Software Defined Radio for time & frequency metrology: demonstration with GNU Radio

J.-M Friedt

FEMTO-ST Time & Frequency department, Besançon, France

jmfriedt@femto-st.fr

slides and references at
`jmfriedt.free.fr`

April 17, 2021

SDR: flexible and stable approach to RF signal processing

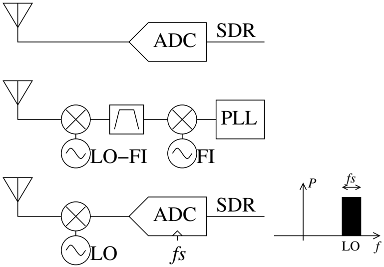
Software Defined Radio (SDR): digital radiofrequency (RF) digital [1] signal processing [2]

- stable: an algorithm will not drift over time ($\neq$ passive component, e.g. capacitor) or with environmental conditions
- flexible: ability to tune operating conditions without halting operation
- reconfigurable: one hardware, many application only requiring reconfiguration of connections

+ data logging + communication over networks ...

Application to **oscillator metrology**, (secure) **time transfer** and **timing** (of active/passive RADAR signals)

In the current architecture, carrier (LO) will not matter (removed by RF frontend), only bandwidth (sampling rate f_s) will matter $\longrightarrow$



[1] D.A. Mindell, *Digital Apollo: Human and Machine in Spaceflight*, MIT Press (2011)

[2] D.A. Mindell, *Between Human and Machine: Feedback, Control, and Computing before Cybernetics*, Johns Hopkins University Press (2003)

Red Pitaya/STEMLab (baseband)

- T. Preuschoff *& al.*, *Digital laser frequency and intensity stabilization based on the STEMlab platform (originally Red Pitaya)*, Rev. Sci. Instrum. **91** (8) 083001 (2020)
- A. Tourigny-Plante *& al.*, *An open and flexible digital phase-locked loop for optical metrology*, Rev. Sci. Instrum. **89** (9) 093103 (2018)
- P. Mahnke, *Characterization of a commercial software defined radio as high frequency lock-in amplifier for FM spectroscopy*, Rev. Sci. Instrum. **89** (1) 013113 (2018)
- J.A. Sherman *& al.*, *Oscillator metrology with software defined radio*, Rev. Sci. Instrum. **87** (5) 054711 (2016)
- C. Hasselwander *& al.*, *gr-MRI: A software package for magnetic resonance imaging using software defined radios*, Journal of Magnetic Resonance **270** 47–55 (2016)
- F. Balakirev *& al.*, *Resonant ultrasound spectroscopy: The essential toolbox*, Rev. Sci. Instrum. **90** (12) 121401 (2019)
- G.A. Stimpson *& al.*, *An open-source high-frequency lock-in amplifier*, Rev. Sci. Instrum., **90** (9) 094701 (2019)
- A. C. Cárdenas *& al.*, *Phase Noise and Frequency Stability of the Red-Pitaya Internal PLL*, IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control **66** (2) 412–416 (2019)
- S.J. Yu *& al.*, *The performance and limitations of FPGA-based digital servos for atomic, molecular, and optical physics experiments*, Rev. Sci. Instrum. **89** 025107 (2018)

Ettus Research B210:

- Paul Meaney *& al.*, *A 4-channel, vector network analyzer microwave imaging prototype based on software defined radio technology*, Rev. Sci. Instrum. **90** 044708 (2019)

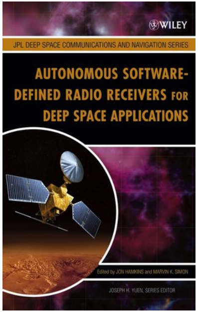
Ettus Research E312:

- S. Prager *& al.*, *Ultrawideband Synthesis for High-Range-Resolution Software-Defined Radar* IEEE Trans. Instrum. & Meas. **69**(6) 3789–3803 (2019)
- S. Prager *& al.*, *Wireless subnanosecond RF synchronization for distributed ultrawideband software-defined radar networks*, IEEE Trans. Microwave Theory and Techniques **68**(11) 4787–4804 (2020)

