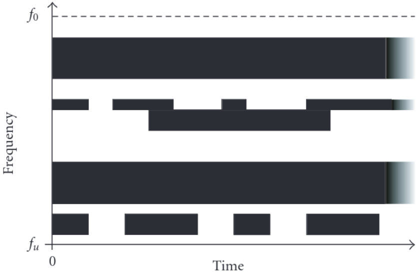


FIGURE 5: FDMA/TDMA signals over the time/frequency plane, spectrum pool.

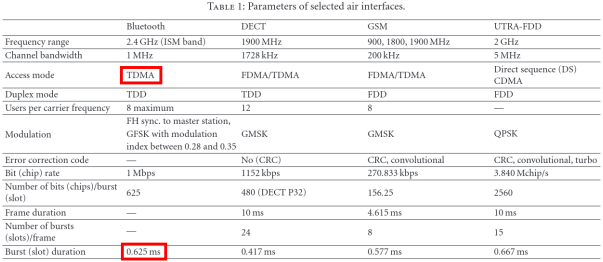


TABLE 1: Parameters of selected air interfaces.

	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)

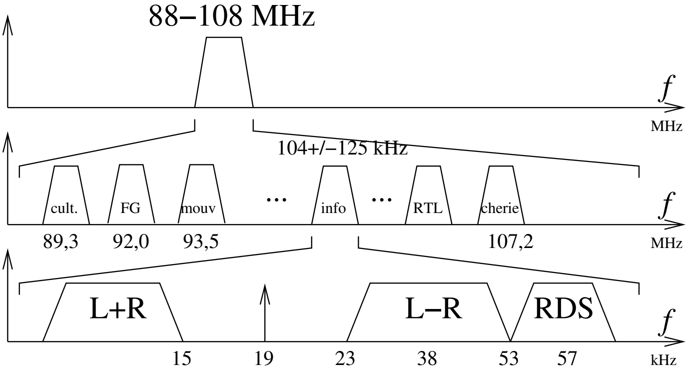
---

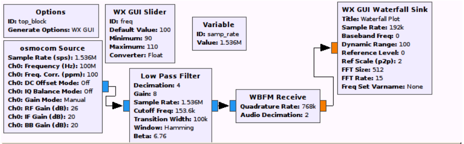
[a] *Time Division Multiple Access*
[b] *Frequency Division Multiple Access*
[c] *Spatial Division Multiple Access*
[d] *Code Division Multiple Access*

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 [11] [12]





Options
ID: top_block
Generate Options: WX GUI

WX GUI Slider
ID: freq
Default Value: 100
Minimum: 90
Maximum: 110
Converter: Float

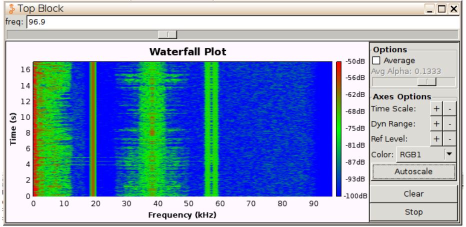
Variable
ID: samp_rate
Value: 1.536M

WX GUI Waterfall Sink
Title: Waterfall Plot
Sample Rate: 192k
Baseband Freq: 0
Dynamic Range: 100
Reference Level: 0
Ref Scale (p2p): 2
FFT Size: 512
FFT Rate: 15
Freq Set Varname: None

osmocom Source
Sample Rate (sps): 1.536M
Ch0: Frequency (Hz): 100M
Ch0: Freq. Corr. (ppm): 100
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 26
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Low Pass Filter
Decimation: 4
Gain: 8
Sample Rate: 1.536M
Cutoff Freq: 153.6k
Transition Width: 100k
Window: Hamming
Beta: 6.76

WBFM Receive
Quadrature Rate: 768k
Audio Decimation: 2



freq: 96.9

---

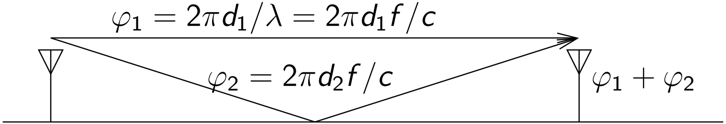
[11] B. Bloessl, First Steps in Receiving Digital Information with RDS/TMC, FOSDEM 2015, at `fosdem.org/2015/schedule/event/sdr_rds_tmc/`

[12] D. 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

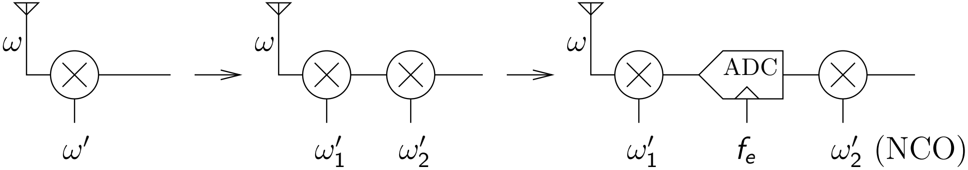
FHSS : Frequency Hopping Spread Spectrum

- 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

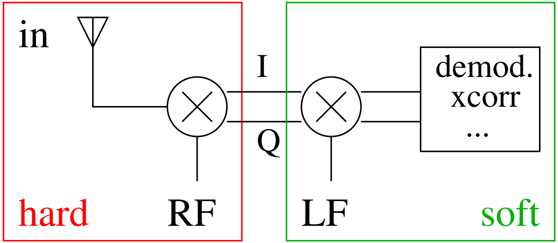
- Technical challenge of frequency hopping: a PLL is slow to stabilize, a DDS slow to configure
- Two step frequency transposition:
  $\exp(j \cdot (\omega - \omega') \cdot t) = \exp(j \cdot (\omega - \omega'_1 - \omega'_2) \cdot t)$ if $\omega' = \omega'_1 + \omega_2$
- We select ω'_1 in the radiofrequency 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 [13]
  `firdes.low_pass(gain,samp_rate,cutoff,transition,firdes.WIN_HAMMING)`



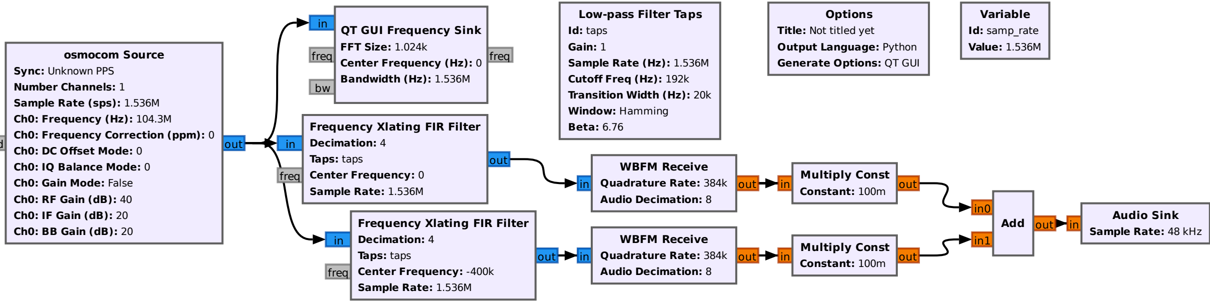
GNURadio's `Frequency Xlating FIR Filter`: frequency transposition followed by a low-pass filter, using as arguments the frequency offset (NCO) and the cutoff frequency (FIR taps).

---

[13] `https://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)



osmocom Source
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 1.536M
Ch0: Frequency (Hz): 104.3M
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 40
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 1.536M

Frequency Xlating FIR Filter
Decimation: 4
Taps: taps
Center Frequency: 0
Sample Rate: 1.536M

Frequency Xlating FIR Filter
Decimation: 4
Taps: taps
Center Frequency: -400k
Sample Rate: 1.536M

Low-pass Filter Taps
Id: taps
Gain: 1
Sample Rate (Hz): 1.536M
Cutoff Freq (Hz): 192k
Transition Width (Hz): 20k
Window: Hamming
Beta: 6.76

Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 1.536M

WBFM Receive
Quadrature Rate: 384k
Audio Decimation: 8

WBFM Receive
Quadrature Rate: 384k
Audio Decimation: 8

Multiply Const
Constant: 100m

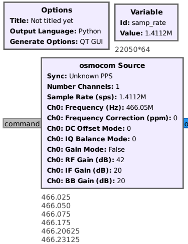
Multiply Const
Constant: 100m

Add

Audio Sink
Sample Rate: 48 kHz

FDMA: multichannel analysis

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

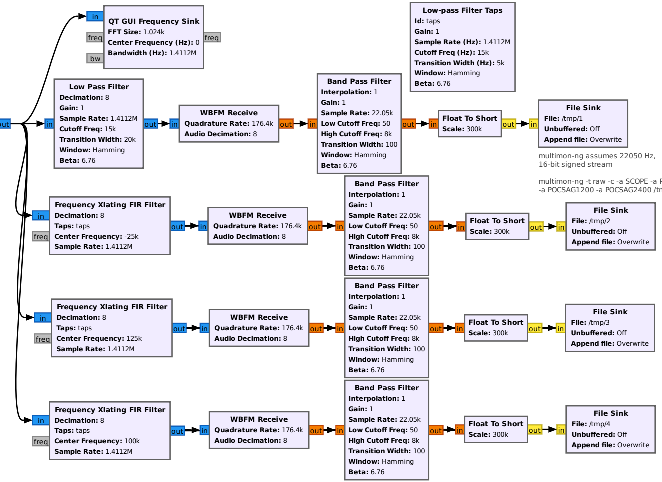


Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 1.4112M
22050*64

osmocom Source
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 1.4112M
Ch0: Frequency (Hz): 466.05M
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 42
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

466.025
466.050
466.075
466.175
466.20625
466.23125



QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 1.4112M

Low-pass Filter Taps
Id: taps
Gain: 1
Sample Rate (Hz): 1.4112M
Cutoff Freq (Hz): 15k
Transition Width (Hz): 5k
Window: Hamming
Beta: 6.76

Low Pass Filter
Decimation: 8
Gain: 1
Sample Rate: 1.4112M
Cutoff Freq: 15k
Transition Width: 20k
Window: Hamming
Beta: 6.76

WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

Band Pass Filter
Interpolation: 1
Gain: 1
Sample Rate: 22.05k
Low Cutoff Freq: 50
High Cutoff Freq: 8k
Transition Width: 100
Window: Hamming
Beta: 6.76

Float To Short
Scale: 300k

File Sink
File: /tmp/1
Unbuffered: Off
Append file: Overwrite

multimon-ng assumes 22050 Hz,
16-bit signed stream

multimon-ng -t raw -c -a SCOPE -a P
-a POCSAG1200 -a POCSAG2400 /tr

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: -25k
Sample Rate: 1.4112M

WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

Band Pass Filter
Interpolation: 1
Gain: 1
Sample Rate: 22.05k
Low Cutoff Freq: 50
High Cutoff Freq: 8k
Transition Width: 100
Window: Hamming
Beta: 6.76

Float To Short
Scale: 300k

File Sink
File: /tmp/2
Unbuffered: Off
Append file: Overwrite

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: 125k
Sample Rate: 1.4112M

WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

Band Pass Filter
Interpolation: 1
Gain: 1
Sample Rate: 22.05k
Low Cutoff Freq: 50
High Cutoff Freq: 8k
Transition Width: 100
Window: Hamming
Beta: 6.76

Float To Short
Scale: 300k

File Sink
File: /tmp/3
Unbuffered: Off
Append file: Overwrite

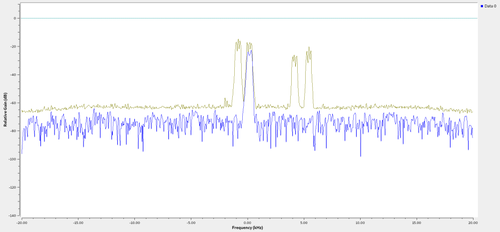
Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: 100k
Sample Rate: 1.4112M

WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

Band Pass Filter
Interpolation: 1
Gain: 1
Sample Rate: 22.05k
Low Cutoff Freq: 50
High Cutoff Freq: 8k
Transition Width: 100
Window: Hamming
Beta: 6.76

Float To Short
Scale: 300k

File Sink
File: /tmp/4
Unbuffered: Off
Append file: Overwrite



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).

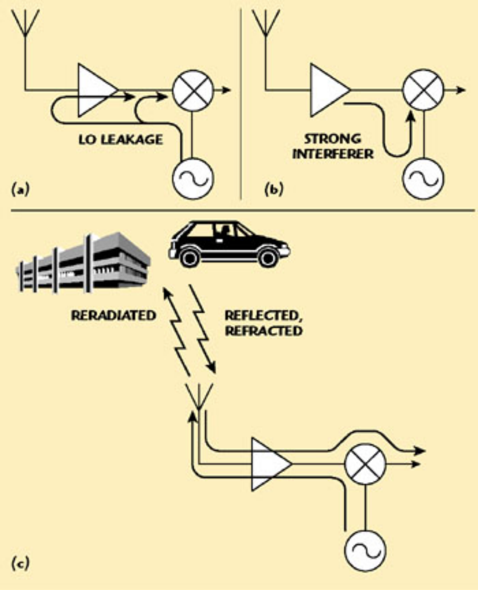


Fig. 8 DC offset mechanisms.

---

[a] A. 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`

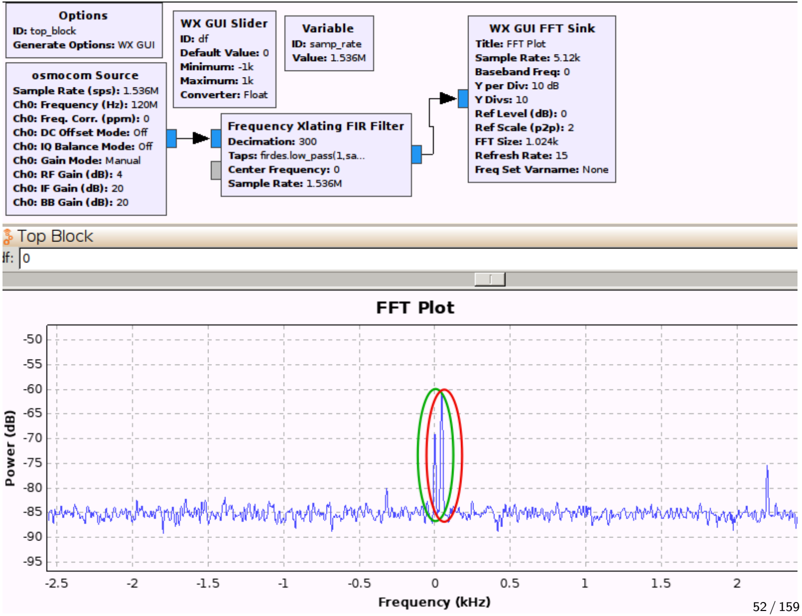
[b] B. Razavi, *Design considerations for Direct-Conversion Receivers*, IEEE Trans. on Circuits and Systems **44** (6), June 1997 (428–)

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



Options
ID: top_block
Generate Options: WX GUI

WX GUI Slider
ID: df
Default Value: 0
Minimum: -1k
Maximum: 1k
Converter: Float

Variable
ID: samp_rate
Value: 1.536M

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 5.12k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

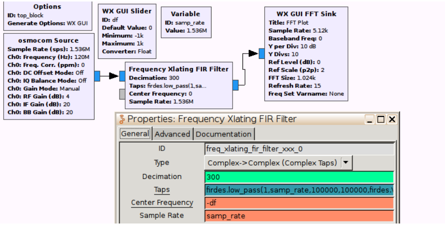
osmocom Source
Sample Rate (sps): 1.536M
Ch0: Frequency (Hz): 120M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 4
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Frequency Xlating FIR Filter
Decimation: 300
Taps: firdes.low_pass(1,sa...
Center Frequency: 0
Sample Rate: 1.536M

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)



Options
ID: top_block
Generate Options: WX GUI

WX GUI Slider
ID: df
Default Value: 0
Minimum: -1k
Maximum: 1k
Converter: Float

Variable
ID: samp_rate
Value: 1.536M

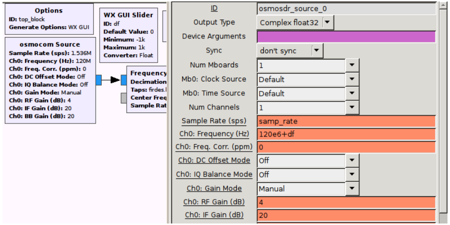
WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 5.12k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

osmocom Source
Sample Rate (sps): 1.536M
Ch0: Frequency (Hz): 120M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 4
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Frequency Xlating FIR Filter
Decimation: 300
Taps: firdes.low_pass(1,sa...
Center Frequency: 0
Sample Rate: 1.536M

Properties: Frequency Xlating FIR Filter

ID: freq_xlating_fir_filter_xxx_0
Type: Complex->Complex (Complex Taps)
Decimation: 300
Taps: firdes.low_pass(1,samp_rate,100000,100000,firdes.
Center Frequency: -df
Sample Rate: samp_rate



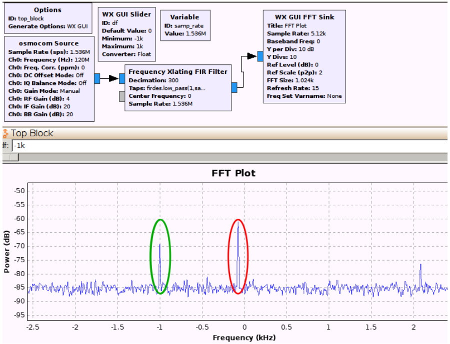
Options
ID: top_block
Generate Options: WX GUI

WX GUI Slider
ID: df
Default Value: 0
Minimum: -1k
Maximum: 1k
Converter: Float

osmocom Source
Sample Rate (sps): 1.536M
Ch0: Frequency (Hz): 120M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 4
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Frequency
Decimation
Taps: firdes.l
Center Freq
Sample Rat

ID: osmosdr_source_0
Output Type: Complex float32
Device Arguments:
Sync: don't sync
Num Mboards: 1
Mb0: Clock Source: Default
Mb0: Time Source: Default
Num Channels: 1
Sample Rate (sps): samp_rate
Ch0: Frequency (Hz): 120e6+df
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 4
Ch0: IF Gain (dB): 20



Options
ID: top_block
Generate Options: WX GUI

WX GUI Slider
ID: df
Default Value: 0
Minimum: -1k
Maximum: 1k
Converter: Float

Variable
ID: samp_rate
Value: 1.536M

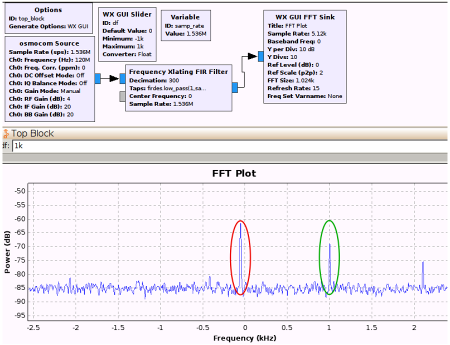
WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 5.12k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

osmocom Source
Sample Rate (sps): 1.536M
Ch0: Frequency (Hz): 120M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 4
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Frequency Xlating FIR Filter
Decimation: 300
Taps: firdes.low_pass(1,sa...
Center Frequency: 0
Sample Rate: 1.536M

Top Block

df: -1k



Options
ID: top_block
Generate Options: WX GUI

WX GUI Slider
ID: df
Default Value: 0
Minimum: -1k
Maximum: 1k
Converter: Float

Variable
ID: samp_rate
Value: 1.536M

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 5.12k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

osmocom Source
Sample Rate (sps): 1.536M
Ch0: Frequency (Hz): 120M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 4
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Frequency Xlating FIR Filter
Decimation: 300
Taps: firdes.low_pass(1,sa...
Center Frequency: 0
Sample Rate: 1.536M

Top Block

df: 1k

-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 [13] [14] 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 [15] [16]

All satellites transmit on the same carrier frequency

---

[13] `http://spaceflightnow.com/2014/10/13/gps-modernization-continues-with-quick-pace-of-launches/`

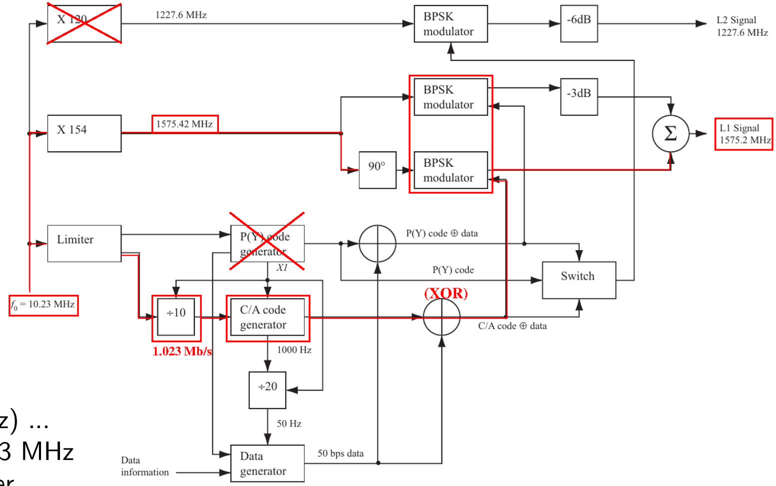
[14] B. Hubert, *How do GPS/Galileo really work & how the* `galmon.eu` *monitors all navigation satellites*, MCH (2022), `https://program.mch2022.org/mch2022/talk/QTUAXG/` at `https://www.youtube.com/watch?v=XjRHl2Rr9tQ`

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

[16] L. 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 principle [17]:



- 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 ($XOR(x, x) = 0$)

---

[17] K. 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 $s(t)$.

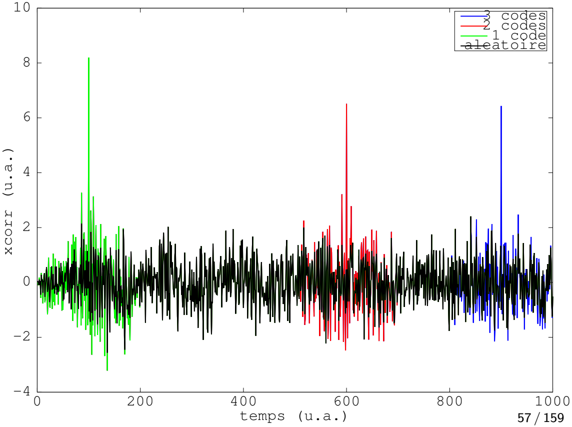
$$xcorr(\tau) = \int_{-\infty}^{+\infty} s(t) \times p^*(t+\tau)dt$$

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 $s(t)$.