Ettus Research N210:

- C. Andrich *& al.*, *High-precision measurement of sine and pulse reference signals using software-defined radio* IEEE Trans. Instrum. & Meas. **67** (5) 1132–1141 (2018)

SDR in space

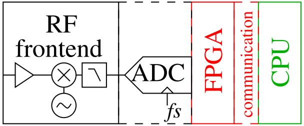


"In addition to easing the scheduling and configuration burden, an autonomous radio also will **gracefully handle unpredictable or anomalous events**. For example, during entry, descent, and landing (EDL), a spacecraft can undergo large Doppler swings caused by rocket firings, parachute openings, backshell ejection, and a bouncing landing on the surface. Even when all scheduled events occur successfully, there may be Doppler uncertainty due to unpredictable properties of the atmosphere. Ideally, the communication link should operate whether or not each of the EDL events is successful, but the uncertainties involved typically lead to liberal link margins—for example, the **Mars Exploration Rovers** observed link margins that sometimes exceeded 10 dB. An autonomous radio could substantially reduce this margin because it would handle any Doppler swing nearly optimally. Unfortunately, such flexible technology is not available on NASA's currently flying missions. In perhaps the most glaring example of this, NASA engineers discovered in 2000 that a receiver aboard Cassini, launched in 1997, would fail during the Huygens probe descent onto Titan because it did **not properly account for the Doppler profile of the probe**. Increasing the loop bandwidth of the synchronization loops would have easily fixed the problem, but, unfortunately, these loop bandwidths were hard-wired to fixed values on the spacecraft. With superior engineering and enormous dedication, NASA and the European Space Agency were still able to save the mission by slightly altering the original trajectory, but this solution required forming a large and expensive international recovery team to find the appropriate recommendations on how to overcome the radio's severe limitations."

J. Hamkins & al. *Autonomous Software-Defined Radio Receivers for Deep Space Applications*, Deep Space Communications and Navigation Series (NASA/JPL, 2006), p.2:
`descanso.jpl.nasa.gov/monograph/series9/Descanso9_Full_rev2.pdf`

Free Opensource development frameworks

Need to address both the FPGA (fast, massively parallel) and the CPU (flexible, high level language, networking, user interface ...)



- Pyrpl (`https://pyrpl.readthedocs.io/`)
- Chisel & SpinalHDL (Scala language) at `https://www.chisel-lang.org/` and `https://github.com/SpinalHDL/SpinalHDL`
- nMigen & LiteX (`https://github.com/enjoy-digital/litex`)
- Oscimp Digital (FEMTO-ST: `https://github.com/oscimp/oscimpDigital/`)
- Edalize (`https://github.com/olofk/edalize`)
- **GNU Radio** and RFNoC (for Ettus Research hardware), `gr-verilog`

GNU Radio:

- free opensource signal processing framework
- digital signal processing blocks in C++ (or Python)...
- ... connected through a Python description of datastream.
- Real time processing ($\gg$ GNU/Octave or Python post-processing)
- Graphical User Interface for generating Python scripts: GNU Radio Companion

Fundamental of time transfer

How to demonstrate time transfer with SDR?

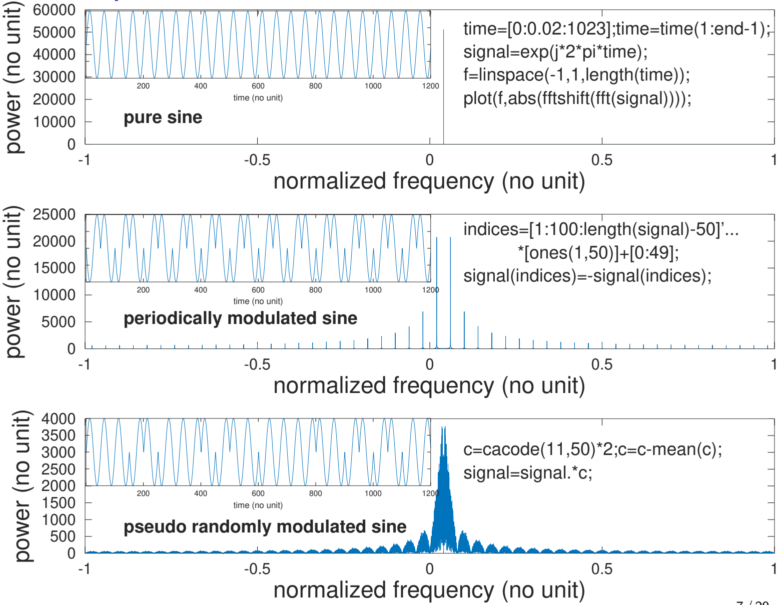
- RADAR range resolution: $\Delta R \geq \frac{c_0}{2B}$ ($c_0 = 300$ m/μs, bandwidth B)
- spectrum spreading: maximize B by all means (pulse, frequency sweep, frequency steps, noise ...)
- receive time delayed copies of the transmitted signal: matched filter = correlation (search for delayed copies of the emitted signal)

$$xcorr(x,y)(\tau) = \int_{-T/2}^{T/2} x(t)y(t+\tau)dt \Rightarrow \text{ identify } \tau \text{ maximizing } xcorr$$

- maximize averaging time T to smooth out noise
- maximize B for the correlation peak width $1/B$ to be as narrow as possible
- Pulse Compression Ratio: $B \times T$

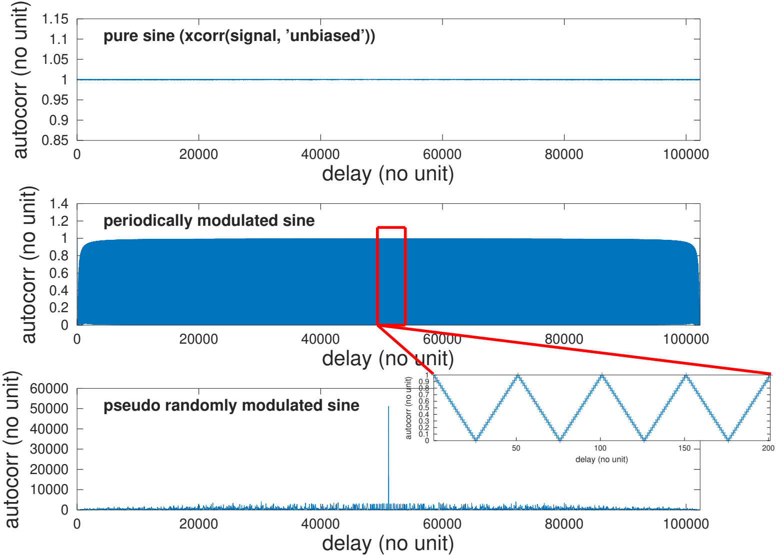
Spectrum spreading numerical experiments

- Carrier frequency and bandwidth are two unrelated quantities which can be tuned independently
- Carrier frequency defined by first frequency transposition stage (RF frontend) whereas bandwidth defined by ADC sampling rate
- Binary Phase shift keying: $\varphi \in [0; \pi]$ for spectrum spreading



Spectrum spreading numerical experiments

- Carrier frequency and bandwidth are two unrelated quantities which can be tuned independently
- Carrier frequency defined by first frequency transposition stage (RF frontend) whereas bandwidth defined by ADC sampling rate
- Binary Phase shift keying: $\varphi \in [0; \pi]$ for spectrum spreading



Spectrum spreading numerical experiments

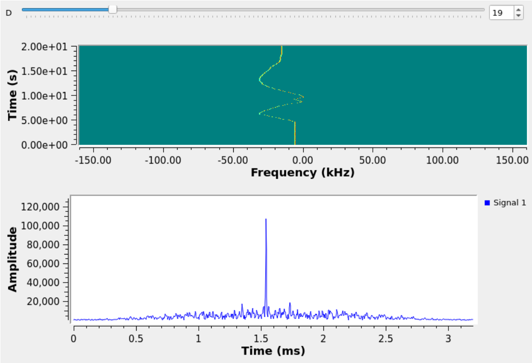
From convolution to correlation:

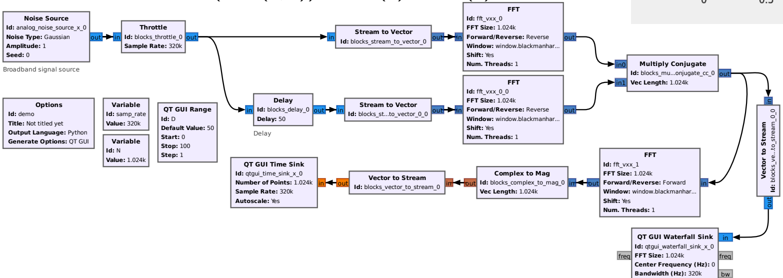
- Convolution: $conv(s,r)(\tau) = \int s(t)r(\tau - t)dt$
- Practical computation of convolution:

$$FT(conv(s,r)) = FT(s) \cdot FT(r)$$

- Correlation: $corr(s,r)(\tau) = \int s(t)r(t+\tau)dt$
- Convolution $\rightarrow$ correlation: time reversal
- since $\exp(j\omega t)^* = \exp(-j\omega t)$, we conclude

$$FT(corr(s,r)) = FT(s) \cdot FT^*(r)$$



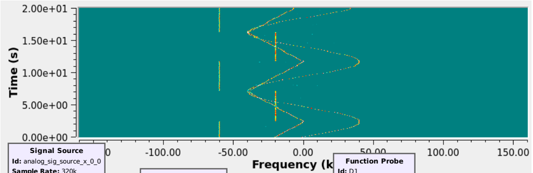


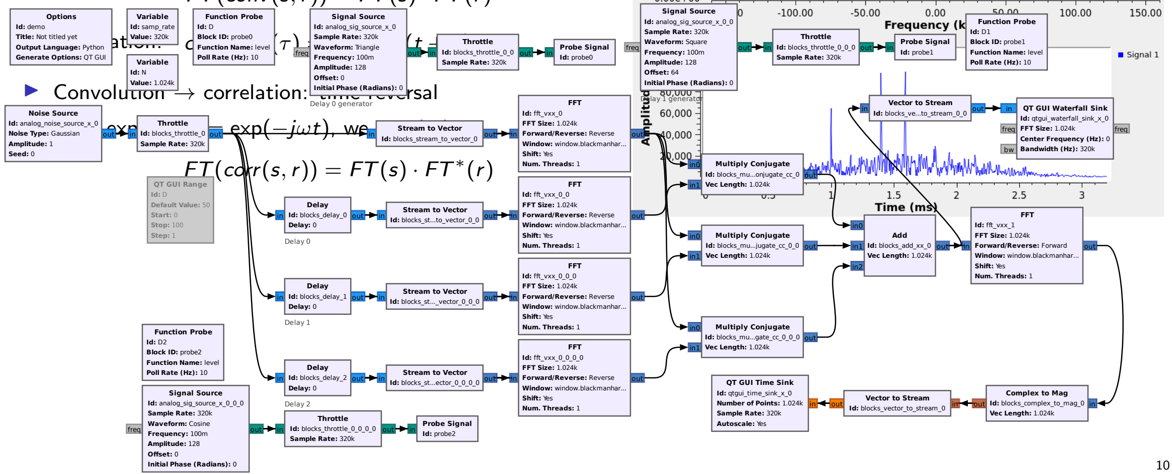
Spectrum spreading numerical experiments

From convolution to correlation:

- Convolution: $conv(s,r)(\tau) = \int s(t)r(\tau - t)dt$
- Practical computation of convolution:

$$FT(conv(s,r)) = FT(s) \cdot FT(r)$$





ation: c (τ) $(t -$

- Convolution $\rightarrow$ correlation: time reversal

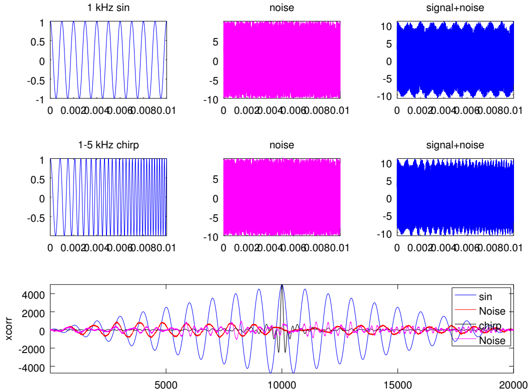
exp $\exp(-j\omega t)$, we

$$FT(corr(s,r)) = FT(s) \cdot FT^*(r)$$

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 $1/B$

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

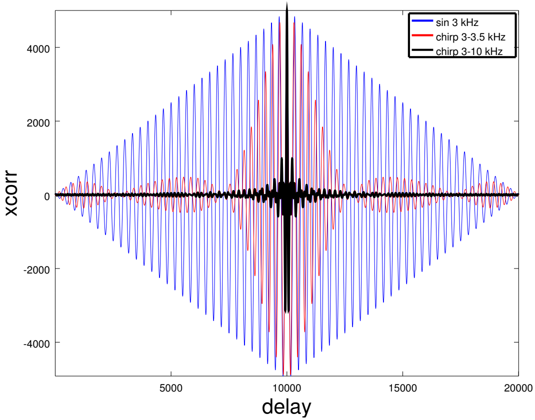



```
time=[0:1e-6:1e-2]; %samp. rate=1 us

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 $1/B$

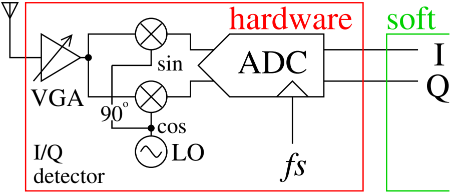
pulse compression ratio (PCR) $= B \cdot T$