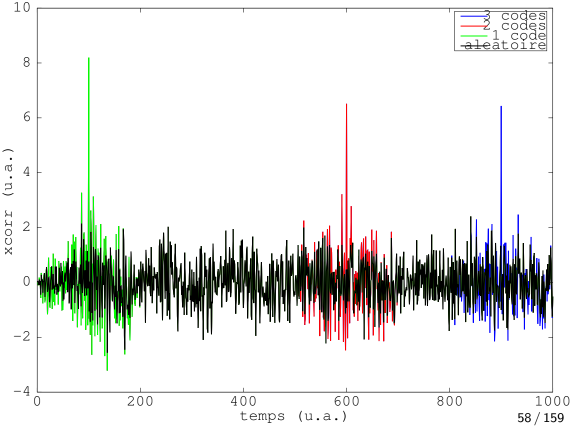
$$xcorr(\tau) = \int_{-\infty}^{+\infty} s(t) \times p^*(t+\tau)dt$$

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):

```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 $s(t)$.

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

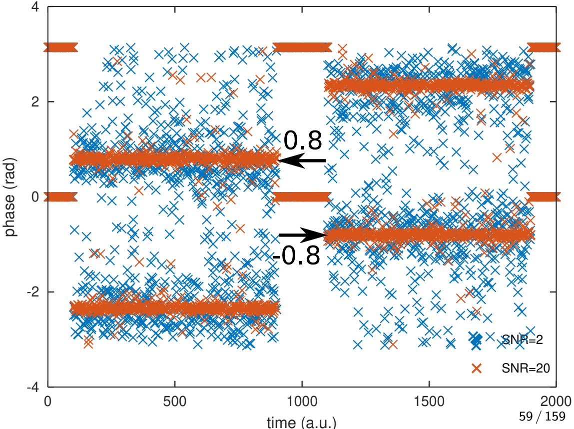
becoming for discrete time

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

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 $s(t)$.

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

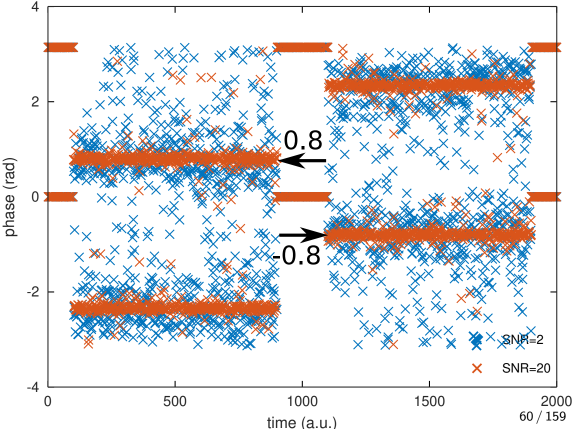
becoming for discrete time

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

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()
```

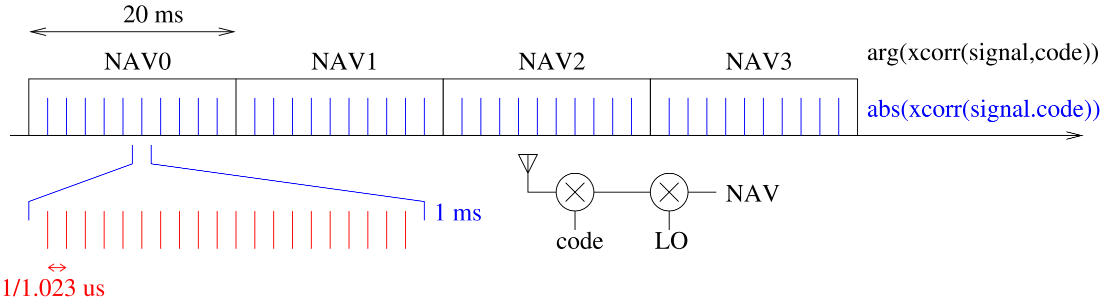
+ **phase** of the complex signal is transferred to the cross-correlation



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 additionnally XORed [18] over the previous signal (50 bps)



- 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?** $df \ll (\text{code duration})^{-1}$

---

[18] C. Moore, *Spread Spectrum Satcom Hacking*, DEFCON 23 (2015) `https://www.youtube.com/watch?v=2aBXpho5b7w`

CDMA: decoding GPS

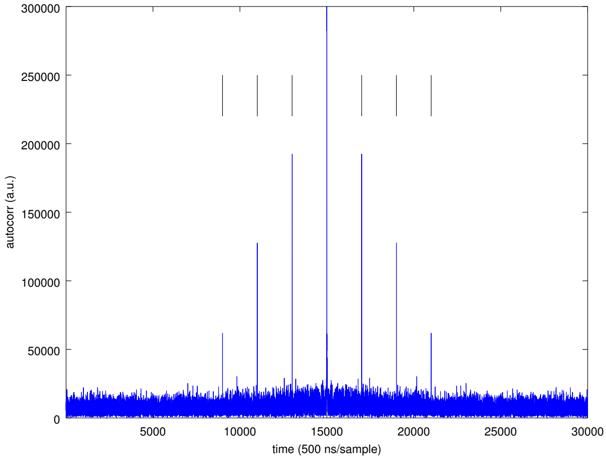
Even if we did not know the GPS encoding scheme, knowing that this code repeats is enough to assess whether a GPS signal is usable, ∀ LO offset:

$$\boxed{\textbf{autocorrelation}}$$

```
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 $p(t)$: $xcorr(x,x)=\int p(t)\exp(j2\pi\delta ft)\cdot p^*(t+\tau)\exp(2\pi\delta f(t+\tau))dt=\exp(j2\pi\delta f\tau)\int p(t)\cdot p^*(t+\tau)dt=\exp(j2\pi\delta f\tau)xcorr(p,p)$

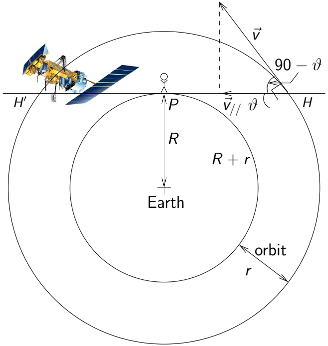
$$\boxed{|xcorr(x,x)|=|xcorr(p,p)|\quad\forall\delta f}$$



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

CDMA: decoding GPS

- Decoding GPS is *only* possible if the carrier frequency is accurately known ...
- ... whcih 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) !

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 [19]

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

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

---

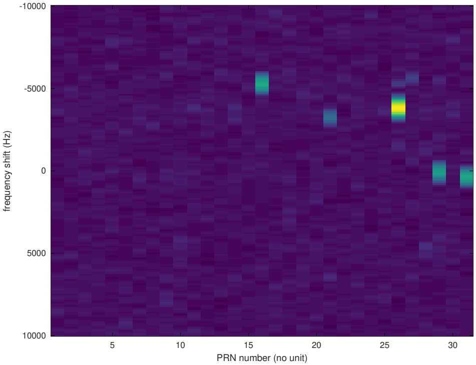
[19] Time-domain implementation on FPGA allows for pipelined computation as samples are collected

Basics on GPS encoding

GPS acquisition in 10 lines of Matlab program [20] (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
```

- 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



---

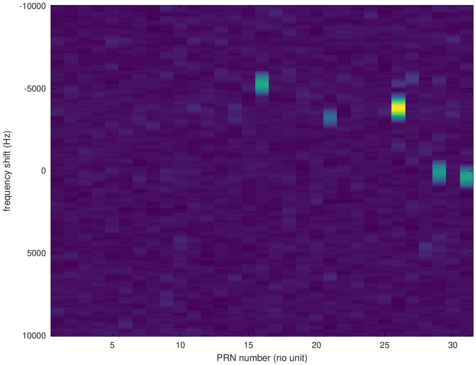
[20] using the C/A code generator `https://www.mathworks.com/matlabcentral/fileexchange/14670-gps-c-a-code-generator`

Basics on GPS encoding

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

```
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
```

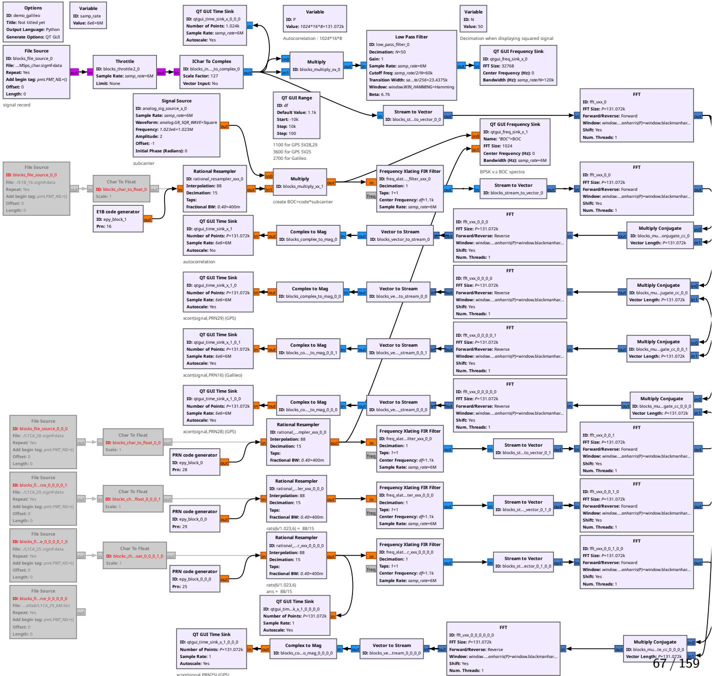
- Replace loops (inefficient) with matrix multiplication
- Parallelizing the frequency operations halves the computation time



Same but using GNU Radio

PRN sequences as Matlab archives:
`https://github.com/danipascual/GNSS-matlab/`

PRN generators in Python:
`https://github.com/pmonta/GNSS-DSP-tools`



Options
ID: demo_galileo
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 6e6=6M

File Source
ID: blocks_file_source_0
File: ...MSps_char.sigmf-data
Repeat: Yes
Add begin tag: pmt.PMT_NIL=()
Offset: 0
Length: 0

signal record

Throttle
ID: blocks_throttle2_0
Sample Rate: samp_rate=6M
Limit: None

IChar To Complex
ID: blocks_in..._to_complex_0
Scale Factor: 127
Vector Input: No

QT GUI Time Sink
ID: qtgui_time_sink_x_0_0_0
Number of Points: 1.024k
Sample Rate: samp_rate=6M
Autoscale: Yes

Variable
ID: P
Value: 1024*16*8=131.072k

Autocorrelation : 1024*16*8

Multiply
ID: blocks_multiply_xx_0

Low Pass Filter
ID: low_pass_filter_0
Decimation: N=50
Gain: 1
Sample Rate: samp_rate=6M
Cutoff Freq: samp_rate/2/N=60k
Transition Width: sa...te/256=23.4375k
Window: window.WIN_HAMMING=Hamming
Beta: 6.76

Variable
ID: N
Value: 50

Decimation when displaying squared signal

QT GUI Frequency Sink
ID: qtgui_freq_sink_x_0
FFT Size: 32768
Center Frequency (Hz): 0
Bandwidth (Hz): samp_rate/N=120k

Signal Source
ID: analog_sig_source_x_0
Sample Rate: samp_rate=6M
Waveform: analog.GR_SQR_WAVE=Square
Frequency: 1.023e6=1.023M
Amplitude: 2
Offset: -1
Initial Phase (Radians): 0

subcarrier

QT GUI Range
ID: df
Default Value: 1.1k
Start: -10k
Stop: 10k
Step: 100

1100 for GPS SV28,29
3600 for GPS SV25
2700 for Galileo

Stream to Vector
ID: blocks_st...to_vector_0_0

FFT
ID: fft_vxx_0
FFT Size: P=131.072k
Forward/Reverse: Forward
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

QT GUI Frequency Sink
ID: qtgui_freq_sink_x_1
Name: "BOC"=BOC
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): samp_rate=6M

BPSK v.s BOC spectra

File Source
ID: blocks_file_source_0_0
File: ./E1B_16.sigmf-data
Repeat: Yes
Add begin tag: pmt.PMT_NIL=()
Offset: 0
Length: 0

Char To Float
ID: blocks_char_to_float_0
Scale: 1

E1B code generator
ID: epy_block_1
Prn: 16

Rational Resampler
ID: rational_resampler_xxx_0
Interpolation: 88
Decimation: 15
Taps:
Fractional BW: 0.40=400m

Multiply
ID: blocks_multiply_xx_1

create BOC=code*subcarrier

Frequency Xlating FIR Filter
ID: freq_xlat..._filter_xxx_0
Decimation: 1
Taps: 1=1
Center Frequency: df=1.1k
Sample Rate: samp_rate=6M

Stream to Vector
ID: blocks_stream_to_vector_0

FFT
ID: fft_vxx_0_0
FFT Size: P=131.072k
Forward/Reverse: Forward
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

QT GUI Time Sink
ID: qtgui_time_sink_x_1
Number of Points: P=131.072k
Sample Rate: 6e6=6M
Autoscale: No

autocorrelation

Complex to Mag
ID: blocks_complex_to_mag_0

Vector to Stream
ID: blocks_vector_to_stream_0

FFT
ID: fft_vxx_0_0_0
FFT Size: P=131.072k
Forward/Reverse: Reverse
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

Multiply Conjugate
ID: blocks_mu...onjugate_cc_0
Vector Length: P=131.072k

QT GUI Time Sink
ID: qtgui_time_sink_x_1_0
Number of Points: P=131.072k
Sample Rate: 6e6=6M
Autoscale: Yes

xcorr(signal,PRN29) (GPS)

Complex to Mag
ID: blocks_complex_to_mag_0_0

Vector to Stream
ID: blocks_ve...to_stream_0_0

FFT
ID: fft_vxx_0_0_0_0
FFT Size: P=131.072k
Forward/Reverse: Reverse
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

Multiply Conjugate
ID: blocks_mu...jugate_cc_0_0
Vector Length: P=131.072k

QT GUI Time Sink
ID: qtgui_time_sink_x_1_0_1
Number of Points: P=131.072k
Sample Rate: 6e6=6M
Autoscale: Yes

xcorr(signal,PRN16) (Galileo)

Complex to Mag
ID: blocks_co..._to_mag_0_0_1

Vector to Stream
ID: blocks_ve..._stream_0_0_1

FFT
ID: fft_vxx_0_0_0_0_1
FFT Size: P=131.072k
Forward/Reverse: Reverse
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

Multiply Conjugate
ID: blocks_mu...gate_cc_0_0_1
Vector Length: P=131.072k

QT GUI Time Sink
ID: qtgui_time_sink_x_1_0_0
Number of Points: P=131.072k
Sample Rate: 6e6=6M
Autoscale: Yes

xcorr(signal,PRN28) (GPS)

Complex to Mag
ID: blocks_co..._to_mag_0_0_0

Vector to Stream
ID: blocks_ve..._stream_0_0_0

FFT
ID: fft_vxx_0_0_0_0_0
FFT Size: P=131.072k
Forward/Reverse: Reverse
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

Multiply Conjugate
ID: blocks_mu...gate_cc_0_0_0
Vector Length: P=131.072k

File Source
ID: blocks_file_source_0_0_0
File: ./L1CA_28.sigmf-data
Repeat: Yes
Add begin tag: pmt.PMT_NIL=()
Offset: 0
Length: 0

Char To Float
ID: blocks_char_to_float_0_0
Scale: 1

PRN code generator
ID: epy_block_0
Prn: 28

Rational Resampler
ID: rational_...mpler_xxx_0_0
Interpolation: 88
Decimation: 15
Taps:
Fractional BW: 0.40=400m

Frequency Xlating FIR Filter
ID: freq_xlat...ilter_xxx_0_0
Decimation: 1
Taps: 1=1
Center Frequency: df=1.1k
Sample Rate: samp_rate=6M

Stream to Vector
ID: blocks_st..._to_vector_0_1

FFT
ID: fft_vxx_0_0_1
FFT Size: P=131.072k
Forward/Reverse: Forward
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

File Source
ID: blocks_fi...rce_0_0_0_1
File: ./L1CA_29.sigmf-data
Repeat: Yes
Add begin tag: pmt.PMT_NIL=()
Offset: 0
Length: 0

Char To Float
ID: blocks_ch...float_0_0_1
Scale: 1

PRN code generator
ID: epy_block_0_0
Prn: 29

Rational Resampler
ID: rational_...ler_xxx_0_0_0
Interpolation: 88
Decimation: 15
Taps:
Fractional BW: 0.40=400m

rats(6/1.023,6) = 88/15

Frequency Xlating FIR Filter
ID: freq_xlat...ter_xxx_0_0_0
Decimation: 1
Taps: 1=1
Center Frequency: df=1.1k
Sample Rate: samp_rate=6M

Stream to Vector
ID: blocks_st..._vector_0_1_0

FFT
ID: fft_vxx_0_0_1_0
FFT Size: P=131.072k
Forward/Reverse: Forward
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

File Source
ID: blocks_fi...e_0_0_0_0_1_0
File: ./L1CA_25.sigmf-data
Repeat: Yes
Add begin tag: pmt.PMT_NIL=()
Offset: 0
Length: 0

Char To Float
ID: blocks_ch...oat_0_0_0_1_0
Scale: 1

PRN code generator
ID: epy_block_0_0_0
Prn: 25

Rational Resampler
ID: rational_...r_xxx_0_0_0_0
Interpolation: 88
Decimation: 15
Taps:
Fractional BW: 0.40=400m

rats(6/1.023,6)
ans = 88/15

Frequency Xlating FIR Filter
ID: freq_xlat...r_xxx_0_0_0_0
Decimation: 1
Taps: 1=1
Center Frequency: df=1.1k
Sample Rate: samp_rate=6M

Stream to Vector
ID: blocks_st...ector_0_1_0_0

FFT
ID: fft_vxx_0_0_1_0_0
FFT Size: P=131.072k
Forward/Reverse: Forward
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

QT GUI Time Sink
ID: qtgui_tim...k_x_1_0_0_0_0
Number of Points: P=131.072k
Sample Rate: 1
Autoscale: Yes

File Source
ID: blocks_fi...rce_0_0_0_0_0
File: ...atlab/L1CA_29_6M.bin
Repeat: Yes
Add begin tag: pmt.PMT_NIL=()
Offset: 0
Length: 0

QT GUI Time Sink
ID: qtgui_time_sink_x_1_0_0_0
Number of Points: P=131.072k
Sample Rate: 1
Autoscale: Yes

xcorr(signal,PRN25) (GPS)

Complex to Mag
ID: blocks_co...o_mag_0_0_0_0

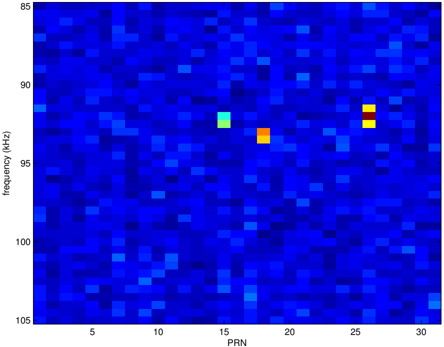
Vector to Stream
ID: blocks_ve...tream_0_0_0_0

FFT
ID: fft_vxx_0_0_0_0_0_0
FFT Size: P=131.072k
Forward/Reverse: Reverse
Window: window....anharris(P)=window.blackmanhar...
Shift: Yes
Num. Threads: 1

Multiply Conjugate
ID: blocks_mu...te_cc_0_0_0_0
Vector Length: P=131.072k

CDMA: decoding GPS

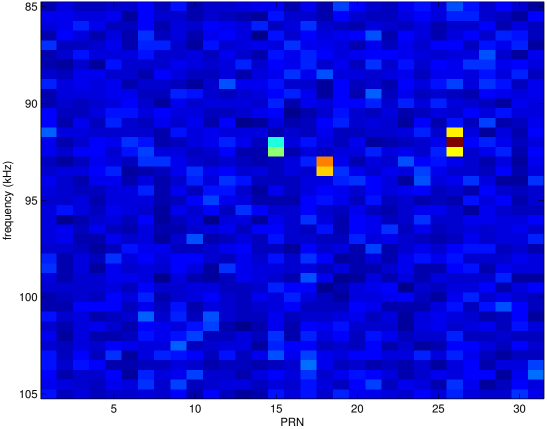
- CDMA basics: each useful bit (*navigation data*) is transmitted with its associated satellite identifier (PRN).
- All satellites transmit on the same carrier (1575.42 MHz), only their unique identifier allows differentiating each source.
- Each identifier is repeated every millisecond, NAV is at 50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

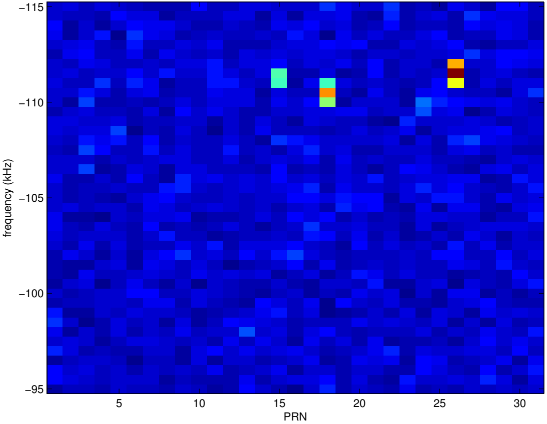


```
Searching for GPS Satellites in L1 band...
[ ... .  15  .  .  18  .  .  .  .  .  .  24  .  26  .  ... . ]
Total signal acquisition run time 11.5126 [seconds]
Reference Time:
  GPS Week: 811
  GPS TOW:  483088 38647.040000
  ~ UTC:    Fri Mar 13 15:11:29 2015
Current TOW obtained from SUPL assistance = 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.
```

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

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





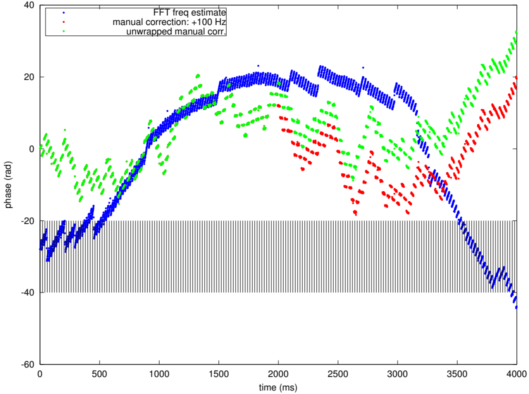
E4000 DVB-T
+59 ppm bias
=+91 kHz at 1575.42 MHz

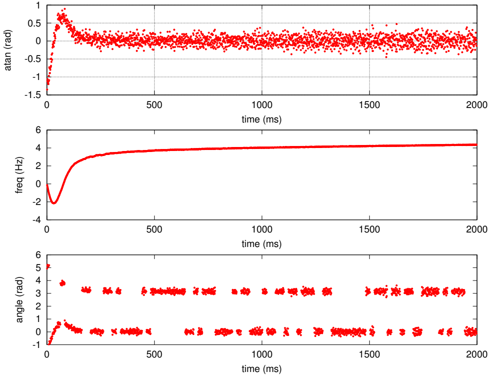
R820T DVB-T
-68 ppm bias
=-107 kHz at 1575.42 MHz

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

CDMA: decoding GPS

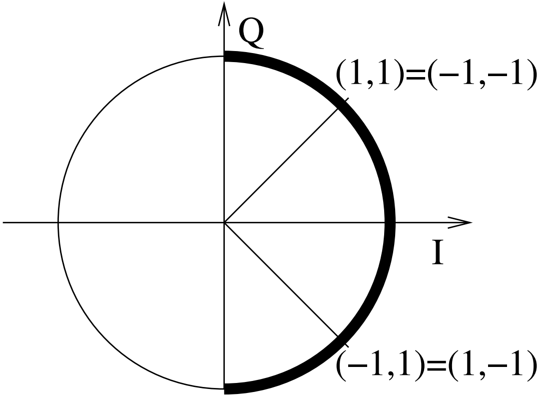
- Cross-correlating the received RF signal with orthogonal codes allows for identifying the source of the signal, but the message is lost
- once the **acquisition** phase is completed, **tracking** by controlling LO on the received carrier
- challenge: the phase is used both to encode the message and 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

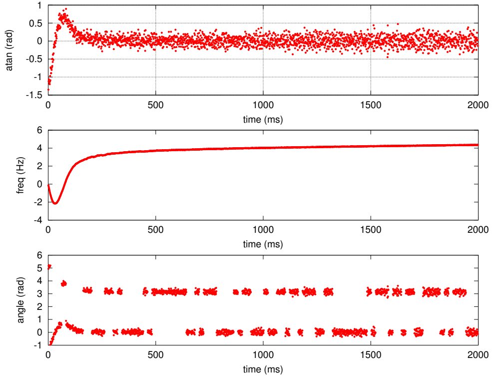




CDMA: decoding GPS

- 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 ?
- atan(Q/I) v.s atan2(Q, I): Q/I cannot detect 180° phase rotation, while atan2 provides NAV..