```
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-');
```

Why SDR handles complex numbers, ...

- real signal Fourier transform is conjugate symmetric (negative frequency and positive frequency magnitude equal)
- the spectrum transposed from RF band $s = A\exp(j\omega t + \varphi)$ to baseband need not be symmetric $\Rightarrow$ complex mixing to create I and Q (Identity and Quadrature)
- $I = s \cdot \cos(\omega_{RF} t)$ and $Q = s \cdot \sin(\omega_{RF} t)$ so that $A = |I + jQ|$ and $\varphi = arg(I + jQ)$ if $\omega_{RF} = \omega$
- In other words ... imagine a single frequency transposition $s(t) \cdot \cos(\omega_{RF} t)$: if the modulation is on the amplitude, then $A\cos\varphi = 0$ if $\varphi = \pi/2$, $\forall A$.
- Solution: add a second signal maximized when $\cos\varphi = 0$, i.e. using sin
- since $\sin(x)$ is $\cos(x + \pi/2)$: quadrature of the local oscillator



… and double frequency transposition: digital IQ vs analog IQ [3]

also used in the NanoVNA (`https://github.com/ttrftech/NanoVNA`)

MPC-S1 **Sentinel-1**

Ref	
MPC Nom:	DI-MPC-IPFDPM
MPC Ref:	MPC-0307
Issue/Revision:	2/2
Date:	07/06/2019

Sentinel-1 Level 1 Detailed Algorithm Definition

4.1 Raw Data Analysis

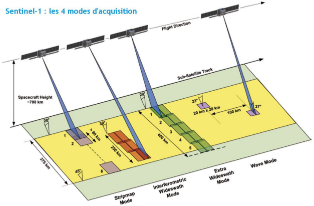
Raw data analysis is required in order to perform corrections of the I and Q channels of the raw signal data. The classical raw data correction (applied for instance in the case of ENVISAT-ASAR and RADARSAT-2) involves (see also Section 9.2):

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

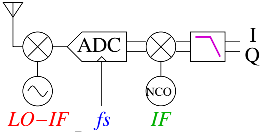
For Sentinel-1 however, the instrument's receive module performs the demodulation in the digital domain, therefore the I/Q gain imbalance and I/Q non-orthogonality corrections are no longer necessary.

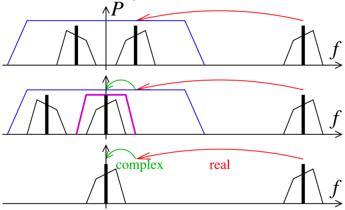
The raw data analysis necessary for the raw data correction of ASAR data is defined in [R-6]. Since the IPF also supports the processing of ASAR data, for completeness, the ASAR raw data analysis scheme is reproduced in this section.

Even though for Sentinel-1 the I/Q gain imbalance and the I/Q non-orthogonality corrections are not necessary, they will be made available optionally, using configuration input parameters. Irrespective to the correction flag though, the Raw Data Analysis described in this section will be performed and the results reported for both ASAR and Sentinel-1 data.



- real case: $I = s(t)\cos(\omega_{RF} t)$ $Q = s(t)\cdot(1+\varepsilon)\sin(\omega_{RF} t + \delta\varphi)$: analog IQ imbalance
- avoid analog IQ imbalance with dual transposition step
- digital domain frequency transposition: `Xlating FIR Filter`

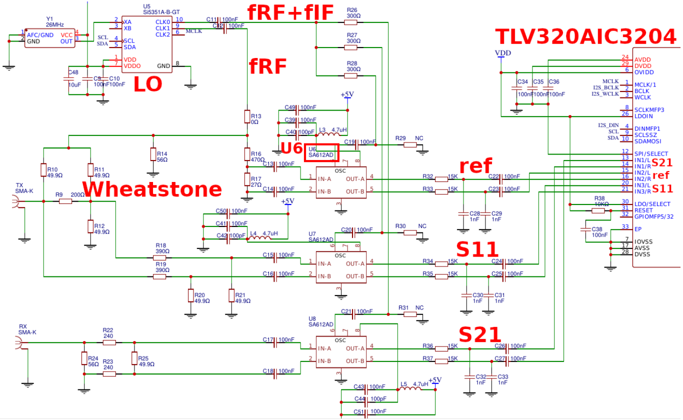




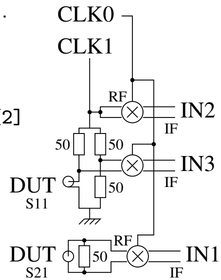
[3] `http://www.esa.int/var/esa/storage/images/esa_multimedia/images/2016/03/sentinel-1_radar_mission/15857809-1-eng-GB/Sentinel-1_radar_mission_pillars.jpg`

NanoVNA frequency transposition architecture

Openhardware[4]/opensource[5] vector network analyzer



- main.c: `static int32_t frequency_offset = 5000;` the intermediate frequency sampled by the audio codec, with (si5351.c) CLK0: frequency + offset and CLK1: frequency when requesting `si5351_set_frequency_with_offset(uint32_t freq, int offset, uint8_t drive_strength)`.
- Audio signal sampled at 48 kHz ⇒ `sincos_tbl[48][2]` in `dsp.c` for transposition by 5 kHz in `dsp_process()` of `dsp.c`).

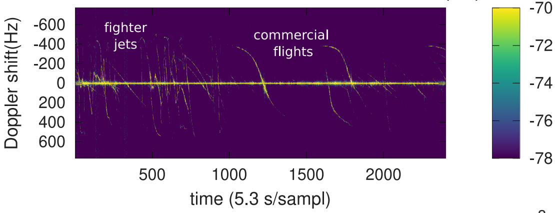


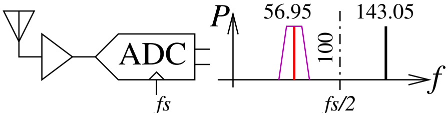
[4] https://raw.githubusercontent.com/hugen79/NanoVNA-H/master/doc/Schematic_nanovna-H_REV3_4_2.pdf

[5] https://github.com/ttrftech/NanoVNA

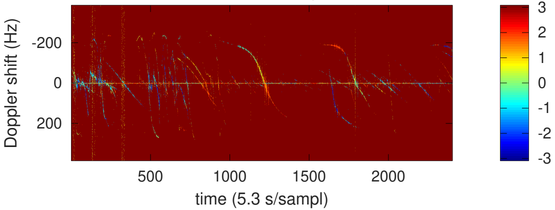
Benefits of digital: example of distributed passive RADAR (4 antennas)

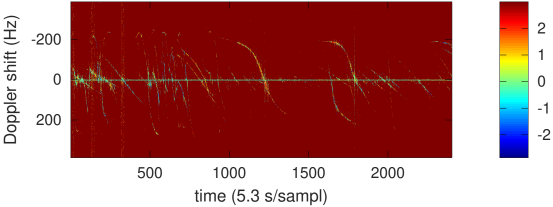
- aliasing/higher order Nyquist modes (case of CW RADAR receiver) – challenge of synchronizing RF frontends (random and time varying phase offset at PLL output)