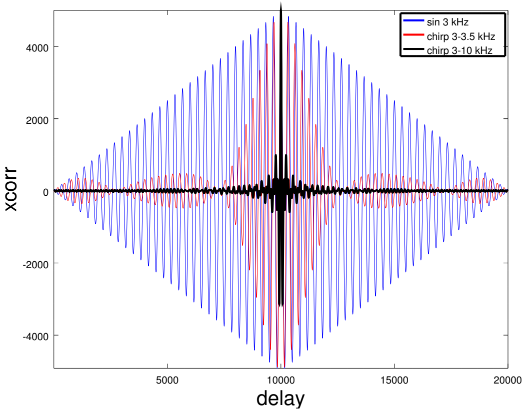




Pulse compression basics

- 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 $1/B$

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

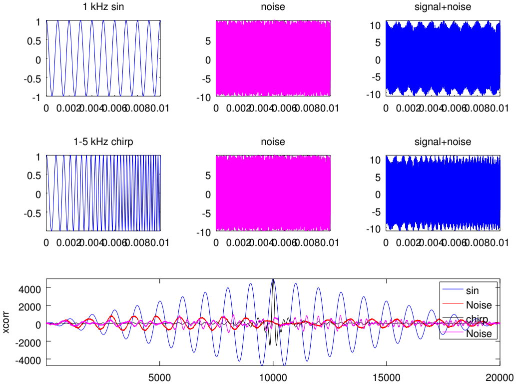


```
time=[0:1e-6:1e-2]; %samp. rate=1 us

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 $1/B$

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

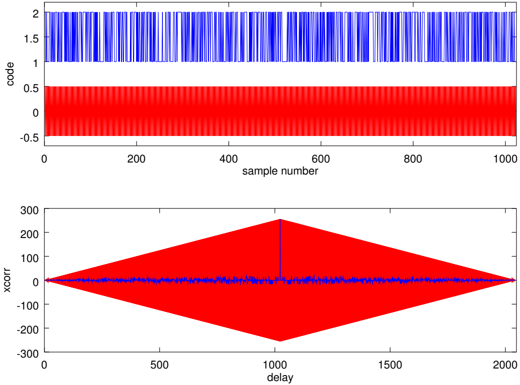


```
time=[0:1e-6:1e-2]; %samp. rate=1 us

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 $1/B$

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



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

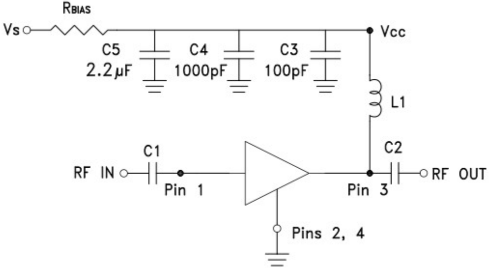
```
noise=rand(1023,1)’*7;
noise=noise-mean(noise);
b=[1:1023];
b=mod(b,2);b=b-mean(b);
plot(xcorr(b+noise,b),’r’);hold on

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 $k_B = 1.38 \cdot 10^{-23}$ 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

$$10 \cdot \log_{10}(\underbrace{1.38 \cdot 10^{-23}}_{k_B} \times \underbrace{293}_{T}) + \underbrace{30}_{W \to mW} = -174 \text{ dBm/Hz}$$

Application Circuit



Recommended Bias Resistor Values for Icc= 35 mA, Rbias= (Vs - Vcc) / Icc

Supply Voltage (Vs)	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. RBIAS provides DC bias stability over temperature.

---

[a] Since $J = W \cdot s$, then $k_B \times T$ is J=W/Hz

Link budget

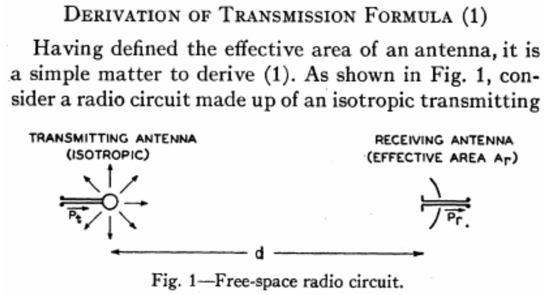
- a radiofrequency (electrical) power is emitted, either isotropically or in a directional pattern with an antenna gain G_1: $P_E \times G_1$
- this power spreads on a sphere centered on the emitter: in the case of isotropic emitter, the area of this sphere is, at a distance d, $4\pi d^2$
- if $G_1 > 1$, 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 $20\log_{10}(c/4/\pi) = 147.5$ dB



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.

---

[a] H.T.Friis *A Note on a Simple Transmission Formula*, Proc.I.R.E. 254-–256 (1946) ⟶

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. $FSPL = 182$ dB$\Rightarrow P_R = -152$ dBW$= -\mathbf{122}$ 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 ?

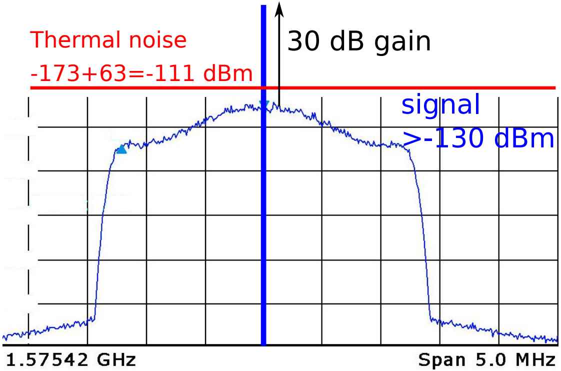
1. 2 MHz bandwidth (1023 kHz $\mathbb{C}$) so that $10\log_{10}(10^6) = 63$ dB
2. $-174 + 63 = -\mathbf{111}$ dBm> -122 dBm !
3. but 30 dB=1023 kHz/1 kHz pulse compression: $-122 + 30 = -\mathbf{92} > 111$ dBm ($SNR \simeq 22$ dB after compression)
4. the cross-correlation brings the signal out of the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

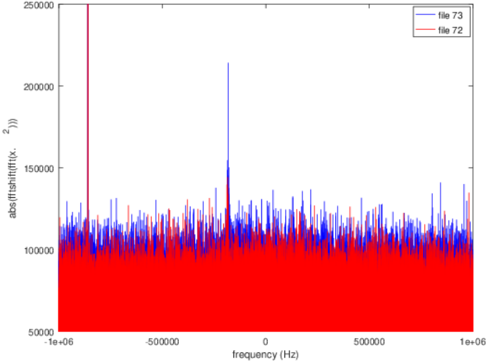
Back to GPS (BPSK)

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

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

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





Coarse estimate of (twice) the Doppler shift+frequency offset[21]

---

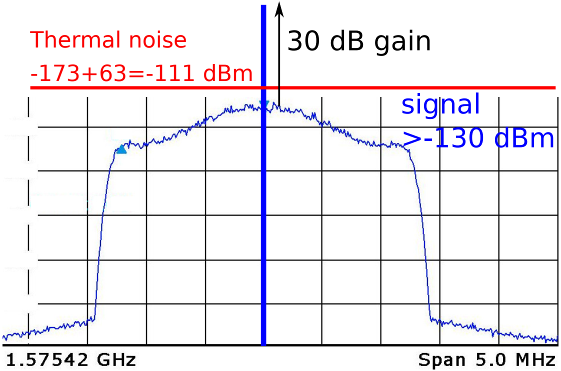
[21] P. Boven, *Observe, Hack, Make: GPS* (2013): used in Vaisala RS80 radiosonde

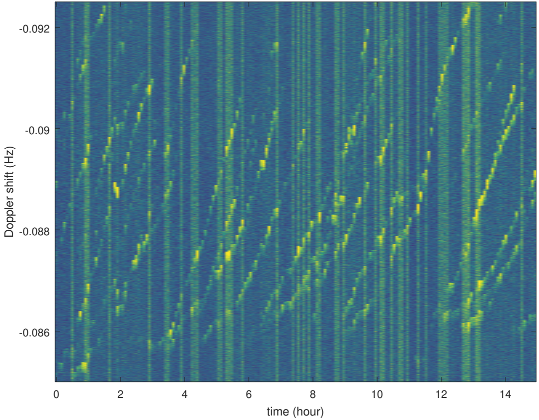
Back to GPS (BPSK)

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

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

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





Coarse estimate of (twice) the Doppler shift+frequency offset[21]

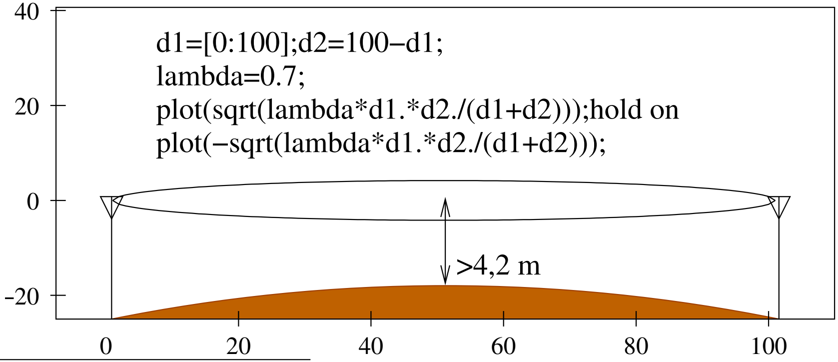
---

[21] P. Boven, *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)

  $r = \sqrt{\frac{n\lambda\cdot d_1\cdot d_2}{d_1+d_2}}$, $n \in \mathbb{N}$ ($n = 2p$ additive interferences, $n = 2p+1$ 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 [22], ground soil)



```
d1=[0:100];d2=100–d1;
lambda=0.7;
plot(sqrt(lambda*d1.*d2./(d1+d2)));hold on
plot(–sqrt(lambda*d1.*d2./(d1+d2)));
```

---

[22] É. Carry, J.-M Friedt, *Fonctionnement d'un téléphone mobile sous l'eau*, Hackable **35** (Oct.-Nov.-Dec. 2020) at `jmfriedt.free.fr/hackable_telephone.pdf`

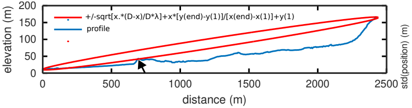
Example of Fresnel zone: talking to a glacier!

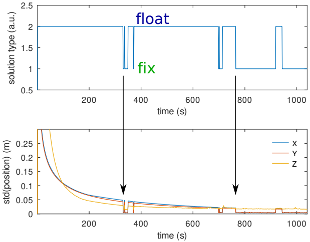
868-MHz LoRa ("Long Range") protocol to communicate from a basestation to sensors in the moraine of an Arctic glacier

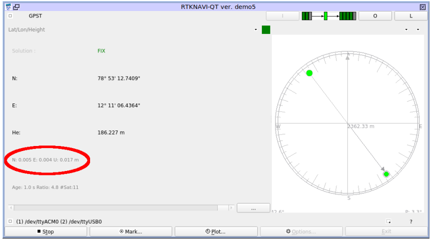
Range: $D = 2.4$ km (moraine)

Fresnel zone clearance:

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







Solution : FIX

N: 78° 53' 12.7409"

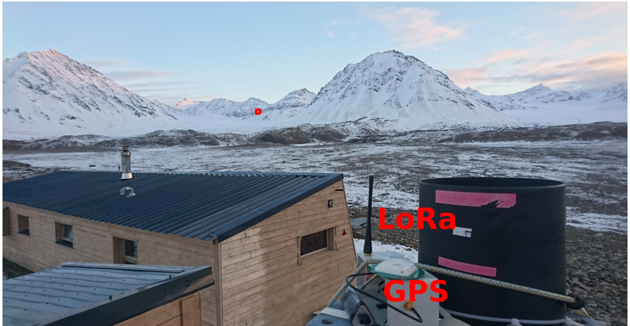
E: 12° 11' 06.4364"

He: 186.227 m

N: 0.005 E: 0.004 U: 0.017 m

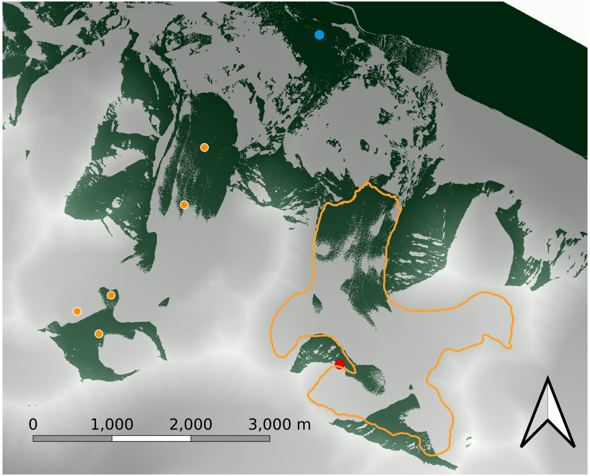
Age: 1.0 s Ratio: 4.8 #Sat:11

☐ (1) /dev/ttyACM0 (2) /dev/ttyUSB0

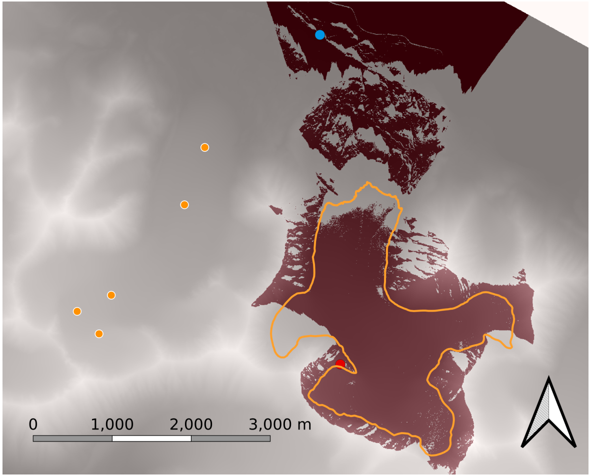


Radiofrequency communication range (QGis: viewshed analysis)

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



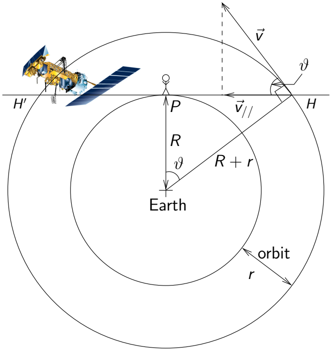
From Corbel station (blue dot)
including Midtre Lovén glacier (S. Filhol, Oslo Univ.)



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:
$\cos(\vartheta) = \frac{R}{r+R}$ 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:

$\cos(\vartheta) = \frac{R}{r+R}$ 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)

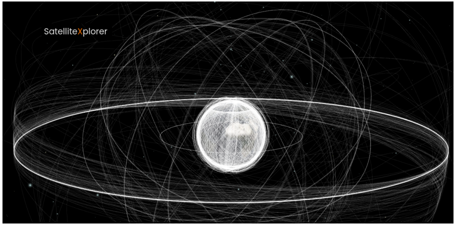
(Puy de Dome: $r \simeq 1.5$ km $\Rightarrow d \simeq 140$ km)

$\Rightarrow$ 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 $\simeq$100 min (Earth monitoring satellites, communication satellite e.g Starlink ...) $\Rightarrow$ communication stations close to the poles

Tromsø, Norway (70° N)



also in Lonyearbyen (78° N)

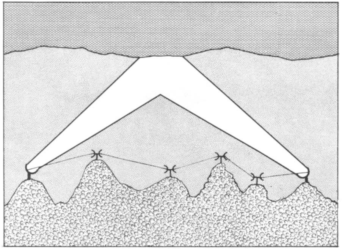
`https://geoxc-apps.bd.esri.com/space/satellite-explorer/`

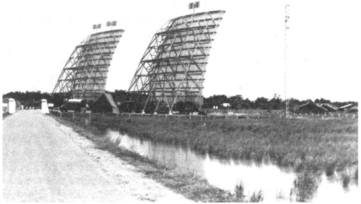
B. Wormdal, *The Satellite War* (2011)

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)

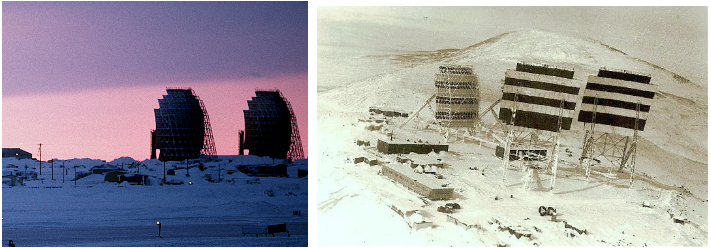
Effective Earth radius increases due to refraction of electromagnetic beam.

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
$\Rightarrow$ 66 km communication range.

How to link two transceivers at distances > 100 km ?



Standard atmosphere (lower N_s in cold regions):

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

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

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

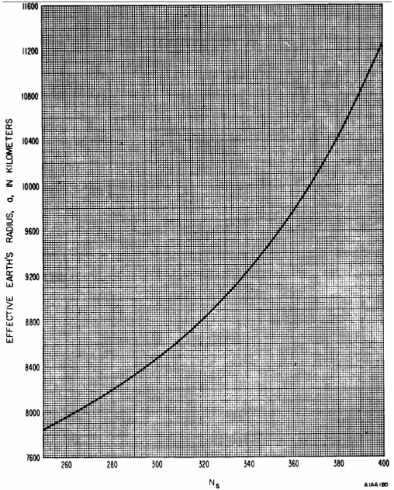


Figure 6-12. Effective Earth's Radius, a, Versus Surface Refractivity, N_s

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

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

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 errror $\Rightarrow P_R \propto \cos(\vartheta)^2$
- Gain G determined by effective area $A_{eff} = G\frac{\lambda^2}{4\pi}$

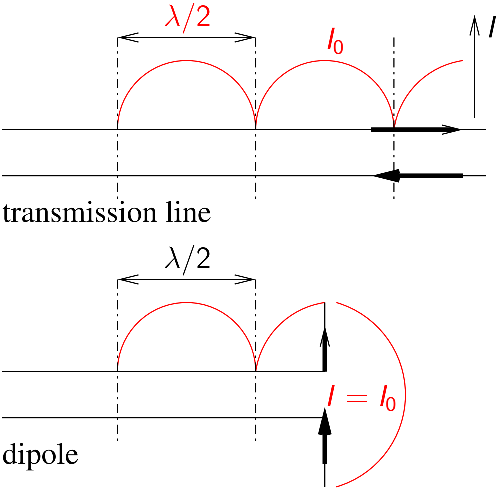
Antenna

- 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 errror $\Rightarrow P_R \propto \cos(\vartheta)^2$
- 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 [23] [24]



- 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

---

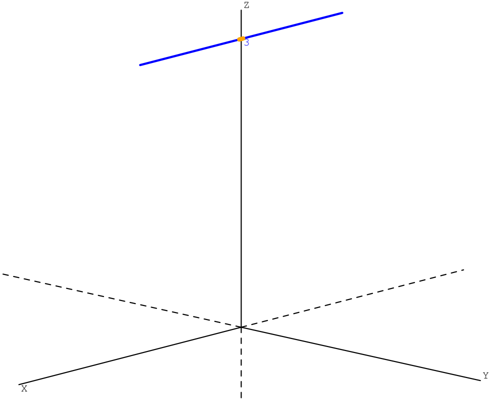
[23] C.A. Balanis, *Antenna Theory, Analysis and Design, 3rd Ed.*, Wiley Interscience (2005)

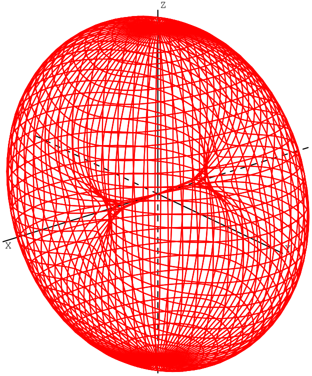
[24] C. Balanis, *The Evolution of Antenna Technology*, keynote at GNU Radio Conference (GRCon) 2023 at `https://www.youtube.com/watch?v=bp34nkSP4pM`

Dipole basics: radiation diagram... [25]

of a dipole simulated using NEC2 (`nec2c` package for Debian/GNU Linux), viewed using `xnecview` and now `xnec2c`

- Free space: null along the dipole axis





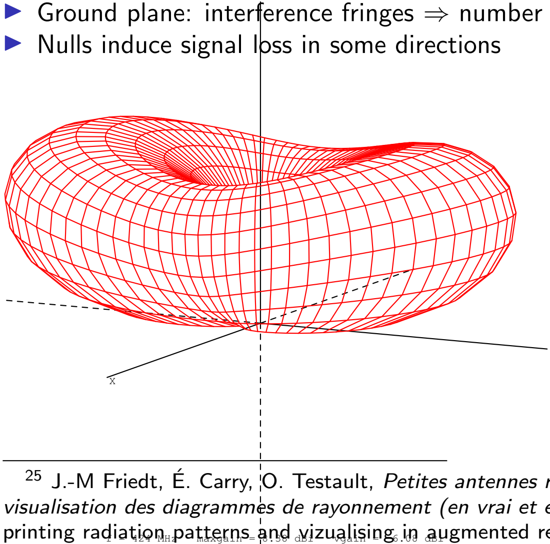
---

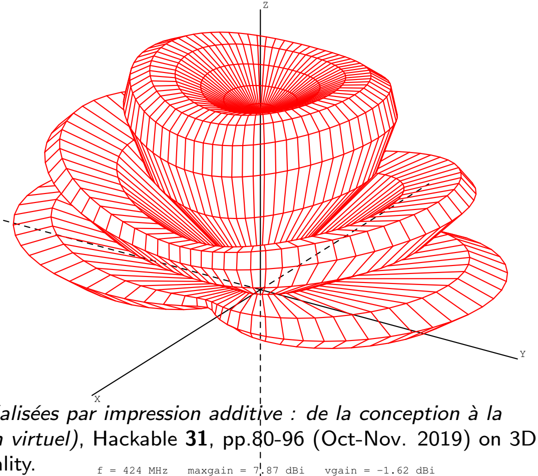
[25] J.-L. Smith, *Basic NEC with broadcast applications*, Focal Press (2008)

Dipole basics: radiation diagram...[25]

of a dipole simulated using NEC2 (`nec2c` package for Debian/GNU Linux), viewed using `xnecview` and now `xnec2c`

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

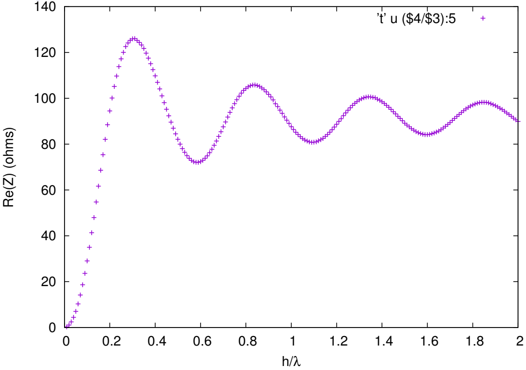


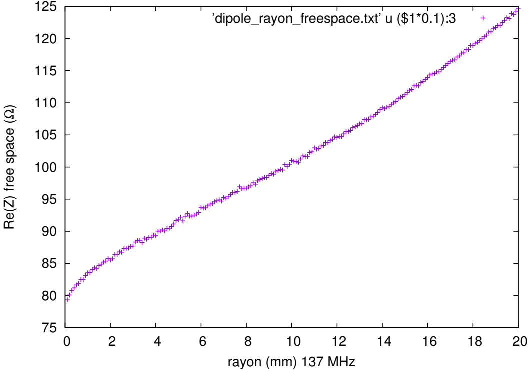


---

[25] 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 vizualising in augmented reality.

Dipole basics: from the ideal to the real dipole



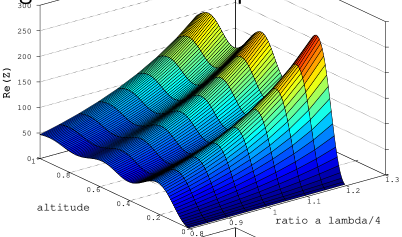


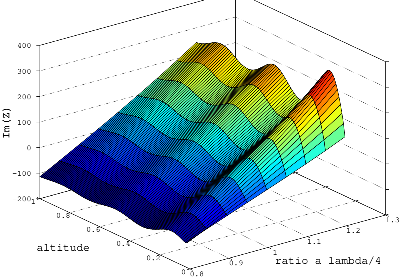
- Z depends on the radius of the radiating wire and its distance to ground
- Simulation example using NEC2:

```
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 [26]

- *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
- `xnecview` 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): $\Gamma = -1$ for total reflection at shorted end, $\Gamma = 0$ when no reflection, $\Gamma = +1$ for total reflection at open end
- $SWR = \frac{|V_{max}|}{|V_{min}|} = \frac{1+|\Gamma|}{1-|\Gamma|}$: 1 for perfect match when $Z_{load} = Z^*_{source}$, and $\to \infty$ for mismatch

---

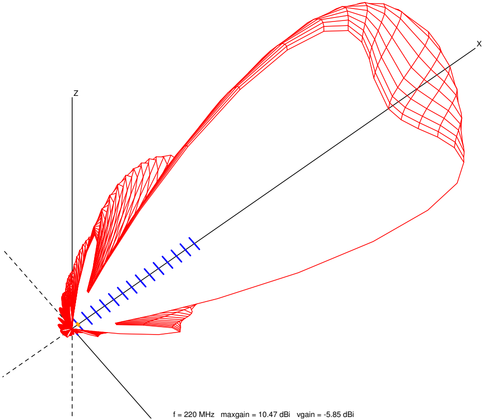
[26] https://en.wikipedia.org/wiki/Standing_wave_ratio

Yagi-Uda directional antenna

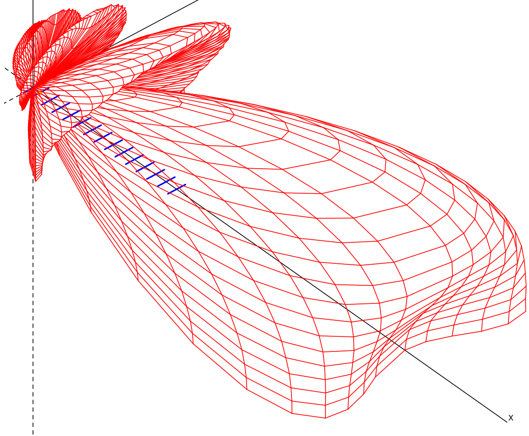
- a radiating element replaces the cable between the emitter and the receiver
- characteristic dimensions: $\lambda/2$ or, over a conducting plane, $\lambda/4$ by using mirror charges,
- impedance, directivity, polarization (NEC2, `www.nec2.org`)

```
#   freq.    Zr    Zi   SWR    gain   f/b   phi   theta
     220    26.1  2.2   1.9   10.47    -    0     76.5
```

$$\frac{P_R}{P_E} = \left(\frac{SWR-1}{SWR+1}\right)^2$$



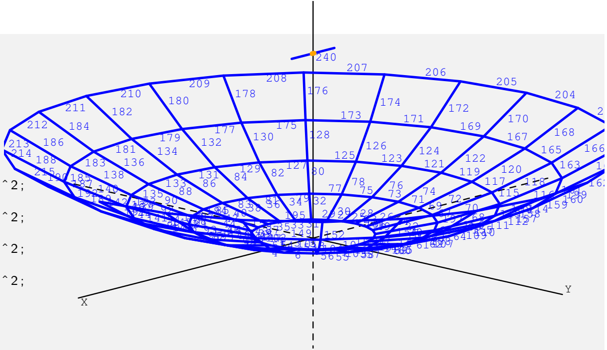
f = 220 MHz maxgain = 10.47 dBi vgain = -5.85 dBi



f = 220 MHz maxgain = 10.47 dBi vgain = -1.91 dBi

Optimizing antenna geometry

Let us assume that we are not aware of parabola characteristics ... how can we optimize the position of the feed ? [27]



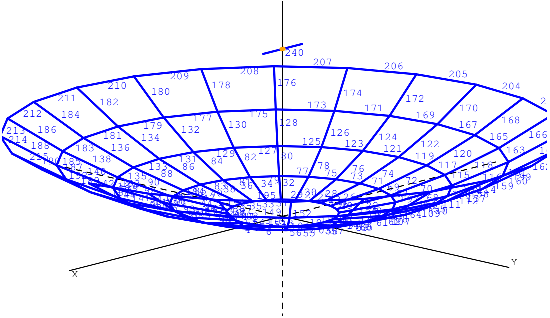
```
function maxgain=parabole_optim(ydipole)
freq=1500; % frequency in MHz
lambda=300/freq;
p=0.3; % focal point position 1/(4*p)
dtheta=10;
dr=0.1;
n=13;  % points/segment
rad=lambda/100;
r=[dr:dr:1-dr]; theta=[0:dtheta:360-dtheta]*pi/180;
x1=r'*cos(theta); y1=r'*sin(theta); r1=abs(x1+i*y1);z1=p*r1.^2;
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\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;
```

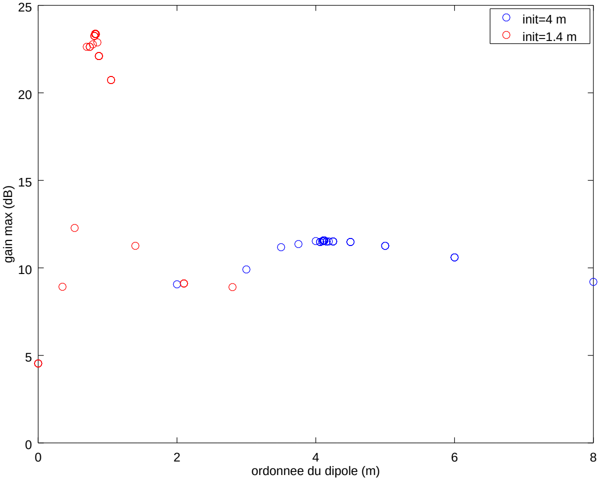
---

[27] J.-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]





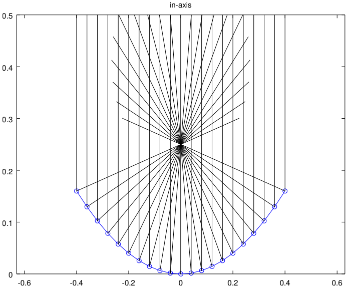
```
freq=1500; % frequency in MHz
lambda=300/freq;
p=0.3; % focal point position 1/(4*p)
dtheta=10;
dr=0.1;
n=13;     % points/segment
X0=1.4;  % ydipole initialization
Xopt=fminsearch('parabole_optim', X0);
printf("optim=%f theory=%f\n",Xopt,1/(4*p))
```

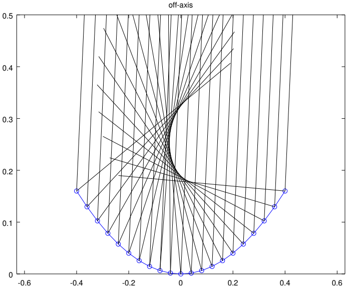
---

[a] J.-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`

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 $$\frac{A}{\lambda^2}\cdot\frac{4\pi}{1}\underbrace{\longrightarrow}_{\text{dB}}\sim 10\log_{10}\left(\frac{\pi^2 D^2}{\lambda^2}\right)$$
- 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 geostationnary satellite within a 30 km radius=$0.1^\circ \ll 5^\circ$)

NEC rules (Method of Moments validity)

- 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.

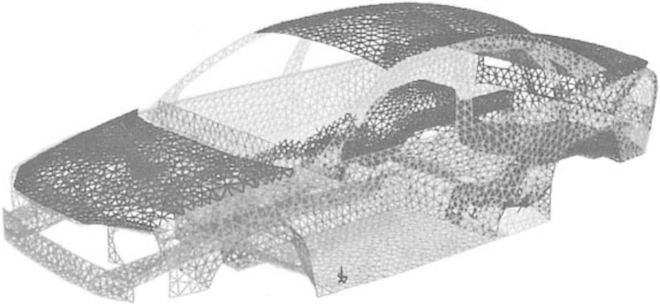


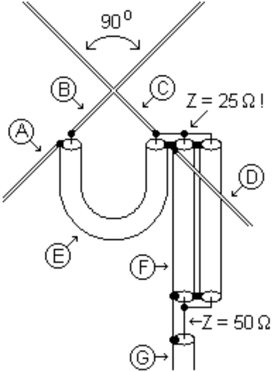
Fig. 6. NEC mesh of the considered geometry consisting of 23 449 segments used for performance comparisons.

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.



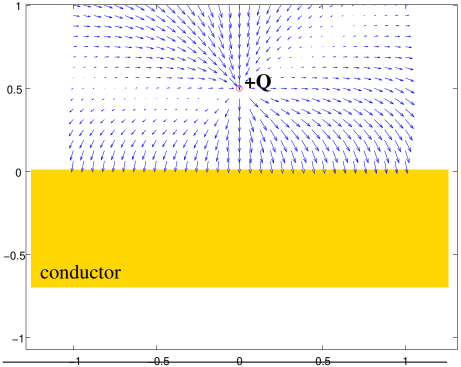
`hans.mayer.tv/html/crossdipole137.html`

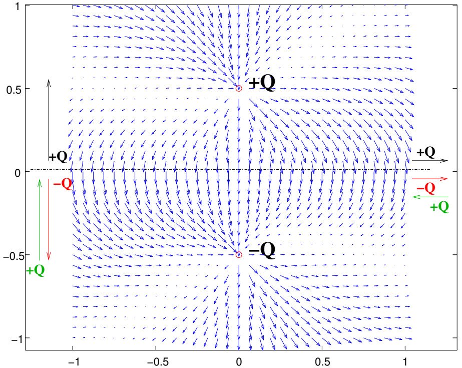
A, B, C, D: 510 mm
E: 50 Ω, 361 mm
F: 75 Ω × 2, 361 mm
G: 50 Ω

`http://jcoppens.com/ant/qfh/img/noaa17qfh.jpg` →

The monopole

- Mirror charge principle [28] [29]: 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=$36.5 + j21.5\ \Omega \Rightarrow$ shorten dipole to remove inductive component





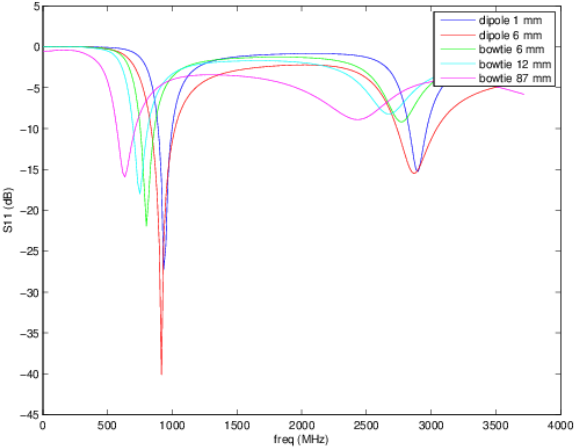
---

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

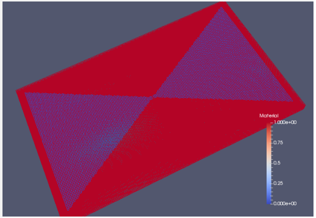
[29] R.P. Feynman & al., *The Feynman lectures on physics, vol.2* (1964), Chap 6-7 "The method of images" at `www.feynmanlectures.caltech.edu/II_06.html`

Antennas in complex media

- NEC2 can only handle wire-antennas in air
- homogeneous dielectric media: scale by $n = \sqrt{\varepsilon_r}$
- 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/`)

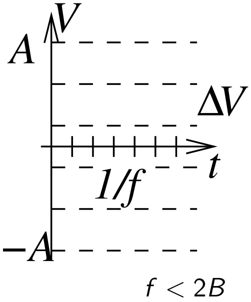


```
#title: Bowtie antenna
#domain: 0.10 0.050 0.200
#dx_dy_dz: 0.001 0.001 0.001
#time_window: 60e-9

#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 [32] [33]

1. Quantized [30] /noisy [31] 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 → power ; quantization → noise (SNR))

---

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

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

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

[33] O. 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=mBrIOSaPRDc`

Channel capacity

1. Quantized [30] /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): $2 \cdot B$

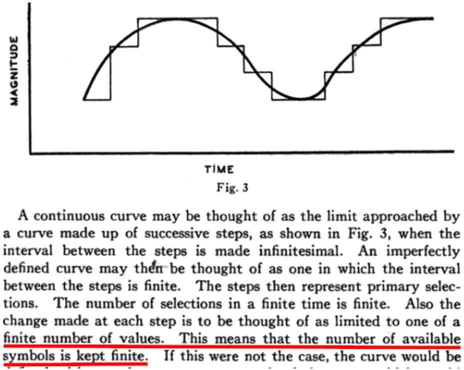


Fig. 3

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

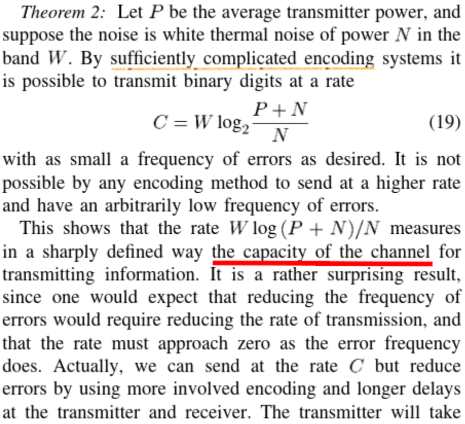
Hartley: $C = 2B \cdot \log_2(1 + A/\Delta V)$
Shannon: $C = B \cdot \log_2(1 + SNR)$
(amplitude → power ; quantization → noise (SNR))

---

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

Channel capacity

1. Quantized /noisy [30] 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

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

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

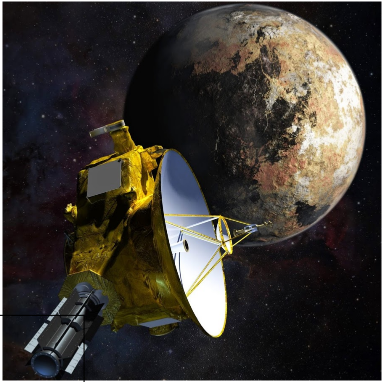
Hartley: $C = 2B \cdot \log_2(1 + A/\Delta V)$
Shannon: $C = B \cdot \log_2(1 + SNR)$
(amplitude → power ; quantization → noise (SNR))

---

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

Channel capacity

1. Quantized [30] /noisy [31] 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 → power ; quantization → 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) [32]

---

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

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

[32] `http://www.mike-willis.com/Tutorial/PF13.htm` & `http://propagation.ece.gatech.edu/ECE6390/project/Fall2010/Projects/group4/comm.html`

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 → power ; quantization → noise (SNR))

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

1. $10\log_{10}(15\ W) = 42$ dBm emitted @ 10 GHz
2. $FSPL = \left(\frac{4\pi \cdot d \cdot f}{c}\right)^2 = 20\log_{10}(d) + 20\log_{10}(f) - 147.55$ dB in SI units, so FSPL=306 dB
3. $P_R = -306 + 42 + 77 = -187$ dBm for a 77 dB gain parabola from 70 m diameter dish – $G = 4\pi A/\lambda^2 \rightarrow 10 \cdot \log_{10}\left((\pi D/\lambda)^2\right)$
4. Sky background noise [30]: $k_B T = 1.38 \cdot 10^{-23} \cdot 20\ K = -186$ dBm/Hz (or -176 dBm/Hz @ 200 K)
5. For small SNR, $\log(1 + SNR) \simeq SNR \Rightarrow C \simeq \frac{B \times P_R}{B \times k_B \cdot T} = \frac{P_R}{k_B \cdot T}$ $\forall B$

---

[30] C.T. Stelzried *& al.*, System Noise Concepts with DSN Applications @ `https://descanso.jpl.nasa.gov/monograph/series10/02_Reid_chapt2.pdf`

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): $2 \cdot B$

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

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

1. $10 \log_{10}(15\ W) = 42$ dBm emitted @ 10 GHz
2. $FSPL = \left(\frac{4\pi \cdot d \cdot f}{c}\right)^2 = 20 \log_{10}(d) + 20 \log_{10}(f) - 147.55$ dB in SI units, so FSPL=306 dB
3. $P_R = -306 + 42 + 77 = -187$ dBm for a 77 dB gain parabola from 70 m diameter dish –
4. Sky background noise : $k_B T = 1.38 \cdot 10^{-23} \cdot 20\ K = -186$ dBm/Hz (or -176 dBm/Hz @ 200 K)
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 dB → $10^{-1/10} \simeq 0.8$ b/s (or -187-(-176)=-11 dB → $10^{-11/10} \simeq 0.08$ 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 → power ; quantization → noise (SNR))

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

1. $10\log_{10}(15\ W) = 42$ dBm emitted @ 10 GHz
2. $FSPL = \left(\frac{4\pi \cdot d \cdot f}{c}\right)^2 = 20\log_{10}(d) + 20\log_{10}(f) - 147.55$ dB in SI units, so FSPL=306 dB
3. $P_R = -306 + 42 + 77 = -187$ dBm for a 77 dB gain parabola from 70 m diameter dish –
4. Sky background noise : $k_B T = 1.38 \cdot 10^{-23} \cdot 20\ K = -186$ dBm/Hz (or -176 dBm/Hz @ 200 K)
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. Additionnal information: HGA is 2.1 m parabola=47 dBi gain @ 10 GHz $\Rightarrow C = -1 + 47 = 46$ dB $\Rightarrow 10^{46/10} = 40000$ bps (or $10^{36/10} \simeq 4000$ bps @ 200 K)

Asymptotic limits [30]

SNR dependent channel capacity with bandwidth:

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

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

$C = B\log_2(1+SNR) \simeq B \times \log_2(SNR)$
doubling $B{=}B'$ yields
$C' = B'\log_2(SNR) = B'\log_2\left(\frac{P_E}{k_B T B'}\right) = 2B\log_2\left(\frac{P_E}{2k_B TB}\right)$
$= 2B\left(\log_2\left(\frac{P_E}{k_B TB}\right) - 1\right)$
$\simeq 2B\log_2\left(\frac{P_E}{k_B TB}\right)$ since $SNR \gg 1$
$\Rightarrow C' \simeq 2C$
Doubling B doubles C

bandwidth limited regime

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

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

$B\log_2(1+SNR) \simeq B \times SNR$
$= B \times \frac{P_E}{k_B TB} = \frac{P_E}{k_B T}$

No impact of B, only P_E defines C

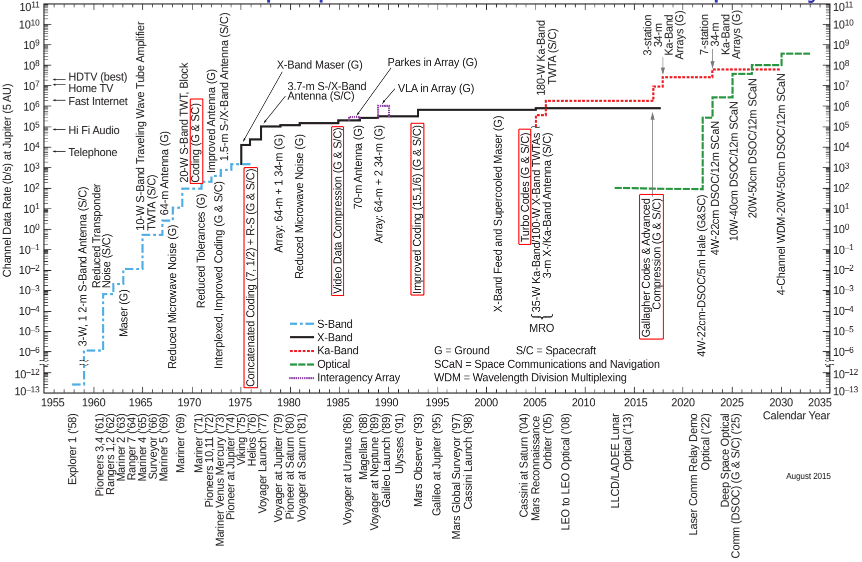
power limited regime

Spatial communications, deep space or LEO satellites

---

[30] J.-M Friedt, H. Boeglen, *Décodage d'images numéeriques 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 Capability[31]



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

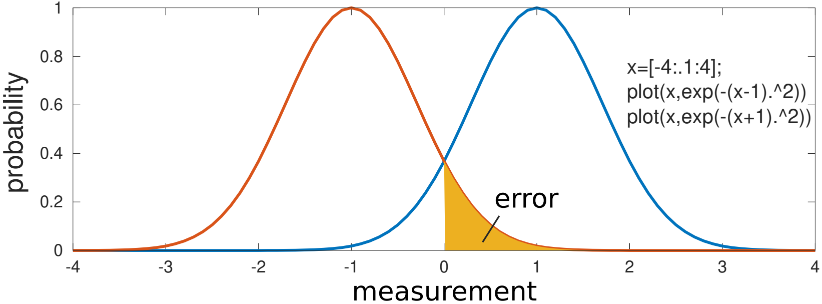
Name	Frequency range
VHF	30–300 MHz
UHF	300–3000 MHz
L	1–2 GHz
S	2–4 GHz
C	4–8 GHz
X	8–12 GHz
Ku	12–18 GHz
K	18–27 GHz
Ka	27–40 GHz
E	60–90 GHz

---

[31] `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



```
x=[-4:.1:4];
plot(x,exp(-(x-1).^2))
plot(x,exp(-(x+1).^2))
```

received power/noise

$$SNR = \frac{P_R}{k_B \cdot T \cdot B}$$

$\downarrow$ received power/noise density

$$C/N_0 = \frac{P_R}{k_B \cdot T} = SNR \cdot B$$

$\downarrow$ E_b energy/bit= carrier/spectral efficiency

$$E_b/N_0 = \frac{P_R/R}{k_B \cdot T} = C/N_0/R$$

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

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

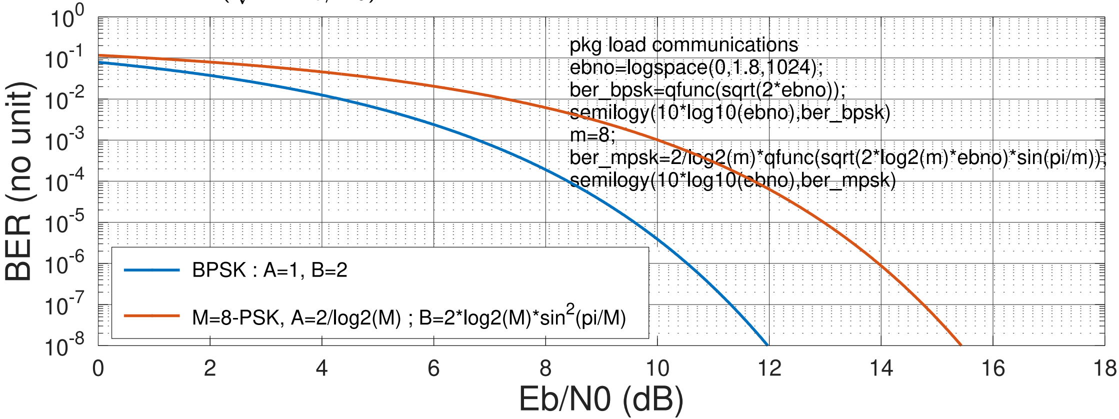
Abscissa: energy in each symbol to noise power ratio [32] E_b/N_0

---

[32] `https://noisecom.com/Portals/0/webinars/SN%20CN%20EbNo.pdf` for C/N v.s C/N_0 v.s E_b/N_0

Bit Error Rate (BER) [33]

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



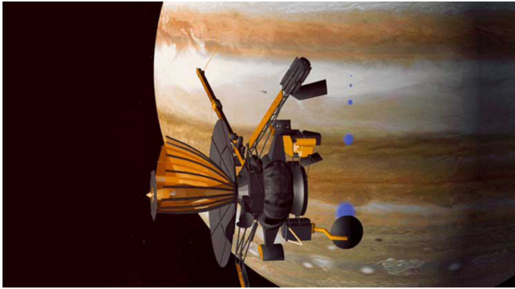
```
pkg load communications
ebno=logspace(0,1.8,1024);
ber_bpsk=qfunc(sqrt(2*ebno));
semilogy(10*log10(ebno),ber_bpsk)
m=8;
ber_mpsk=2/log2(m)*qfunc(sqrt(2*log2(m)*ebno)*sin(pi/m));
semilogy(10*log10(ebno),ber_mpsk)
```

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)

---

[33] J.-M Friedt, H. Boeglen, *Décodage d'images numéeriques 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.

[34] [35]

Analysis:

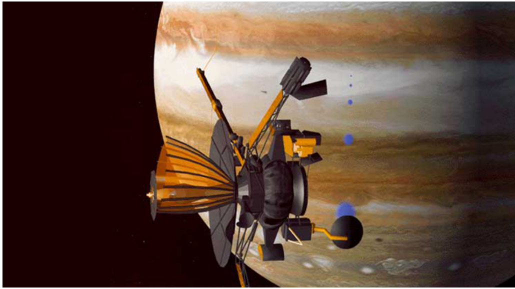
- X-band = 10 GHz $\Rightarrow G_1 = 10 \times \log_{10}\left(\left(\pi\frac{D}{\lambda}\right)^2\right) = 54$ dBi=250000
- lost antenna gain: 134000/250000 $\simeq$ 0.5 bps → 2.5 bps with 7 dBi LGA (https://www2.jpl.nasa.gov/galileo/faqgen.html)

---

[34] https://www2.jpl.nasa.gov/galileo/hga_fact.html

[35] https://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.

[...]

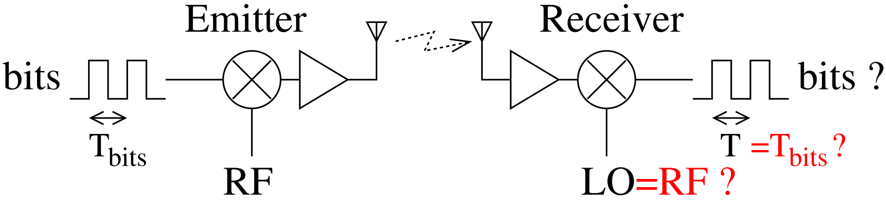
Without any new enhancements, the low-gain antenna's data transmission rate at Jupiter would be limited to only 8-16 bits per second (bps), compared to the high-gain's 134,400 bps.

34 35

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

Digital stream synchronization

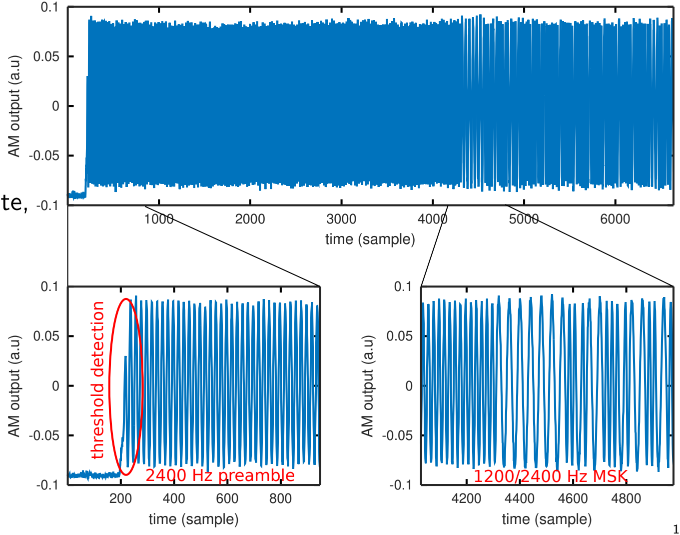
1. Synchronizing LO on RF: demodulation
2. Nominal datarate generated by a remote oscillator on the emitter ⇒ synchronization of the sampling times on the receiver
3. Startup sequence ("start bit"): ACARS 16-char long pre-key
4. Feedback on bit transitions (**clock recovery**)



ACARS and `gr-acars` [36]

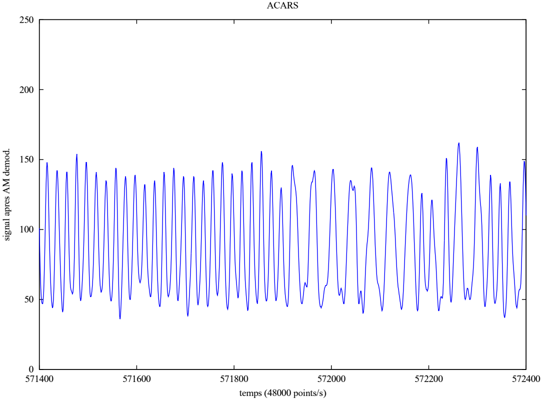
Aircraft Communications Addressing and Reporting System:

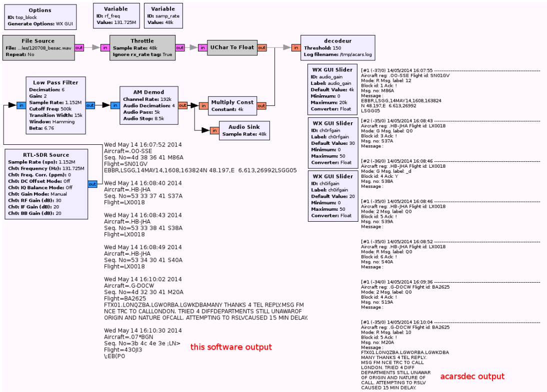
- Protocol developed in the 70s for communicating between plane pilots and ground
- Telemetry and free text
- As with all aeronautical protocols, AM mod.
- Information is MSK modulated at 2400 bps: 0=half 1200 Hz period, 1=full 2400 Hz period
- Differential encoding (0=keep bit state, 1=change bit state)
- 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 [37] ?





Options
ID: top_block
Generate Options: WX GUI

Variable
ID: rf_freq
Value: 131.725M

Variable
ID: samp_rate
Value: 48k

File Source
File: ...les/120708_besac.wav
Repeat: No

Throttle
Sample Rate: 48k
Ignore rx_rate tag: True

UChar To Float

decodeur
Threshold: 150
Log filename: /tmp/acars.log

Low Pass Filter
Decimation: 6
Gain: 2
Sample Rate: 1.152M
Cutoff Freq: 500k
Transition Width: 15k
Window: Hamming
Beta: 6.76

AM Demod
Channel Rate: 192k
Audio Decimation: 4
Audio Pass: 5k
Audio Stop: 8.5k

Multiply Const
Constant: 4k

Audio Sink
Sample Rate: 48k

RTL-SDR Source
Sample Rate (sps): 1.152M
Ch0: Frequency (Hz): 131.725M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 30
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

WX GUI Slider
ID: audio_gain
Label: audio_gain
Default Value: 4k
Minimum: 0
Maximum: 20k
Converter: Float

WX GUI Slider
ID: ch0rfgain
Label: ch0rfgain
Default Value: 30
Minimum: 0
Maximum: 50
Converter: Float

WX GUI Slider
ID: ch0ifgain
Label: ch0ifgain
Default Value: 20
Minimum: 0
Maximum: 50
Converter: Float

```
Wed May 14 16:07:52 2014
Aircraft=.OO-SSE
Seq. No=4d 38 36 41 M86A
Flight=SN01GV
EBBR,LSGG,14MAY14,1608,163824N 48.197,E  6.613,26992LSGG05

Wed May 14 16:08:40 2014
Aircraft=.HB-JHA
Seq. No=53 33 37 41 S37A
Flight=LX0018

Wed May 14 16:08:43 2014
Aircraft=.HB-JHA
Seq. No=53 33 38 41 S38A
Flight=LX0018

Wed May 14 16:08:49 2014
Aircraft=.HB-JHA
Seq. No=53 34 30 41 S40A
Flight=LX0018

Wed May 14 16:10:02 2014
Aircraft=.G-DOCW
Seq. No=4d 32 30 41 M20A
Flight=BA2625
FTX01.LONQZBA.LGWORBA.LGWKDBAMANY THANKS 4 TEL REPLY.MSG FM
NCE TRC TO CALLLONDON. TRIED 4 DIFFDEPARTMENTS STILL UNAWAROF
ORIGIN AND NATURE OFCALL. ATTEMPTING TO RSLVCAUSED 15 MIN DELAY.

Wed May 14 16:10:30 2014
Aircraft=.07*BGN
Seq. No=3b 4c 4e 3e ;LN>
Flight=430JI3
\;EB(PO
```

this software output

```
[#1 (-37/0) 14/05/2014 16:07:55 --------------------------------------------------
Aircraft reg: .OO-SSE Flight id: SN01GV
Mode: R Msg. label: 12
Block id: 1 Ack: !
Msg. no: M86A
Message :
EBBR,LSGG,14MAY14,1608,163824
N 48.197,E  6.613,26992
LSGG05
[#2 (-35/0) 14/05/2014 16:08:43 --------------------------------------------------
Aircraft reg: .HB-JHA Flight id: LX0018
Mode: G Msg. label: Q0
Block id: 3 Ack: !
Msg. no: S37A
Message :


[#2 (-36/0) 14/05/2014 16:08:46 --------------------------------------------------
Aircraft reg: .HB-JHA Flight id: LX0018
Mode: G Msg. label: _d
Block id: 4 Ack: Y
Msg. no: S38A
Message :


[#1 (-35/0) 14/05/2014 16:08:46 --------------------------------------------------
Aircraft reg: .HB-JHA Flight id: LX0018
Mode: 2 Msg. label: Q0
Block id: 5 Ack: !
Msg. no: S39A
Message :


[#1 (-35/0) 14/05/2014 16:08:52 --------------------------------------------------
Aircraft reg: .HB-JHA Flight id: LX0018
Mode: R Msg. label: Q0
Block id: 6 Ack: !
Msg. no: S40A
Message :


[#1 (-34/0) 14/05/2014 16:09:36 --------------------------------------------------
Aircraft reg: .G-DOCW Flight id: BA2625
Mode: 2 Msg. label: Q0
Block id: 4 Ack: !
Msg. no: S19A
Message :


[#1 (-35/0) 14/05/2014 16:10:04 --------------------------------------------------
Aircraft reg: .G-DOCW Flight id: BA2625
Mode: R Msg. label: 10
Block id: 5 Ack: !
Msg. no: M20A
Message :
FTX01.LONQZBA.LGWORBA.LGWKDBA
MANY THANKS 4 TEL REPLY.
MSG FM NCE TRC TO CALL
LONDON. TRIED 4 DIFF
DEPARTMENTS STILL UNAWAR
OF ORIGIN AND NATURE OF
CALL. ATTEMPTING TO RSLV
CAUSED 15 MIN DELAY.
```

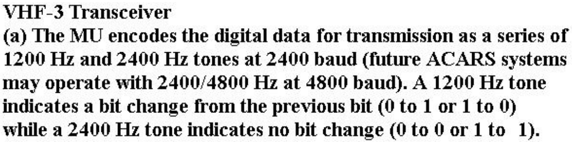
acarsdec output

---

[37] J.-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`

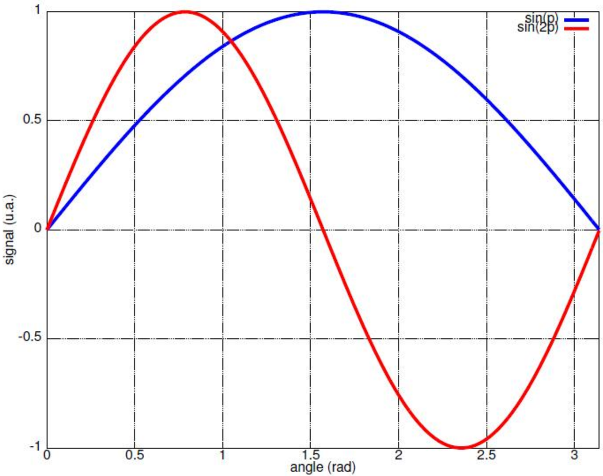
Digital mode demodulation: ACARS (prototyping)

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: convolution[38] with one period = bandpass filter

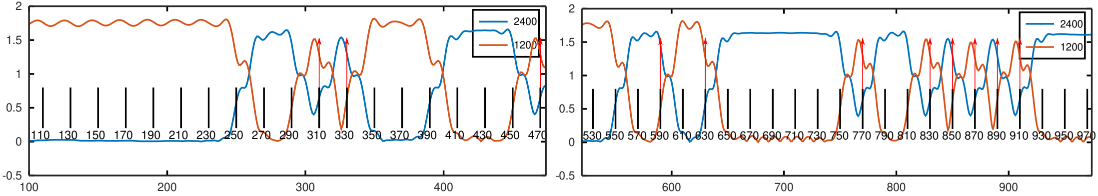
```
fs=48000;
bps=2400;
f2400=exp(j*2*pi*2400*[0:1/fs:1/bps*2]');f2400=f2400(1:end-1);
f1200=exp(j*2*pi*1200*[0:1/fs:1/bps*2]');f1200=f1200(1:end);
% s2400=abs(conv(y,f2400));s2400=s2400(200:end);  % convolution
% s1200=abs(conv(y,f1200));s1200=s1200(200:end);
% b=firls(256,[0 3400 4500 fs/2]*2/fs,[1 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')
```

---

[38] $FFT(x \circledast y) = FFT(x) \cdot FFT(y)$

Bit decoding

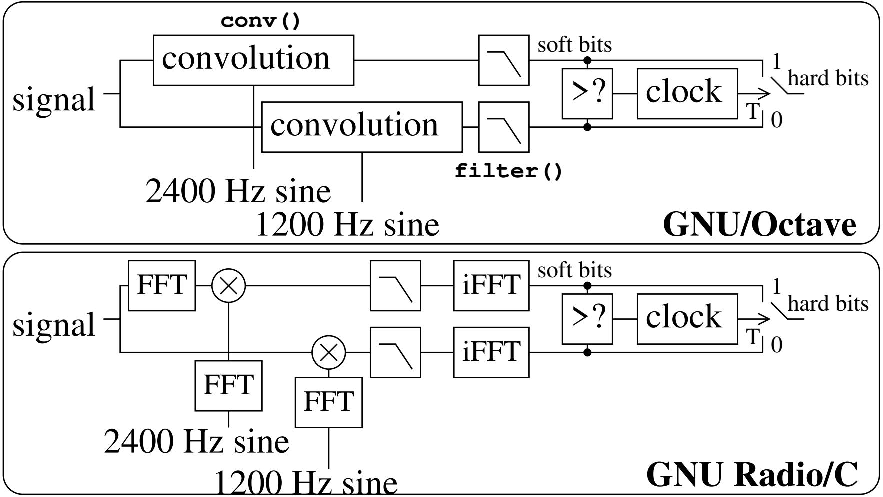
- Detect message as **rise in the standard deviation** of the received signal to avoid continuously running FFTs (might need refinement)
- Two tones define the two bit states: convolution with one period = bandpass filter
- Check every bit period the **most probable bit state**: soft bit → hard bit by 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.

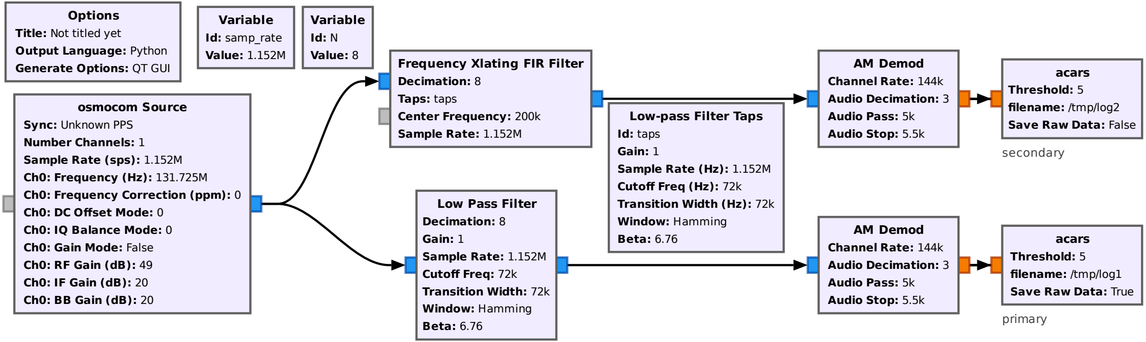
GNU Octave to GNU Radio

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



Bit decoding: FFT implementation

FFTW → GNU Radio FFT



Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 1.152M

Variable
Id: N
Value: 8

osmocom Source
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 1.152M
Ch0: Frequency (Hz): 131.725M
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 49
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: 200k
Sample Rate: 1.152M

Low Pass Filter
Decimation: 8
Gain: 1
Sample Rate: 1.152M
Cutoff Freq: 72k
Transition Width: 72k
Window: Hamming
Beta: 6.76

Low-pass Filter Taps
Id: taps
Gain: 1
Sample Rate (Hz): 1.152M
Cutoff Freq (Hz): 72k
Transition Width (Hz): 72k
Window: Hamming
Beta: 6.76

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log2
Save Raw Data: False

secondary

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log1
Save Raw Data: True

primary

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

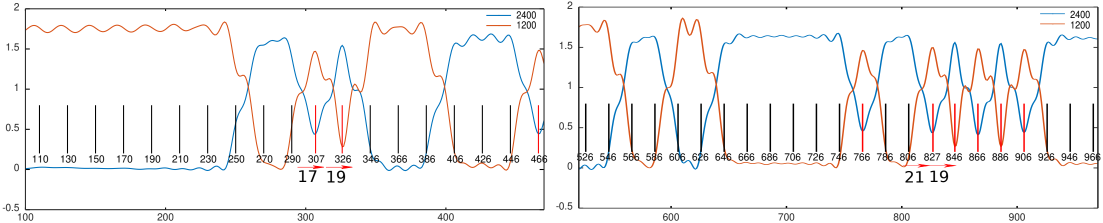
`http://www.fftw.org/fftw3_doc/Thread-safety.html` *planner routines share data (e.g. wisdom and trigonometric tables) between calls and plans … Neither strategy works, however, in the following situation. The "application" is structured as a set of "plugins" which are unaware of each other, and for 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

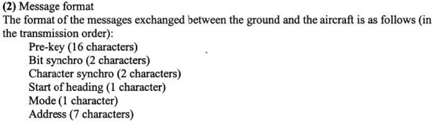


Black vertical lines indicate when successive hard bits are expected to be the same, **red vertical lines indicate when a hard bit is framed with two different hard-bit states and are hence used for clock synchronization**

Results: single receiver, multiple channels

Structure du message [39]:

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):

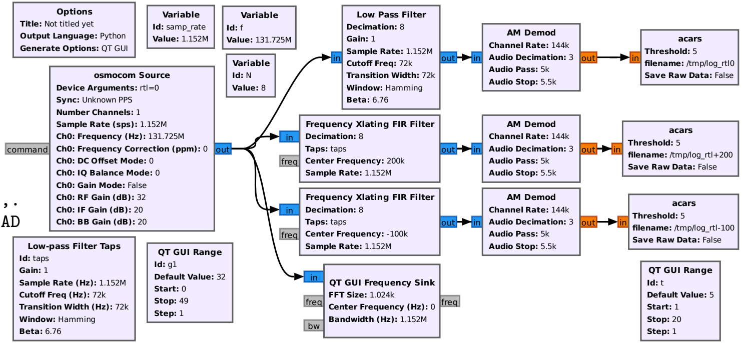
Pre-key (16 characters)
Bit synchro (2 characters)
Character synchro (2 characters)
Start of heading (1 character)
Mode (1 character)
Address (7 characters)

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/FCOAD
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
```



Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 1.152M

Variable
Id: f
Value: 131.725M

Variable
Id: N
Value: 8

osmocom Source
Device Arguments: rtl=0
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 1.152M
Ch0: Frequency (Hz): 131.725M
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 32
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Low-pass Filter Taps
Id: taps
Gain: 1
Sample Rate (Hz): 1.152M
Cutoff Freq (Hz): 72k
Transition Width (Hz): 72k
Window: Hamming
Beta: 6.76

QT GUI Range
Id: g1
Default Value: 32
Start: 0
Stop: 49
Step: 1

Low Pass Filter
Decimation: 8
Gain: 1
Sample Rate: 1.152M
Cutoff Freq: 72k
Transition Width: 72k
Window: Hamming
Beta: 6.76

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: 200k
Sample Rate: 1.152M

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: -100k
Sample Rate: 1.152M

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_rtl0
Save Raw Data: False

acars
Threshold: 5
filename: /tmp/log_rtl+200
Save Raw Data: False

acars
Threshold: 5
filename: /tmp/log_rtl-100
Save Raw Data: False

QT GUI Range
Id: t
Default Value: 5
Start: 1
Stop: 20
Step: 1

---

[39] http://www.scancat.com/Code-30_html_Source/acars.html

Sentences and parity bit

8th bit is odd-parity bit

Constant bit rate clock increment:

```
+*2.F-GZNOH12D23CAF4195#DFB 965  965000011 50 50
 60 62318319148145 07 08 00 26 -06-310-308-309-310-...
-239-239-239-310-310-311-309-310 016 017-392-391-026
 000 00Q.09..*POU/*: P/0 P/P_/P/ P/P_/PQP|uU
```

the messages ends up garbled

```
CRC:000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000110000110111001100111011
1011101101011
```

parity bit: desynchronization as non-0 parity bits.

Clock synchronization

```
+*2.F-GZNOH12D23CAF4195#DFB 965  965000011 50 50
 60 62318319148145 07 08 00 26 -06-310-308-309-310-...
-239-239-239-310-310-311-309-310 016 017-392-391-026
 000 001109113005029 000 000 000 000 00102
```

full message properly decoded

```
CRC:000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000
0000000000011
```

parity bit state

---

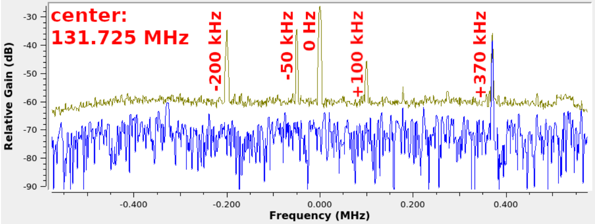
42 sentences collected over 6 minutes from a location a few kilometers from Roissy Charles de Gaulle airport in Paris: 15 would have been corrupted if decoded without clock synchronization

⇒ **35% success rate improvement**

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



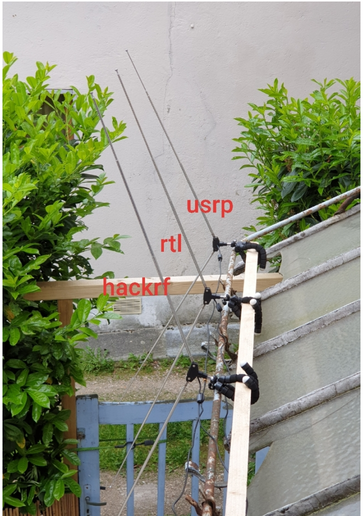
↓ 131.825 MHz

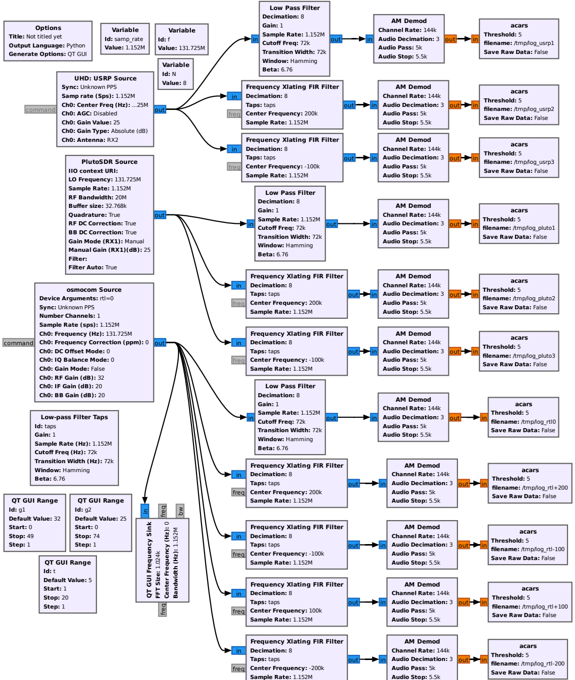
```
Fri May  1 12:09:16 2020
Aircraft=..N22UB
Seq. No=53 30 32 41 S02A
Flight=ZD0001
ETX

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 (T. Lavarenne)

Each source connected to a different antenna





Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 1.152M

Variable
Id: f
Value: 131.725M

Variable
Id: N
Value: 8

UHD: USRP Source
Sync: Unknown PPS
Samp rate (Sps): 1.152M
Ch0: Center Freq (Hz): ...25M
Ch0: AGC: Disabled
Ch0: Gain Value: 25
Ch0: Gain Type: Absolute (dB)
Ch0: Antenna: RX2

PlutoSDR Source
IIO context URI:
LO Frequency: 131.725M
Sample Rate: 1.152M
RF Bandwidth: 20M
Buffer size: 32.768k
Quadrature: True
RF DC Correction: True
BB DC Correction: True
Gain Mode (RX1): Manual
Manual Gain (RX1)(dB): 25
Filter:
Filter Auto: True

osmocom Source
Device Arguments: rtl=0
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 1.152M
Ch0: Frequency (Hz): 131.725M
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 32
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Low-pass Filter Taps
Id: taps
Gain: 1
Sample Rate (Hz): 1.152M
Cutoff Freq (Hz): 72k
Transition Width (Hz): 72k
Window: Hamming
Beta: 6.76

QT GUI Range
Id: g1
Default Value: 32
Start: 0
Stop: 49
Step: 1

QT GUI Range
Id: g2
Default Value: 25
Start: 0
Stop: 74
Step: 1

QT GUI Range
Id: t
Default Value: 5
Start: 1
Stop: 20
Step: 1

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 1.152M

Low Pass Filter
Decimation: 8
Gain: 1
Sample Rate: 1.152M
Cutoff Freq: 72k
Transition Width: 72k
Window: Hamming
Beta: 6.76

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_usrp1
Save Raw Data: False

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: 200k
Sample Rate: 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_usrp2
Save Raw Data: False

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: -100k
Sample Rate: 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_usrp3
Save Raw Data: False

Low Pass Filter
Decimation: 8
Gain: 1
Sample Rate: 1.152M
Cutoff Freq: 72k
Transition Width: 72k
Window: Hamming
Beta: 6.76

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_pluto1
Save Raw Data: False

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: 200k
Sample Rate: 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_pluto2
Save Raw Data: False

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: -100k
Sample Rate: 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_pluto3
Save Raw Data: False

Low Pass Filter
Decimation: 8
Gain: 1
Sample Rate: 1.152M
Cutoff Freq: 72k
Transition Width: 72k
Window: Hamming
Beta: 6.76

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_rtl0
Save Raw Data: False

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: 200k
Sample Rate: 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_rtl+200
Save Raw Data: False

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: -100k
Sample Rate: 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_rtl-100
Save Raw Data: False

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: 100k
Sample Rate: 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

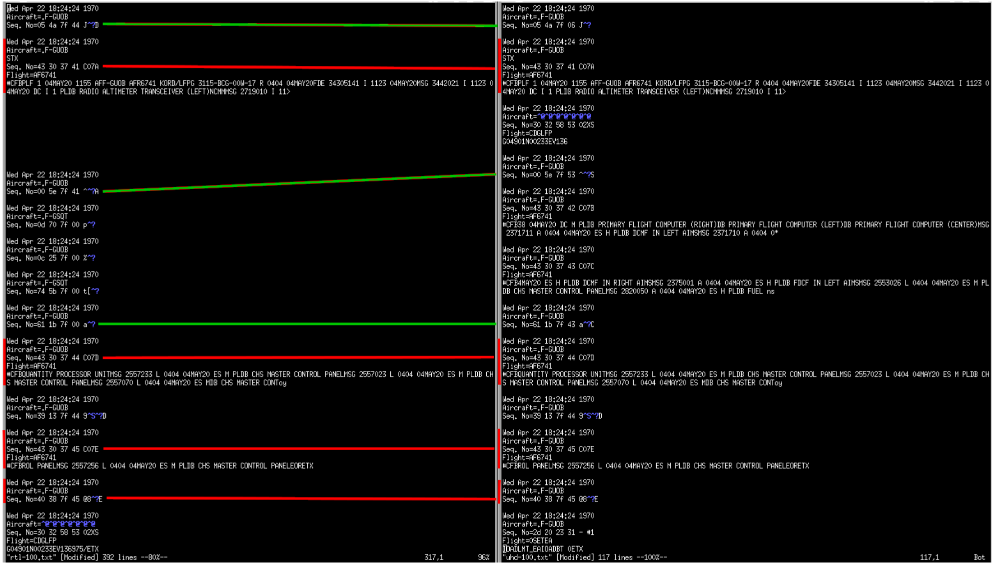
acars
Threshold: 5
filename: /tmp/log_rtl+100
Save Raw Data: False

Frequency Xlating FIR Filter
Decimation: 8
Taps: taps
Center Frequency: -200k
Sample Rate: 1.152M

AM Demod
Channel Rate: 144k
Audio Decimation: 3
Audio Pass: 5k
Audio Stop: 5.5k

acars
Threshold: 5
filename: /tmp/log_rtl-200
Save Raw Data: False

Results: multiple receivers



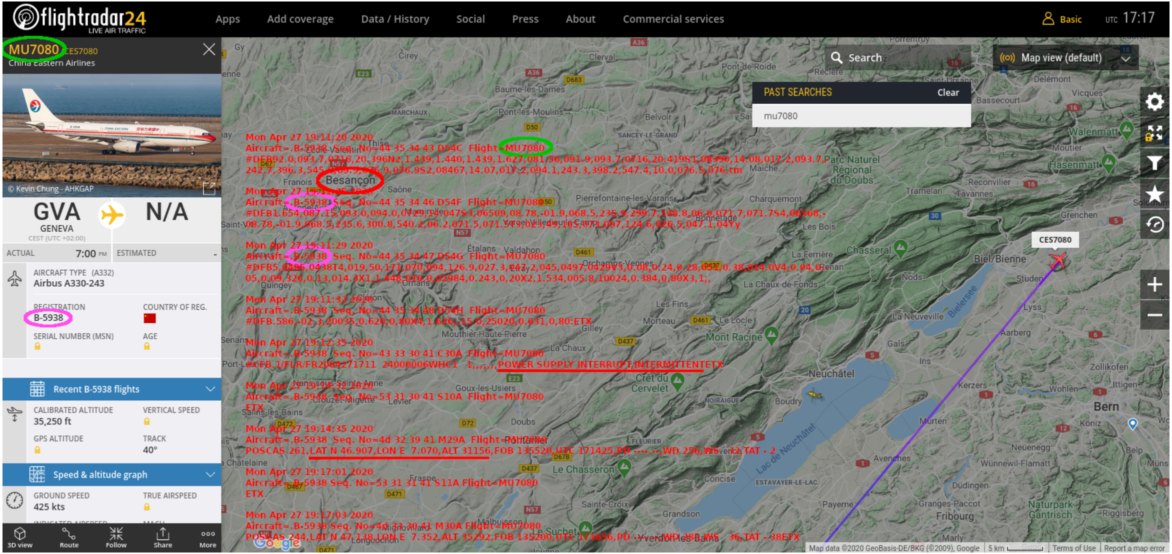
```
Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
Seq. No=05 4a 7f 44 J^?D

Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
STX
Seq. No=43 30 37 41 C07A
Flight=AF6741
#CFBPLF 1 04MAY20 1155 AFF-GUOB AFR6741 KORD/LFPG 3115-BCG-00W-17 R 0404 04MAY20FDE 34305141 I 1123 04MAY20MSG 3442021 I 1123 0
4MAY20 DC I 1 PLDB RADIO ALTIMETER TRANSCEIVER (LEFT)NCMMMSG 2719010 I 11>









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

Wed Apr 22 18:24:24 1970
Aircraft=.F-GSQT
Seq. No=0d 70 7f 00 p^?

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

Wed Apr 22 18:24:24 1970
Aircraft=.F-GSQT
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. No=43 30 37 44 C07D
Flight=AF6741
#CFBQUANTITY PROCESSOR UNITMSG 2557233 L 0404 04MAY20 ES M PLDB CHS MASTER CONTROL PANELMSG 2557023 L 0404 04MAY20 ES M PLDB CH
S MASTER CONTROL PANELMSG 2557070 L 0404 04MAY20 ES MDB CHS MASTER CONToy

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

Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
Seq. No=43 30 37 45 C07E
Flight=AF6741
#CFBROL PANELMSG 2557256 L 0404 04MAY20 ES M PLDB CHS MASTER CONTROL PANELEORETX

Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
Seq. No=40 38 7f 45 @8^?E

Wed Apr 22 18:24:24 1970
Aircraft=^@^@^@^@^@^@^@
Seq. No=30 32 58 53 02XS
Flight=CDGLFP
G04901N00233EV136975/ETX
"rtl-100.txt" [Modified] 392 lines --80%--                                              317,1         96%
```

```
Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
Seq. No=05 4a 7f 06 J^?

Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
STX
Seq. No=43 30 37 41 C07A
Flight=AF6741
#CFBPLF 1 04MAY20 1155 AFF-GUOB AFR6741 KORD/LFPG 3115-BCG-00W-17 R 0404 04MAY20FDE 34305141 I 1123 04MAY20MSG 3442021 I 1123 0
4MAY20 DC I 1 PLDB RADIO ALTIMETER TRANSCEIVER (LEFT)NCMMMSG 2719010 I 11>

Wed Apr 22 18:24:24 1970
Aircraft=^@^@^@^@^@^@^@
Seq. No=30 32 58 53 02XS
Flight=CDGLFP
G04901N00233EV136

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. No=43 30 37 42 C07B
Flight=AF6741
#CFB38 04MAY20 DC M PLDB PRIMARY FLIGHT COMPUTER (RIGHT)DB PRIMARY FLIGHT COMPUTER (LEFT)DB PRIMARY FLIGHT COMPUTER (CENTER)MSG
 2371711 A 0404 04MAY20 ES H PLDB DCMF IN LEFT AIMSMSG 2371710 A 0404 0*

Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
Seq. No=43 30 37 43 C07C
Flight=AF6741
#CFB4MAY20 ES H PLDB DCMF IN RIGHT AIMSMSG 2375001 A 0404 04MAY20 ES H PLDB FDCF IN LEFT AIMSMSG 2553026 L 0404 04MAY20 ES M PL
DB CHS MASTER CONTROL PANELMSG 2820050 A 0404 04MAY20 ES H PLDB FUEL ns

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. No=43 30 37 44 C07D
Flight=AF6741
#CFBQUANTITY PROCESSOR UNITMSG 2557233 L 0404 04MAY20 ES M PLDB CHS MASTER CONTROL PANELMSG 2557023 L 0404 04MAY20 ES M PLDB CH
S MASTER CONTROL PANELMSG 2557070 L 0404 04MAY20 ES MDB CHS MASTER CONToy

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

Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
Seq. No=43 30 37 45 C07E
Flight=AF6741
#CFBROL PANELMSG 2557256 L 0404 04MAY20 ES M PLDB CHS MASTER CONTROL PANELEORETX

Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
Seq. No=40 38 7f 45 @8^?E

Wed Apr 22 18:24:24 1970
Aircraft=.F-GUOB
Seq. No=2d 20 23 31 - #1
Flight=0SETEA
DADLMT_EAIOADBT 0ETX
"uhd-100.txt" [Modified] 117 lines --100%--                                             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

Results: decoded sentences



MU7080 CES7080
China Eastern Airlines

© Kevin Chung - AHKGAP

GVA GENEVA CEST (UTC +02:00)
N/A

ACTUAL 7:00 PM
ESTIMATED -

AIRCRAFT TYPE (A332): Airbus A330-243
REGISTRATION: B-5938
COUNTRY OF REG.
SERIAL NUMBER (MSN)
AGE

Recent B-5938 flights

CALIBRATED ALTITUDE: 35,250 ft
VERTICAL SPEED
GPS ALTITUDE
TRACK: 40°

Speed & altitude graph

GROUND SPEED: 425 kts
TRUE AIRSPEED

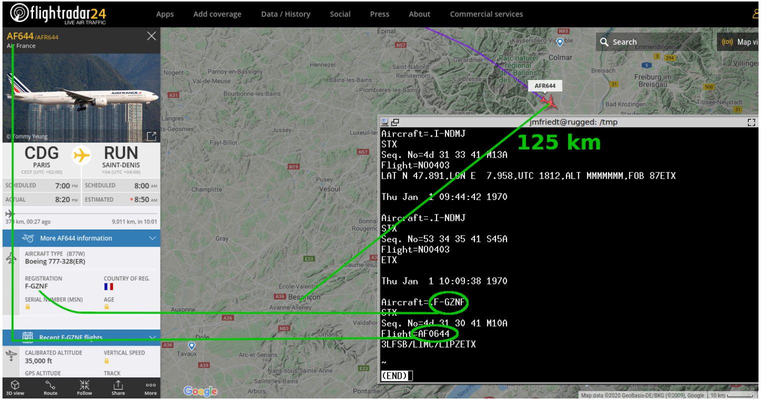
```
Mon Apr 27 19:11:20 2020
Aircraft=.B-5938  Seq. No=44 35 34 43 D54C  Flight=MU7080
#DFB92.0,093.7,0716,20.396N2,1.439,1.440,1.439,1.627,081.56,091.9,093.7,0716,20.419S1,08390,14.08,017.2,093.7,
242.7,396.3,545.?,0?3.9,076.9,076.9S2,08467,14.07,017.2,094.1,243.3,398.2,547.4,10.0,076.5,076.tm

Mon Apr 27 19:11:25 2020
Aircraft=.B-5938  Seq. No=44 35 34 46 D54F  Flight=MU7080
#DFB1.654,087.15,093.0,094.0,0729,14.047S3,06509,08.78,-01.9,068.5,235.9,299.7,538.8,06.0,071.7,071.7S4,06568,-
08.78,-01.9,068.5,235.6,300.8,540.2,06.2,071.5,071.5T3,023,49,165,073,097,124.6,026.5,047.1,04Yy

Mon Apr 27 19:11:29 2020
Aircraft=.B-5938  Seq. No=44 35 34 47 D54G  Flight=MU7080
#DFB5,0495,0438T4,019,50,171,070,094,126.9,027.3,047.2,045,0497,0429V3,0.08,0.24,0.28,051,0.38,014.0V4,0.04,0.
05,0.09,326,0.13,014.3X1,1.448,016.0,02984,0.243,0,20X2,1.534,005.8,10024,0.384,0,80X3,1;,

Mon Apr 27 19:11:32 2020
Aircraft=.B-5938  Seq. No=44 35 34 48 D54H  Flight=MU7080
#DFB.586,-02.3,20035,0.620,0,80X4,1.626,-15.0,25020,0.631,0,80:ETX

Mon Apr 27 19:12:35 2020
Aircraft=.B-5938  Seq. No=43 33 30 41 C30A  Flight=MU7080
#CFB.1/FLR/FR2004271711  24000006WHC1   1,,,,,,,POWER SUPPLY INTERRUPT,INTERMITTENTETX

Mon Apr 27 19:14:32 2020
Aircraft=.B-5938  Seq. No=53 31 30 41 S10A  Flight=MU7080
ETX

Mon Apr 27 19:14:35 2020
Aircraft=.B-5938  Seq. No=4d 32 39 41 M29A  Flight=MU7080
POSCAS 261,LAT N 46.907,LON E  7.070,ALT 31156,FOB 135520,UTC 171425,PD ----.--,WD 256,WS  32,TAT - 2

Mon Apr 27 19:17:01 2020
Aircraft=.B-5938 Seq. No=53 31 31 41 S11A Flight=MU7080
ETX

Mon Apr 27 19:17:03 2020
Aircraft=.B-5938 Seq. No=4d 33 30 41 M30A Flight=MU7080
POSCAS 244,LAT N 47.138,LON E  7.352,ALT 35292,FOB 135200,UTC 171656,PD ----.--,WD 285,WS  36,TAT - 38ETX
```

consistent with ADS-B messages collected from `flightradar24.com`

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 !)

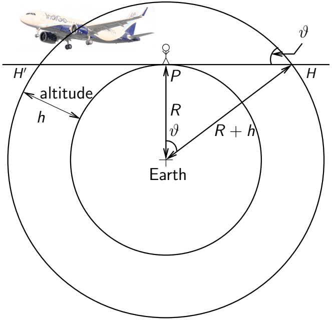


AF644 /AFR644
Air France
© Tommy Yeung
CDG PARIS CEST (UTC +02:00)
RUN SAINT-DENIS +04 (UTC +04:00)
SCHEDULED 7:00 PM
SCHEDULED 8:00 AM
ACTUAL 8:20 PM
ESTIMATED 8:50 AM
370 km, 00:27 ago
9,011 km, in 10:01
More AF644 information
AIRCRAFT TYPE (B77W)
Boeing 777-328(ER)
REGISTRATION
F-GZNF
COUNTRY OF REG.
SERIAL NUMBER (MSN)
AGE
Recent F-GZNF flights
CALIBRATED ALTITUDE
35,000 ft
VERTICAL SPEED
GPS ALTITUDE
TRACK

jmfriedt@rugged: /tmp

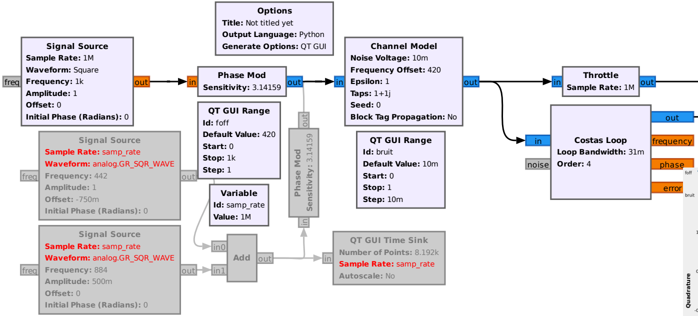
```
Aircraft=.I-NDMJ
STX
Seq. No=4d 31 33 41 M13A
Flight=N00403
LAT N 47.891,LON E  7.958,UTC 1812,ALT MMMMMMM,FOB 87ETX

Thu Jan  1 09:44:42 1970

Aircraft=.I-NDMJ
STX
Seq. No=53 34 35 41 S45A
Flight=N00403
ETX

Thu Jan  1 10:09:38 1970

Aircraft=.F-GZNF
STX
Seq. No=4d 31 30 41 M10A
Flight=AF0644
3LFSB/LIMC/LIPZETX

~
(END)
```

Map data ©2020 GeoBasis-DE/BKG (©2009), Google 10 km



Constellation diagram

- I/Q coefficients represented in the **complex plane** (e.g. BPSK, QPSK [40])
- 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



Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Signal Source
Sample Rate: 1M
Waveform: Square
Frequency: 1k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

Phase Mod
Sensitivity: 3.14159

Channel Model
Noise Voltage: 10m
Frequency Offset: 420
Epsilon: 1
Taps: 1+1j
Seed: 0
Block Tag Propagation: No

Throttle
Sample Rate: 1M

QT GUI Constellation Sink
Number of Points: 8.192k
Autoscale: No

QT GUI Range
Id: foff
Default Value: 420
Start: 0
Stop: 1k
Step: 1

QT GUI Range
Id: bruit
Default Value: 10m
Start: 0
Stop: 1
Step: 10m

Costas Loop
Loop Bandwidth: 31m
Order: 4

Variable
Id: samp_rate
Value: 1M

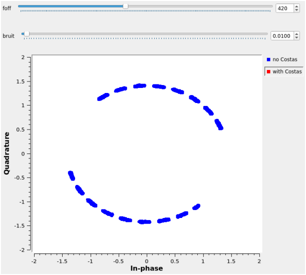
Signal Source
Sample Rate: samp_rate
Waveform: analog.GR_SQR_WAVE
Frequency: 442
Amplitude: 1
Offset: -750m
Initial Phase (Radians): 0

Signal Source
Sample Rate: samp_rate
Waveform: analog.GR_SQR_WAVE
Frequency: 884
Amplitude: 500m
Offset: 0
Initial Phase (Radians): 0

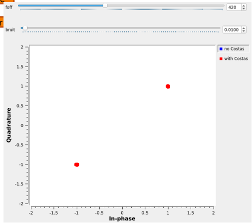
Add

Phase Mod
Sensitivity: 3.14159

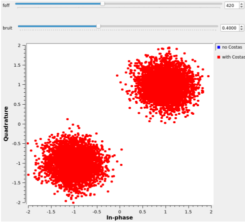
QT GUI Time Sink
Number of Points: 8.192k
Sample Rate: samp_rate
Autoscale: No



BPSK: no Costas



Costas



Costas+noise

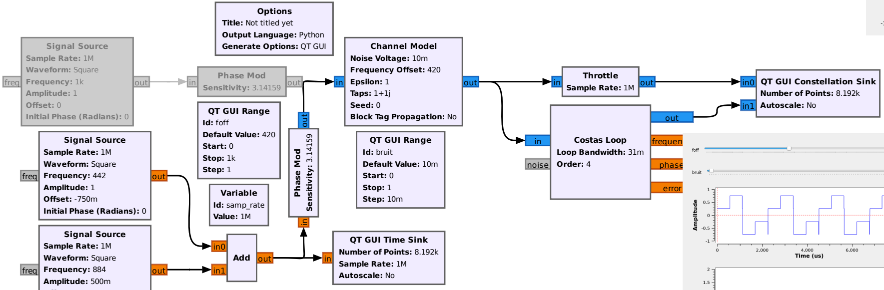
---

[41] Balint's 4th tutorial on YouTube `https://www.youtube.com/watch?v=JMEyN_lvaiE` &

demo: slides/constellation/constellation_analogique.grc

Constellation diagram

- I/Q coefficients represented in the **complex plane** (e.g. BPSK, QPSK [42])
- 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



Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Signal Source
Sample Rate: 1M
Waveform: Square
Frequency: 1k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0

Phase Mod
Sensitivity: 3.14159

QT GUI Range
Id: foff
Default Value: 420
Start: 0
Stop: 1k
Step: 1

Variable
Id: samp_rate
Value: 1M

Phase Mod
Sensitivity: 3.14159

Signal Source
Sample Rate: 1M
Waveform: Square
Frequency: 442
Amplitude: 1
Offset: -750m
Initial Phase (Radians): 0

Signal Source
Sample Rate: 1M
Waveform: Square
Frequency: 884
Amplitude: 500m
Offset: 0
Initial Phase (Radians): 0

Add

QT GUI Time Sink
Number of Points: 8.192k
Sample Rate: 1M
Autoscale: No

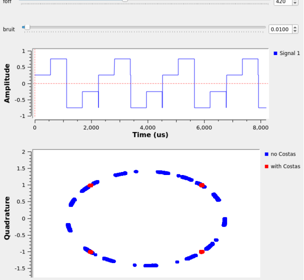
Channel Model
Noise Voltage: 10m
Frequency Offset: 420
Epsilon: 1
Taps: 1+1j
Seed: 0
Block Tag Propagation: No

QT GUI Range
Id: bruit
Default Value: 10m
Start: 0
Stop: 1
Step: 10m

Throttle
Sample Rate: 1M

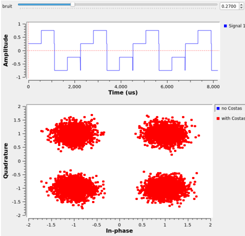
Costas Loop
Loop Bandwidth: 31m
Order: 4

QT GUI Constellation Sink
Number of Points: 8.192k
Autoscale: No



QPSK: no Costas

Costas



Costas+noise

---

[43] Balint's 4th tutorial on YouTube

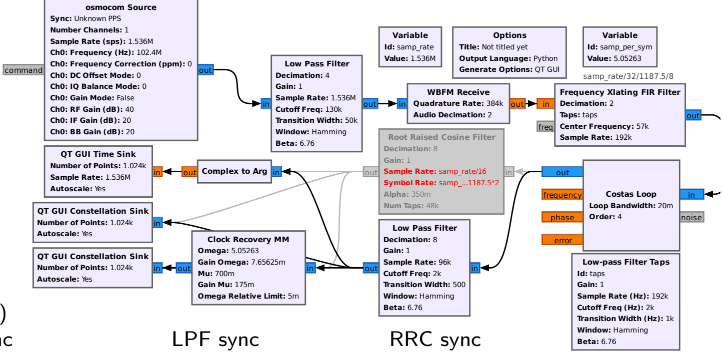
`https://www.youtube.com/watch?v=JMEyN_lvaiE`

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`)



osmocom Source
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 1.536M
Ch0: Frequency (Hz): 102.4M
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 40
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Variable
Id: samp_rate
Value: 1.536M

Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_per_sym
Value: 5.05263
samp_rate/32/1187.5/8

Low Pass Filter
Decimation: 4
Gain: 1
Sample Rate: 1.536M
Cutoff Freq: 130k
Transition Width: 50k
Window: Hamming
Beta: 6.76

WBFM Receive
Quadrature Rate: 384k
Audio Decimation: 2

Frequency Xlating FIR Filter
Decimation: 2
Taps: taps
Center Frequency: 57k
Sample Rate: 192k

QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 1.536M
Autoscale: Yes

Complex to Arg

Root Raised Cosine Filter
Decimation: 8
Gain: 1
Sample Rate: samp_rate/16
Symbol Rate: samp_...1187.5*2
Alpha: 350m
Num Taps: 48k

Costas Loop
Loop Bandwidth: 20m
Order: 4

QT GUI Constellation Sink
Number of Points: 1.024k
Autoscale: Yes

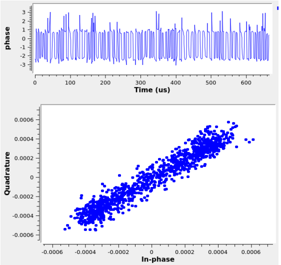
Clock Recovery MM
Omega: 5.05263
Gain Omega: 7.65625m
Mu: 700m
Gain Mu: 175m
Omega Relative Limit: 5m

Low Pass Filter
Decimation: 8
Gain: 1
Sample Rate: 96k
Cutoff Freq: 2k
Transition Width: 500
Window: Hamming
Beta: 6.76

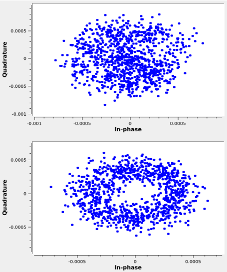
QT GUI Constellation Sink
Number of Points: 1.024k
Autoscale: Yes

Low-pass Filter Taps
Id: taps
Gain: 1
Sample Rate (Hz): 192k
Cutoff Freq (Hz): 2k
Transition Width (Hz): 1k
Window: Hamming
Beta: 6.76

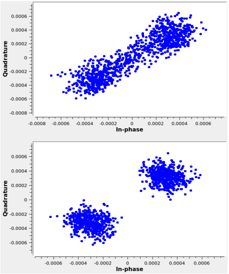
No sync



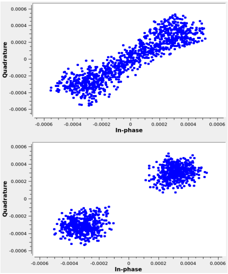
poor sync

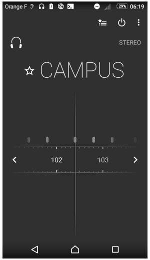


LPF sync



RRC sync



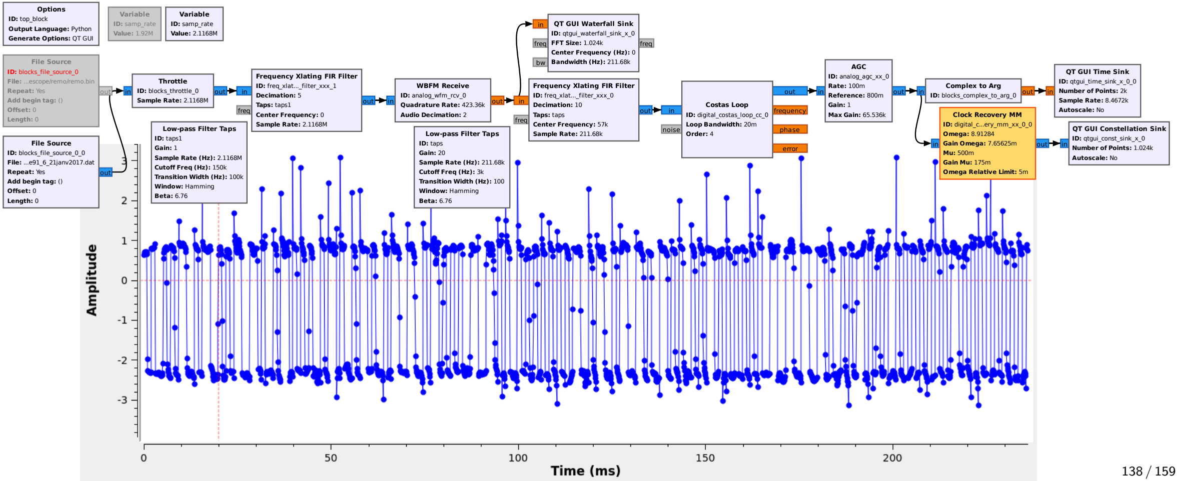


J.-M Friedt, *Radio Data System (RDS) – analyse du canal numérique transmis par les stations radio FM commerciales, introduction aux codes correcteurs d'erreur*, GNU/Linux Magazine France 204 (Mai 2017)

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** ?



Options
ID: top_block
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 1.92M

Variable
ID: samp_rate
Value: 2.1168M

File Source
ID: blocks_file_source_0
File: ...escope/remo/remo.bin
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0

File Source
ID: blocks_file_source_0_0
File: ...e91_6_21janv2017.dat
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0

Throttle
ID: blocks_throttle_0
Sample Rate: 2.1168M

Low-pass Filter Taps
ID: taps1
Gain: 1
Sample Rate (Hz): 2.1168M
Cutoff Freq (Hz): 150k
Transition Width (Hz): 100k
Window: Hamming
Beta: 6.76

Frequency Xlating FIR Filter
ID: freq_xlat..._filter_xxx_1
Decimation: 5
Taps: taps1
Center Frequency: 0
Sample Rate: 2.1168M

WBFM Receive
ID: analog_wfm_rcv_0
Quadrature Rate: 423.36k
Audio Decimation: 2

Low-pass Filter Taps
ID: taps
Gain: 20
Sample Rate (Hz): 211.68k
Cutoff Freq (Hz): 3k
Transition Width (Hz): 100
Window: Hamming
Beta: 6.76

QT GUI Waterfall Sink
ID: qtgui_waterfall_sink_x_0
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 211.68k

Frequency Xlating FIR Filter
ID: freq_xlat..._filter_xxx_0
Decimation: 10
Taps: taps
Center Frequency: 57k
Sample Rate: 211.68k

Costas Loop
ID: digital_costas_loop_cc_0
Loop Bandwidth: 20m
Order: 4

AGC
ID: analog_agc_xx_0
Rate: 100m
Reference: 800m
Gain: 1
Max Gain: 65.536k

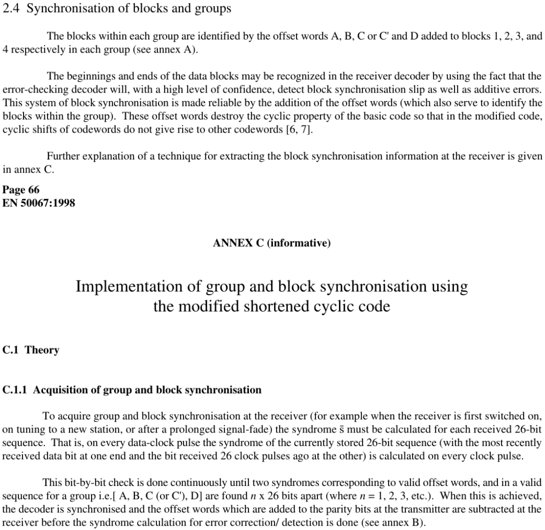
Complex to Arg
ID: blocks_complex_to_arg_0

Clock Recovery MM
ID: digital_c...ery_mm_xx_0_0
Omega: 8.91284
Gain Omega: 7.65625m
Mu: 500m
Gain Mu: 175m
Omega Relative Limit: 5m

QT GUI Time Sink
ID: qtgui_time_sink_x_0_0
Number of Points: 2k
Sample Rate: 8.4672k
Autoscale: No

QT GUI Constellation Sink
ID: qtgui_const_sink_x_0
Number of Points: 1.024k
Autoscale: No

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

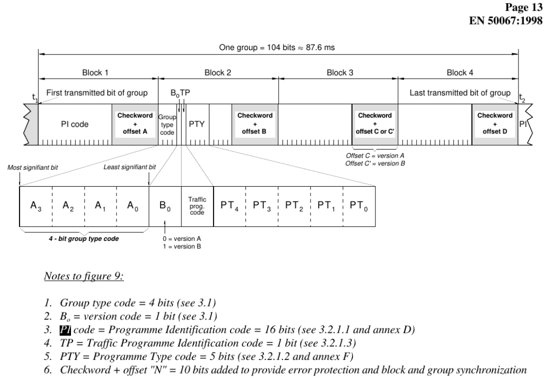
C.1.1 Acquisition of group and block synchronisation

To acquire group and block synchronisation at the receiver (for example when the receiver is first switched on, 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* x 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 solution[44]: use the error correction code on all successive bits received until the consistent CRC is reached …

↓ … but all sentences start with the PI code unique[45]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. *B_o = 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*

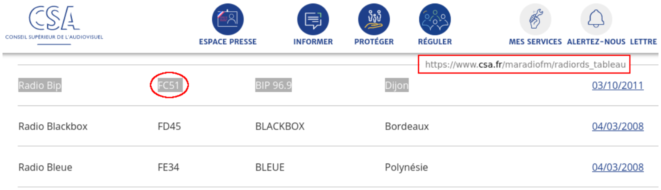
---

[45] *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

[45] List of Access Codes: `https://www.csa.fr/maradiofm/radiords_tableau`, 0xFC3A for Radio Campus

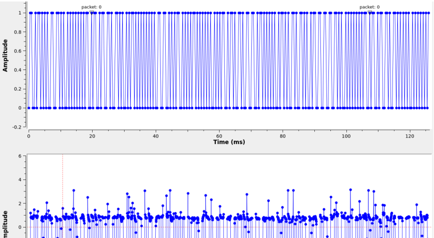
RDS access code: stream header

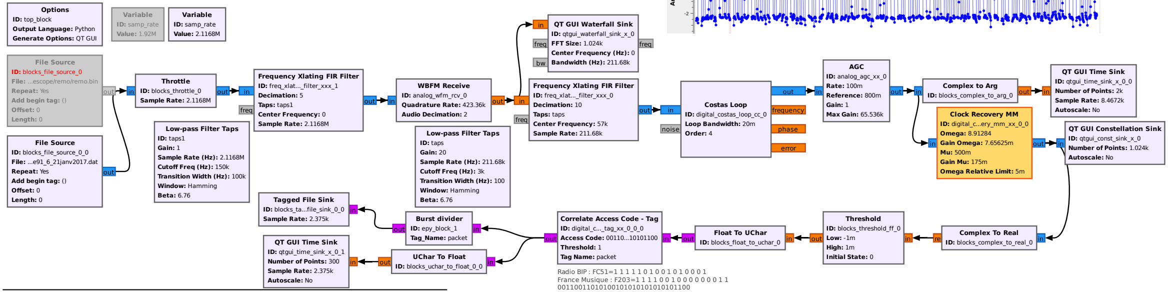
PI is documented as “Program Indentification”: differential Manchester encoding to maximize transisitons (0 → 00 or 11, and 1 → 01 or 10) so that searching (**correlation**) this encoded pattern in the bit sequence must lead to the sentence beginning [46]



https://www.csa.fr/maradiofm/radiords_tableau

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





Options
ID: top_block
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 1.92M

Variable
ID: samp_rate
Value: 2.1168M

File Source
ID: blocks_file_source_0
File: ...escope/remo/remo.bin
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0

File Source
ID: blocks_file_source_0_0
File: ...e91_6_21janv2017.dat
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0

Throttle
ID: blocks_throttle_0
Sample Rate: 2.1168M

Low-pass Filter Taps
ID: taps1
Gain: 1
Sample Rate (Hz): 2.1168M
Cutoff Freq (Hz): 150k
Transition Width (Hz): 100k
Window: Hamming
Beta: 6.76

Frequency Xlating FIR Filter
ID: freq_xlat..._filter_xxx_1
Decimation: 5
Taps: taps1
Center Frequency: 0
Sample Rate: 2.1168M

WBFM Receive
ID: analog_wfm_rcv_0
Quadrature Rate: 423.36k
Audio Decimation: 2

Low-pass Filter Taps
ID: taps
Gain: 20
Sample Rate (Hz): 211.68k
Cutoff Freq (Hz): 3k
Transition Width (Hz): 100
Window: Hamming
Beta: 6.76

QT GUI Waterfall Sink
ID: qtgui_waterfall_sink_x_0
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 211.68k

Frequency Xlating FIR Filter
ID: freq_xlat..._filter_xxx_0
Decimation: 10
Taps: taps
Center Frequency: 57k
Sample Rate: 211.68k

Costas Loop
ID: digital_costas_loop_cc_0
Loop Bandwidth: 20m
Order: 4

AGC
ID: analog_agc_xx_0
Rate: 100m
Reference: 800m
Gain: 1
Max Gain: 65.536k

Complex to Arg
ID: blocks_complex_to_arg_0

QT GUI Time Sink
ID: qtgui_time_sink_x_0_0
Number of Points: 2k
Sample Rate: 8.4672k
Autoscale: No

Clock Recovery MM
ID: digital_c...ery_mm_xx_0_0
Omega: 8.91284
Gain Omega: 7.65625m
Mu: 500m
Gain Mu: 175m
Omega Relative Limit: 5m

QT GUI Constellation Sink
ID: qtgui_const_sink_x_0
Number of Points: 1.024k
Autoscale: No

Complex To Real
ID: blocks_complex_to_real_0

Threshold
ID: blocks_threshold_ff_0
Low: -1m
High: 1m
Initial State: 0

Float To UChar
ID: blocks_float_to_uchar_0

Correlate Access Code - Tag
ID: digital_c..._tag_xx_0_0_0
Access Code: 00110...10101100
Threshold: 1
Tag Name: packet

Burst divider
ID: epy_block_1
Tag_Name: packet

Tagged File Sink
ID: blocks_ta...file_sink_0_0
Sample Rate: 2.375k

UChar To Float
ID: blocks_uchar_to_float_0_0

QT GUI Time Sink
ID: qtgui_time_sink_x_0_1
Number of Points: 300
Sample Rate: 2.375k
Autoscale: No

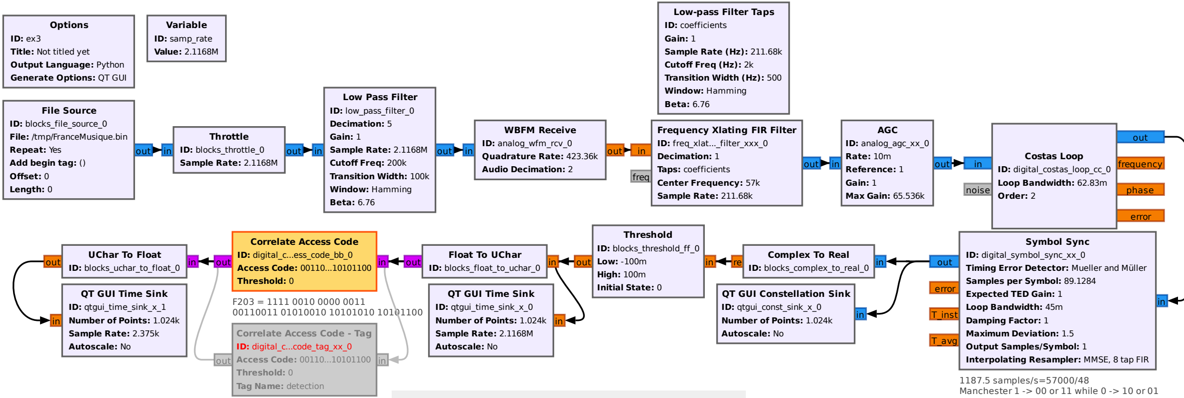
Radio BIP : FC51=1 1 1 1 1 0 1 0 0 1 0 1 0 0 0 1
France Musique : F203=1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 1
00110011010100101010101010101100

---

[46] Processing sequence by T. Lavarenne, Lycée Jean Rostand, Villepinte

GNU Radio $\geq$ 3.9: Symbol Sync block

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



Options
ID: ex3
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 2.1168M

File Source
ID: blocks_file_source_0
File: /tmp/FranceMusique.bin
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0

Throttle
ID: blocks_throttle_0
Sample Rate: 2.1168M

Low Pass Filter
ID: low_pass_filter_0
Decimation: 5
Gain: 1
Sample Rate: 2.1168M
Cutoff Freq: 200k
Transition Width: 100k
Window: Hamming
Beta: 6.76

WBFM Receive
ID: analog_wfm_rcv_0
Quadrature Rate: 423.36k
Audio Decimation: 2

Low-pass Filter Taps
ID: coefficients
Gain: 1
Sample Rate (Hz): 211.68k
Cutoff Freq (Hz): 2k
Transition Width (Hz): 500
Window: Hamming
Beta: 6.76

Frequency Xlating FIR Filter
ID: freq_xlat..._filter_xxx_0
Decimation: 1
Taps: coefficients
Center Frequency: 57k
Sample Rate: 211.68k

AGC
ID: analog_agc_xx_0
Rate: 10m
Reference: 1
Gain: 1
Max Gain: 65.536k

Costas Loop
ID: digital_costas_loop_cc_0
Loop Bandwidth: 62.83m
Order: 2

Symbol Sync
ID: digital_symbol_sync_xx_0
Timing Error Detector: Mueller and Müller
Samples per Symbol: 89.1284
Expected TED Gain: 1
Loop Bandwidth: 45m
Damping Factor: 1
Maximum Deviation: 1.5
Output Samples/Symbol: 1
Interpolating Resampler: MMSE, 8 tap FIR

Complex To Real
ID: blocks_complex_to_real_0

QT GUI Constellation Sink
ID: qtgui_const_sink_x_0
Number of Points: 1.024k
Autoscale: No

Threshold
ID: blocks_threshold_ff_0
Low: -100m
High: 100m
Initial State: 0

Float To UChar
ID: blocks_float_to_uchar_0

QT GUI Time Sink
ID: qtgui_time_sink_x_0
Number of Points: 1.024k
Sample Rate: 2.1168M
Autoscale: No

Correlate Access Code
ID: digital_c...ess_code_bb_0
Access Code: 00110...10101100
Threshold: 0

F203 = 1111 0010 0000 0011
00110011 01010010 10101010 10101100

Correlate Access Code - Tag
ID: digital_c...code_tag_xx_0
Access Code: 00110...10101100
Threshold: 0
Tag Name: detection

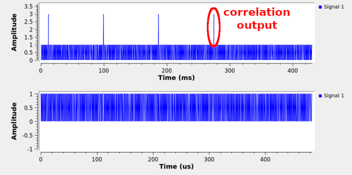
UChar To Float
ID: blocks_uchar_to_float_0

QT GUI Time Sink
ID: qtgui_time_sink_x_1
Number of Points: 1.024k
Sample Rate: 2.375k
Autoscale: No

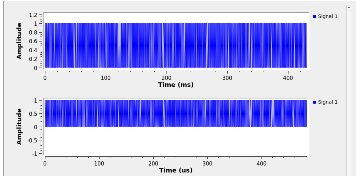
1187.5 samples/s=57000/48
Manchester 1 -> 00 or 11 while 0 -> 10 or 01

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

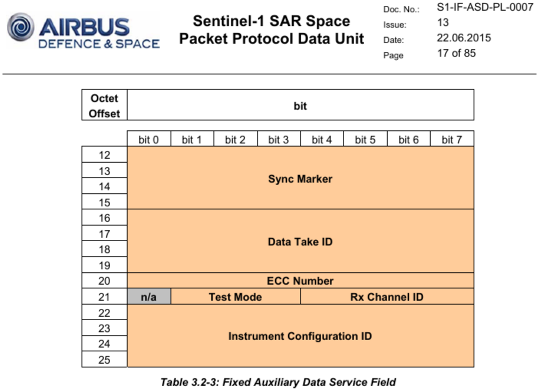
Left right sequence, right wrong sequence



correlation output



Sentinel1 (CCSDS)



AIRBUS DEFENCE & SPACE

Sentinel-1 SAR Space Packet Protocol Data Unit

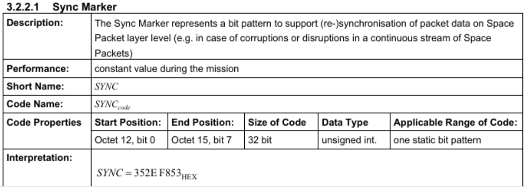
Doc. No.: S1-IF-ASD-PL-0007
Issue: 13
Date: 22.06.2015
Page 17 of 85

Octet Offset	bit 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

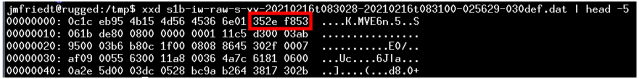
Question: how to find the beginning of sentences in the byte stream of the dowloaded file?



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:



```
jmfriedt@rugged:/tmp$ xxd s1b-iw-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  ................
00000020: 9500 03b6 b80c 1f00 0808 8645 302f 0007  ...........E0/..
00000030: af09 0055 6300 11a8 0036 4a7c 6181 0600  ...Uc....6J|a...
00000040: 0a2e 5d00 03dc 0528 bc9a b264 3817 302b  ..]....(...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.

PACKET PRIMARY HEADER						
PACKET VERSION NUMBER	PACKET IDENTIFICATION			PACKET SEQUENCE CONTROL		PACKET DATA LENGTH
	PACKET TYPE	SEC. HDR. FLAG	APPLICATION PROCESS IDENTIFIER	SEQUENCE FLAGS	PACKET SEQUENCE COUNT OR PACKET NAME	
3 bits	1 bit	1 bit	11 bits	2 bits	14 bits	
2 octets				2 octets		2 octets

Table 3.1-1: Format of Packet Primary Header

AIRBUS
DEFENCE & SPACE

Sentinel-1 SAR Space
Packet Protocol Data Unit

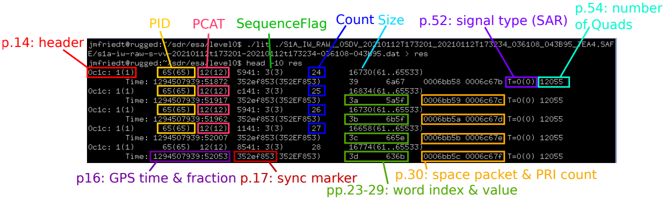
Doc. No.: S1-IF-ASD-PL-0007
Issue: 13
Date: 22.06.2015
Page 14 of 85

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)	• 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"



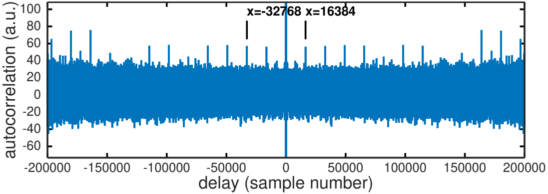
```
jmfriedt@rugged:~/sdr/esa/level0$ ./lit ./S1A_IW_RAW__0SDV_20210112T173201_20210112T173234_036108_043B95_7EA4.SAF
E/s1a-iw-raw-s-vv-20210112t173201-20210112t173234-036108-043b95.dat > res
jmfriedt@rugged:~/sdr/esa/level0$ head -10 res
0c1c: 1(1)       65(65)  12(12)  5941: 3(3)        24       16730(61..65533)
        Time: 1294507939:51872  352ef853(352EF853)          39         6a67     0006bb58 0006c67b T=0(0) 12055
0c1c: 1(1)       65(65)  12(12)  c141: 3(3)        25       16834(61..65533)
        Time: 1294507939:51917  352ef853(352EF853)          3a         5a5f     0006bb59 0006c67c T=0(0) 12055
0c1c: 1(1)       65(65)  12(12)  5941: 3(3)        26       16730(61..65533)
        Time: 1294507939:51962  352ef853(352EF853)          3b         6b5f     0006bb5a 0006c67d T=0(0) 12055
0c1c: 1(1)       65(65)  12(12)  1141: 3(3)        27       16658(61..65533)
        Time: 1294507939:52007  352ef853(352EF853)          3c         665e     0006bb5b 0006c67e T=0(0) 12055
0c1c: 1(1)       65(65)  12(12)  8541: 3(3)        28       16774(61..65533)
        Time: 1294507939:52053  352ef853(352EF853)          3d         636b     0006bb5c 0006c67f T=0(0) 12055
```

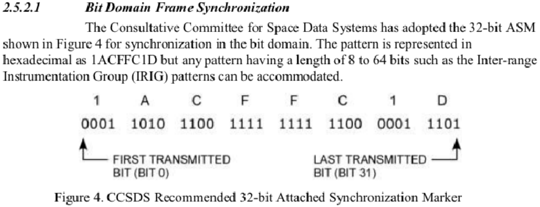
The packet counter is indeed incremented from one packet to the next: consistency of the decoding process[a]

---

[a] https://github.com/jmfriedt/sentinel1_level0

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.

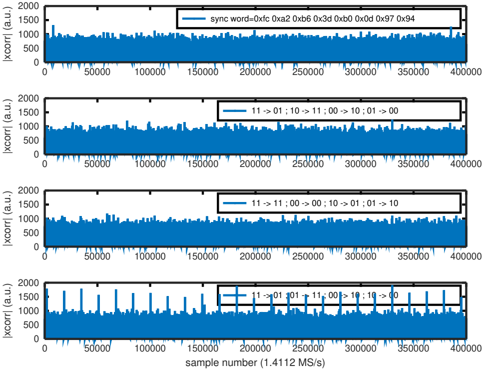
1 A C F F C 1 D

0001 1010 1100 1111 1111 1100 0001 1101

FIRST TRANSMITTED BIT (BIT 0)

LAST TRANSMITTED BIT (BIT 31)

Figure 4. CCSDS Recommended 32-bit Attached Synchronization Marker



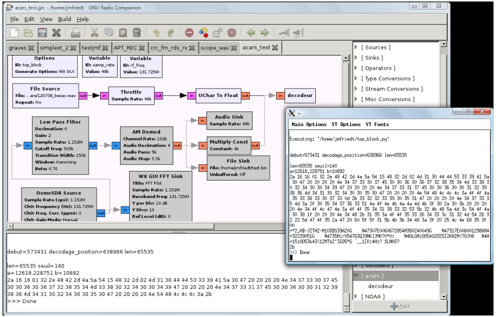
QPSK encodes two bits in each transmitted symbol ($0^\circ = 00$, $90^\circ = 01$, $180^\circ = 11$, $270^\circ = 10$)

Convolutional *Forward Error Correction* – FEC: the synchronization word must be encoded accordingly to be found in the received bit sequence [47]

---

[47] J.-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)

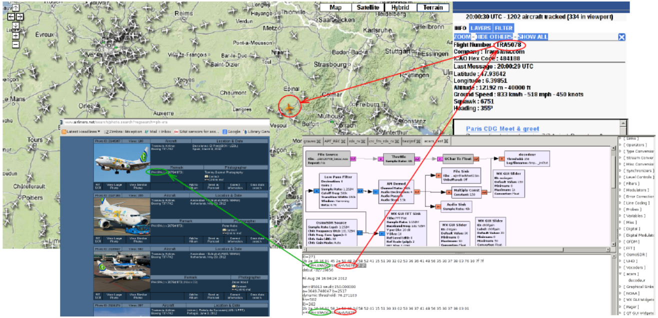
General approach to digital mode demodulation



Options ID: top_block, Generate Options: WX GUI
Variable ID: samp_rate, Value: 48k
Variable ID: rf_freq, Value: 131.725M
File Source File: ...ars/120708_besac.wav, Repeat: No
Throttle Sample Rate: 48k
UChar To Float
decodeur
Low Pass Filter Decimation: 6, Gain: 2, Sample Rate: 1.152M, Cutoff Freq: 500k, Transition Width: 150k, Window: Hamming, Beta: 6.76
AM Demod Channel Rate: 192k, Audio Decimation: 4, Audio Pass: 5k, Audio Stop: 5.5k
Audio Sink Sample Rate: 48k
Multiply Const Constant: 4k
File Sink File: /home/jmfriedt/test.bin, Unbuffered: Off
WX GUI FFT Sink Title: FFT Plot, Sample Rate: 1.152M, Baseband Freq: 131.725M, Y per Div: 10 dB, Y Divs: 10, Ref Level (dB): 0
OsmoSDR Source Sample Rate (sps): 1.152M, Ch0: Frequency (Hz): 131.725M, Ch0: Freq. Corr. (ppm): 0, Ch0: Gain Mode: Manual

```
debut=573431 decodage_position=638966 len=65535

len=65535 seuil=140
a=12618.228751 b=10892
2a 16 16 01 32 2e 48 42 2d 4a 5a 54 15 48 32 2d 02 4d 31 30 44 44 53 33 39 41 5a 30 47 20 20 20 20 4e 34 37 33 30 37 45
30 30 36 30 36 37 32 38 35 34 4d 33 38 30 32 34 30 30 34 39 47 20 20 20 20 4e 34 37 33 31 37 45 30 30 36 30 30 31 32 39
38 36 4d 34 31 30 32 34 30 30 35 30 47 20 20 20 20 4e 54 48 4c 4c 4c 3a 2b
>>> Done
```

```
Executing: "/home/jmfriedt/top_block.py"


debut=573431 decodage_position=638966 len=65535

len=65535 seuil=140
a=12618.228751 b=10892
2a 16 16 01 32 2e 48 42 2d 4a 5a 54 15 48 32 2d 02 4d 31 30 44 44 53 33 39 41 5a
 30 47 20 20 20 20 4e 34 37 33 30 37 45 30 30 36 30 36 37 32 38 35 34 4d 33 38 3
0 32 34 30 30 34 39 47 20 20 20 20 4e 34 37 33 31 37 45 30 30 36 30 30 31 32 39
38 36 4d 34 31 30 32 34 30 30 35 30 47 20 20 20 20 4e 54 48 4c 4c 4c 3a 4f 4f 4a
 35 33 38 33 30 37 33 0d 2b 33 32 32 33 39 30 35 31 47 20 20 20 20 4e 34 37 33 3
5 4d 3a 2f 30 35 34 37 36 33 51 4a 4f 4e 4b 4a 4a 4d 4b 3f 30 2a 30 3b 20 20 20
20 4e 34 4f 4c 47 4e 3a 4f 4f 35 34 31 38 33 32 53 5b 32 4b 48 5a 4d 3c 54 4f 4a
 30 38 1f 20 20 20 4e 34 48 2b 31 35 3a 4f 4f 35 33 26 34 33 3c 31 32 4d 54 26 3
2 22 5d 47 4f 35 2a 47 20 60 5f 5f 31 5b 4b 3b 34 48 3a 3f 20 25 4c 4e 68 35 3f
00
+*2.HB-JZTH2-M10DDS39AZ0G    N47307E0060672854M380240049G    N47317E0060012986M4
+32239051G    N4735M:/054763QJONKJJMK?0*0;    N40LGN:00541832S[2KHZM<TOJ08    N4H
+15:0053&43<12MT&2"]G05*G `__1[K:4H:? %LNh5?
2b
>>> Done
```



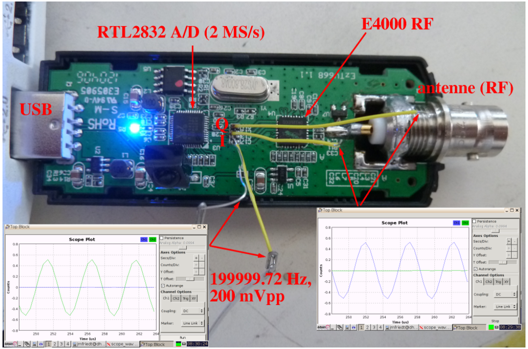
20:00:30 UTC - 1202 aircraft tracked (334 in viewport)
Flight Number : TRA5078
Company : Transavia.com
ICAO Hex Code : 484188
Last Message : 20:00:29 UTC
Latitude : 47.93642
Longitude : 6.39851
Altitude : 12192 m - 40000 ft
Ground Speed : 833 km/h - 518 mph - 450 knots
Squawk : 6751
Heading : 355°

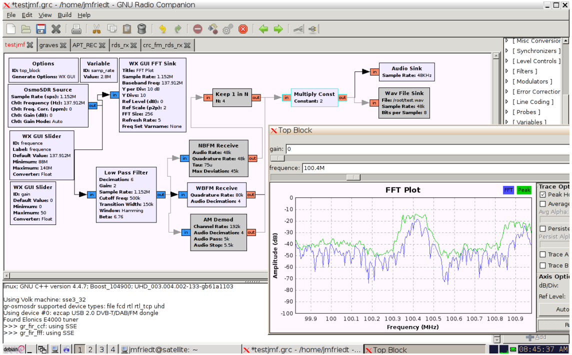
Beyond using available processing blocks: use GNURadio's basic blocks (demodulation) and add custom functionalities (**assuming** external tools are not already available):

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

Software defined radio (SDR)

- `gnuradio` [48] and its associated graphical environment `gnuradio-companion` provide the basic tools for signal prcessing prototyping (acquisition, filters, demodulation, audio and file outputs).
- RTL2832U ADC-USB+R820T(2) frontend based DVB-T receiver [49] ($\leq$10 \$) as a general purpose radiofrequency receiver with(out) IF, providing a dual 2.5 MS/s datastream in the 30–1600 MHz range.





Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 2.8M

OsmoSDR Source
Sample Rate (sps): 1.152M
Ch0: Frequency (Hz): 137.912M
Ch0: Freq. Corr. (ppm): 0
Ch0: Gain (dB): 0
Ch0: Gain Mode: Auto

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 1.152M
Baseband Freq: 137.912M
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 256
Refresh Rate: 5
Freq Set Varname: None

WX GUI Slider
ID: frequence
Label: frequence
Default Value: 137.912M
Minimum: 88M
Maximum: 140M
Converter: Float

WX GUI Slider
ID: gain
Default Value: 0
Minimum: 0
Maximum: 50
Converter: Float

Low Pass Filter
Decimation: 6
Gain: 2
Sample Rate: 1.152M
Cutoff Freq: 500k
Transition Width: 150k
Window: Hamming
Beta: 6.76

NBFM Receive
Audio Rate: 48k
Quadrature Rate: 48k
Tau: 75u
Max Deviation: 45k

WBFM Receive
Quadrature Rate: 80k
Audio Decimation: 4

AM Demod
Channel Rate: 192k
Audio Decimation: 4
Audio Pass: 5k
Audio Stop: 5.5k

Keep 1 in N
N: 4

Multiply Const
Constant: 2

Audio Sink
Sample Rate: 48KHz

Wav File Sink
File: /root/test.wav
Sample Rate: 48k
Bits per Sample: 8

gain: 0
frequence: 100.4M

```
linux; GNU C++ version 4.4.7; Boost_104900; UHD_003.004.002-133-gb61a1103

Using Volk machine: sse3_32
gr-osmosdr supported device types: file fcd rtl rtl_tcp uhd
Using device #0: ezcap USB 2.0 DVB-T/DAB/FM dongle
Found Elonics E4000 tuner
>>> gr_fir_ccf: using SSE
>>> gr_fir_fff: using SSE
```

---

[48] `gnuradio.org`

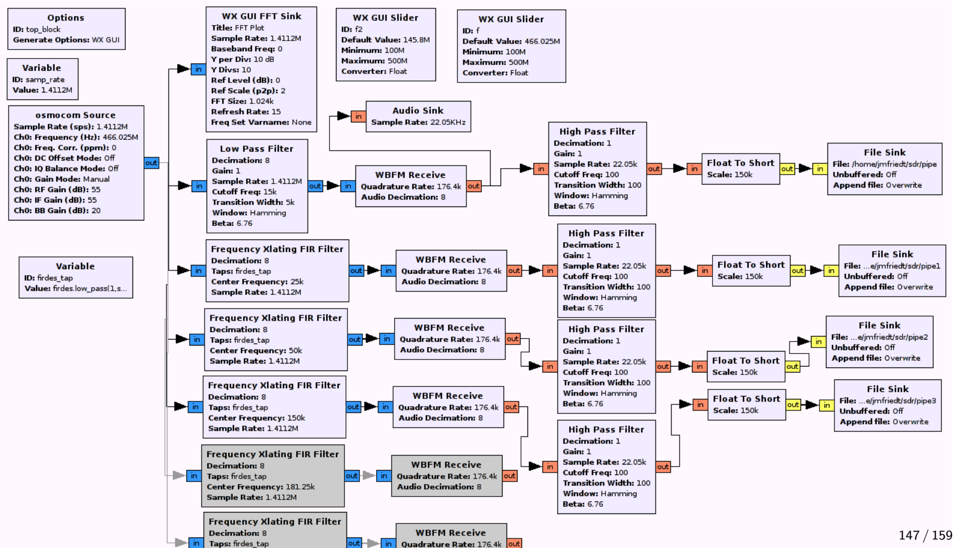
[49] J.-M Friedt, G. Goavec-Mérou, *La réception radiofréquence définie par logiciel (Software Defined Radio – SDR)* GNU/Linux Magazine France 153 (Oct. 2012), pp.4-33, `jmfriedt.free.fr/lm_sdr.pdf`

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 → ×
- filter → FIR
- rectifier → abs()

SDR:

- stable,
- flexible,
- reconfigurable



Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 1.4112M

osmocom Source
Sample Rate (sps): 1.4112M
Ch0: Frequency (Hz): 466.025M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 55
Ch0: IF Gain (dB): 55
Ch0: BB Gain (dB): 20

Variable
ID: firdes_tap
Value: firdes.low_pass(1,s...

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 1.4112M
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

WX GUI Slider
ID: f2
Default Value: 145.8M
Minimum: 100M
Maximum: 500M
Converter: Float

WX GUI Slider
ID: f
Default Value: 466.025M
Minimum: 100M
Maximum: 500M
Converter: Float

Audio Sink
Sample Rate: 22.05KHz

Low Pass Filter
Decimation: 8
Gain: 1
Sample Rate: 1.4112M
Cutoff Freq: 15k
Transition Width: 5k
Window: Hamming
Beta: 6.76

WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

High Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 22.05k
Cutoff Freq: 100
Transition Width: 100
Window: Hamming
Beta: 6.76

Float To Short
Scale: 150k

File Sink
File: /home/jmfriedt/sdr/pipe
Unbuffered: Off
Append file: Overwrite

Frequency Xlating FIR Filter
Decimation: 8
Taps: firdes_tap
Center Frequency: 25k
Sample Rate: 1.4112M

WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

High Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 22.05k
Cutoff Freq: 100
Transition Width: 100
Window: Hamming
Beta: 6.76

Float To Short
Scale: 150k

File Sink
File: ...e/jmfriedt/sdr/pipe1
Unbuffered: Off
Append file: Overwrite

Frequency Xlating FIR Filter
Decimation: 8
Taps: firdes_tap
Center Frequency: 50k
Sample Rate: 1.4112M

WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

High Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 22.05k
Cutoff Freq: 100
Transition Width: 100
Window: Hamming
Beta: 6.76

Float To Short
Scale: 150k

File Sink
File: ...e/jmfriedt/sdr/pipe2
Unbuffered: Off
Append file: Overwrite

Frequency Xlating FIR Filter
Decimation: 8
Taps: firdes_tap
Center Frequency: 150k
Sample Rate: 1.4112M

WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

High Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 22.05k
Cutoff Freq: 100
Transition Width: 100
Window: Hamming
Beta: 6.76

Float To Short
Scale: 150k

File Sink
File: ...e/jmfriedt/sdr/pipe3
Unbuffered: Off
Append file: Overwrite

Frequency Xlating FIR Filter
Decimation: 8
Taps: firdes_tap
Center Frequency: 181.25k
Sample Rate: 1.4112M

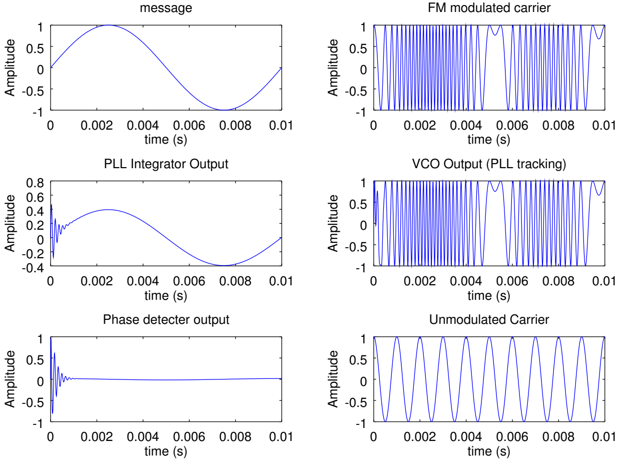
WBFM Receive
Quadrature Rate: 176.4k
Audio Decimation: 8

Frequency Xlating FIR Filter
Decimation: 8
Taps: firdes_tap

WBFM Receive
Quadrature Rate: 176.4k

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

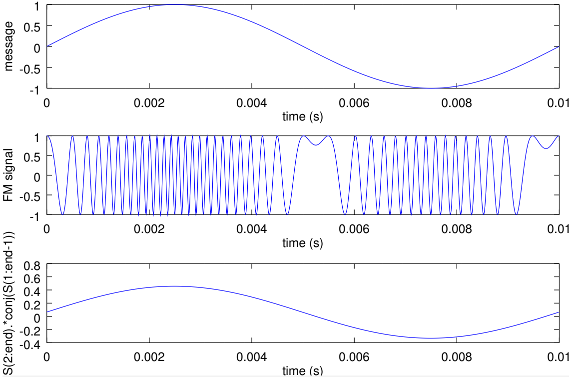
$$\int_{n\cdot T_s}^{(n+1)\cdot T_s} m(\tau)d\tau \simeq T_s \cdot m(n \cdot T_s) \text{ (rectangle)}$$

so that

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

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

This is the *quadrature FM demodulator* implementation of GNURadio [a]



```
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**

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

---

[a] D. Bederov, *Arithmetic based implementation of a quadrature FM Demodulator*, at `https://fosdem.org/2015/schedule/event/sdr_arithmetic/`

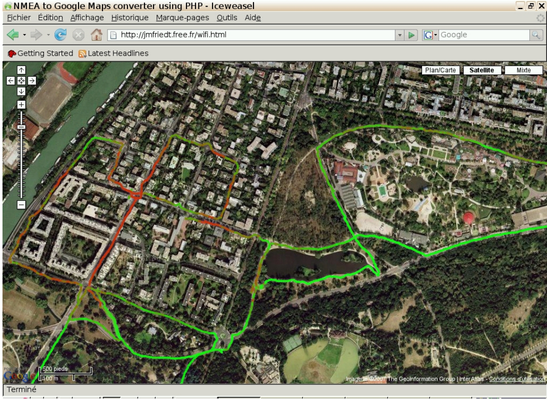
Limitations of the software approach

- limitation on the sampling rate (measurement bandwidth)
- sound card $\leq$ 192 kS/s (16 bits), DVB-T $\leq$ 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

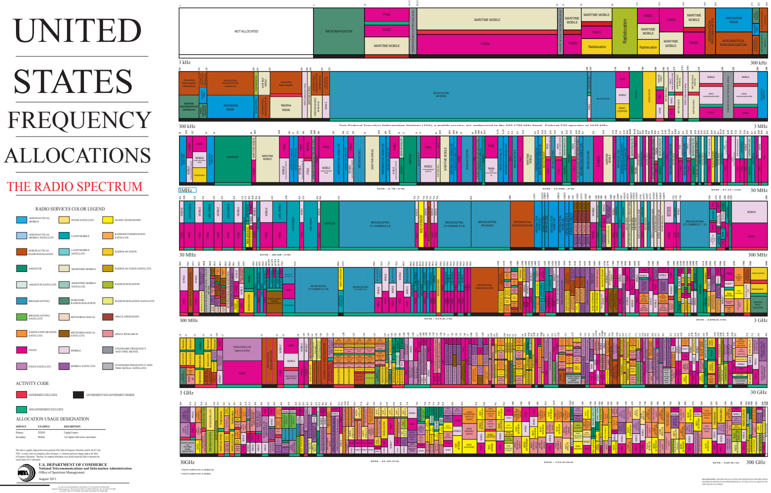
- high power consumption ⇒ consider power saving strategies ⇒ data storage and compression
- lower reliability than wired links
- various technologies depending on the objectives: radiomodems, wifi, bluetooth/zigbee, LoRa
- selecting the right technology depends on the purpose. Consider redundancy (local 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

Include

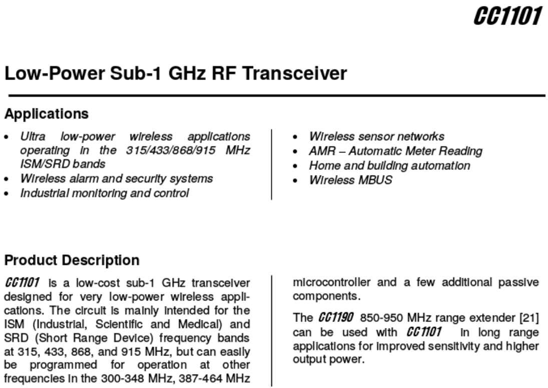
- the allocated frequency band (ISM: 13.56, 433.9 and 2440 MHz),
- **radiated** power,
- duty cycle, type of signal[50] …



---

[50] ETSI EN 300 220, available at `https://www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_40/en_30022001v020401o.pdf`

Emission regulations



CC1101

Low-Power Sub-1 GHz RF Transceiver

Applications

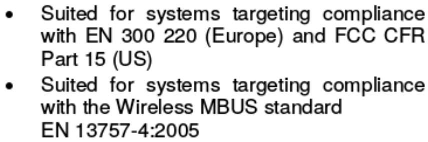
- Ultra low-power wireless applications operating in the 315/433/868/915 MHz ISM/SRD bands
- Wireless alarm and security systems
- Industrial monitoring and control
- Wireless sensor networks
- AMR – Automatic Meter Reading
- Home and building automation
- Wireless MBUS

Product Description

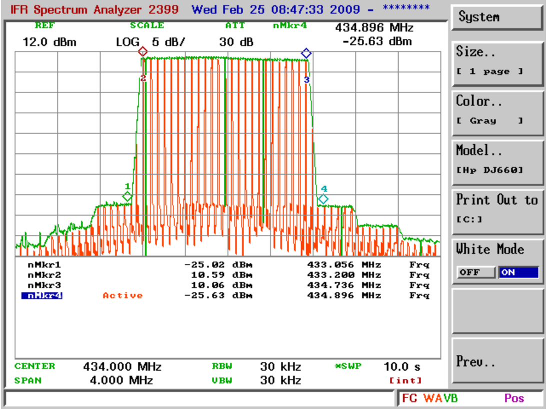
CC1101 is a low-cost sub-1 GHz transceiver designed for very low-power wireless applications. The circuit is mainly intended for the ISM (Industrial, Scientific and Medical) and SRD (Short Range Device) frequency bands at 315, 433, 868, and 915 MHz, but can easily be programmed for operation at other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



```
IFR Spectrum Analyzer 2399   Wed Feb 25 08:47:33 2009 - ********
REF          SCALE          ATT      nMkr4
12.0 dBm     LOG  5 dB/     30 dB           434.896 MHz
                                            -25.63 dBm

nMkr1                 -25.02 dBm     433.056 MHz     Frq
nMkr2                  10.59 dBm     433.200 MHz     Frq
nMkr3                  10.06 dBm     434.736 MHz     Frq
nMkr4      Active     -25.63 dBm     434.896 MHz     Frq

CENTER    434.000 MHz     RBW    30 kHz     *SWP    10.0 s
SPAN        4.000 MHz     VBW    30 kHz             [int]
                                            FC WAVB       Pos
```

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

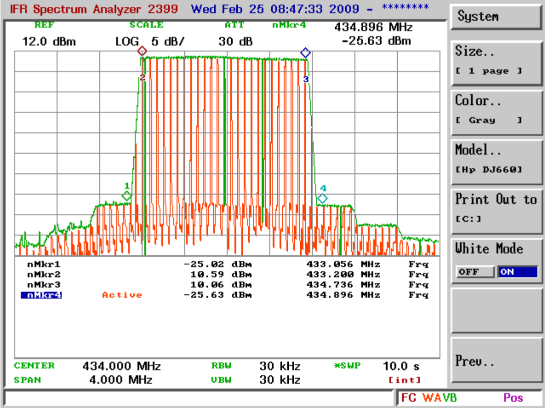
Emission regulations

Include

- the allocated frequency band (ISM: 13.56, 433.9 and 2440 MHz),
- **radiated** power,
- duty cycle, type of signal[50] ...

+

- sensitivity to supply voltage/noise (mains supply voltage noise)
- avoid distrurbing 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



```
IFR Spectrum Analyzer 2399  Wed Feb 25 08:47:33 2009 - ********
  REF          SCALE          ATT     nMkr4    434.896 MHz
12.0 dBm     LOG  5 dB/      30 dB             -25.63 dBm

 nMkr1                      -25.02 dBm     433.056 MHz     Frq
 nMkr2                       10.59 dBm     433.200 MHz     Frq
 nMkr3                       10.06 dBm     434.736 MHz     Frq
 nMkr4        Active        -25.63 dBm     434.896 MHz     Frq

CENTER     434.000 MHz      RBW    30 kHz     *SWP    10.0 s
SPAN         4.000 MHz      VBW    30 kHz            [int]
```

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

---

[50] ETSI EN 300 220, available at `https://www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_40/en_30022001v020401o.pdf`

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. Schafer, *Discrete-Time Signal Processing (3rd Edition)*, Prentice-Hall Signal Processing Series (2009), and videos of his lectures at `ocw.mit.edu/resources/res-6-007-signals-and-systems-spring-2011/video-lectures/lecture-1-introduction/`
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 02/2021[51])
5. *Learning DSP Illustrated* at `https://dspillustrations.com/pages/index.html` (accessed 02/2021)

---

[51] presented 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)