- Doppler shift in RADAR is $\delta f_{Doppler(Hz)} = 2 f_{carrier} \frac{v}{c} \simeq v_{m/s}$ @ $f_{carrier} = 143.05$ MHz





ADC clock controlled by 10 MHz + 1 PPS from White Rabbit for frequency & time synchronization



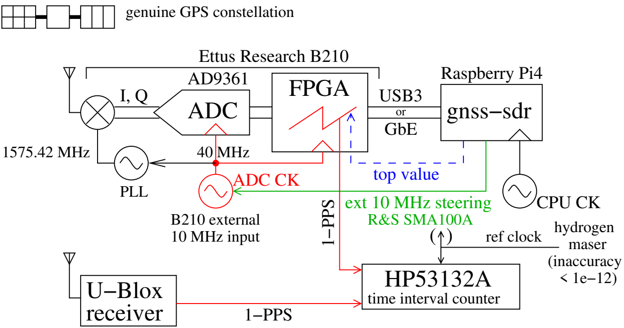


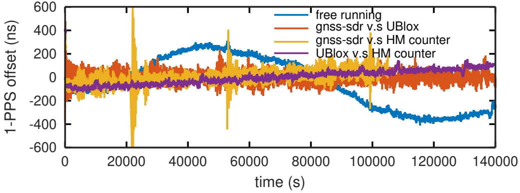
White Rabbit synchronized Ettus Research X310 receivers ($\sigma_\tau \simeq 60$ ps), 200 MS/s recording of GRAVES [6] 143.05 MHz <u>CW reflected by planes – phase analysis</u> $\Rightarrow$ avoid LO mixing by sampling $200 - 143.05 = 56.95$ MHz: DoA analysis.

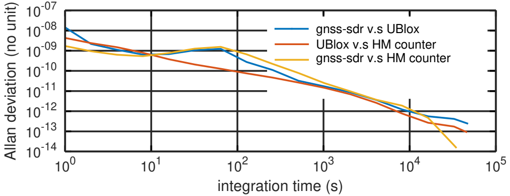
[6] W. Feng, J.-M Friedt, G. Cherniak, M. Sato, *(Yet another) passive RADAR using DVB-T receiver and SDR*, FOSDEM 2018

1-PPS generation from SDR based GNSS receiver (`gnss-sdr`)

- distributed timing: only the ADC can timestamp samples, all further processing is asynchronous







`gnss-sdr` [7] provides a PVT solution with the time delay between a local copy of the PRN sequence and the received signals $\Rightarrow$ control the clock feeding FPGA+ADC accordingly

[7] `https://gnss-sdr.org/` and our fork at `https://github.com/oscimp/gnss-sdr-1PPS`

Drawbacks of digital & SDR

- each operations requires many clock cycles: *low* control bandwidth despite high clock frequencies
- example of the Finite Impulse Response filter:

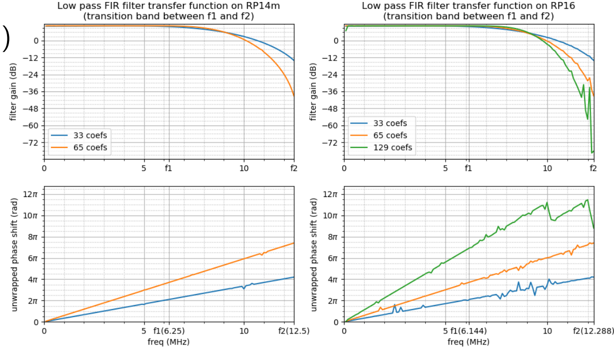
$$y_n = \sum_k^N b_k x_{n-k}$$

will introduce a delay of N clock cycles T_s and a phase shift $\varphi = 2\pi \frac{N}{2} \cdot T_s \cdot f$ at frequency f

$\Rightarrow$ limited closed loop control bandwidth

- low dynamic range (8 bit=48 dB ; 16 bit=96 dB)

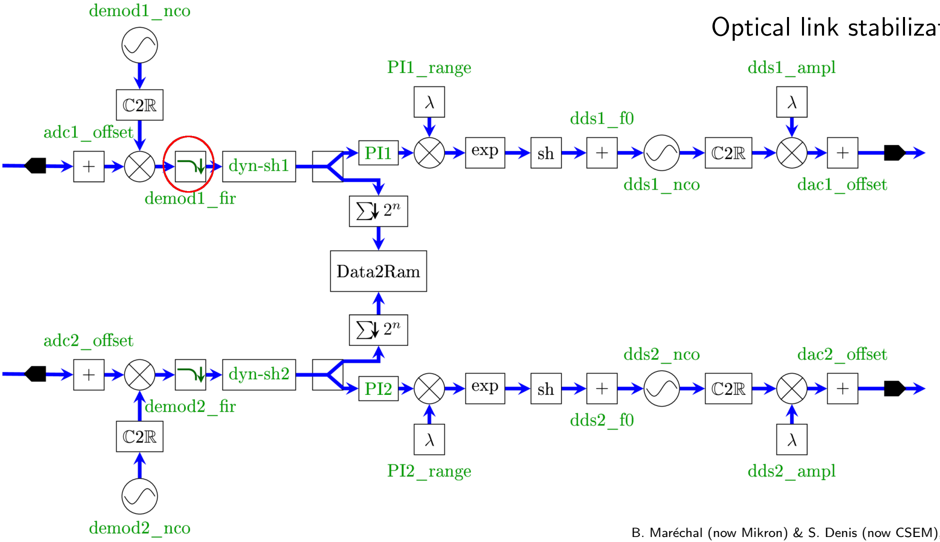
Left: 14-bit Red Pitaya (Zynq 7010), right: 16-bit Red Pitaya (Zynq 7020) ; top: FIR magnitude, bottom: unwrapped phase:



L. Tranchart, FEMTO-ST, Besançon, France

Closed loop control delay

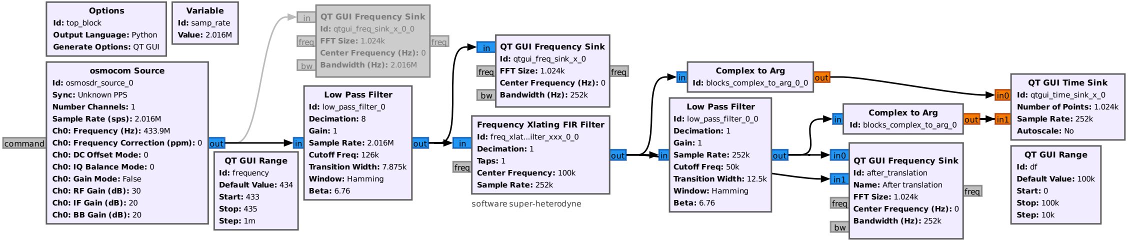
Optical link stabilization control loop:

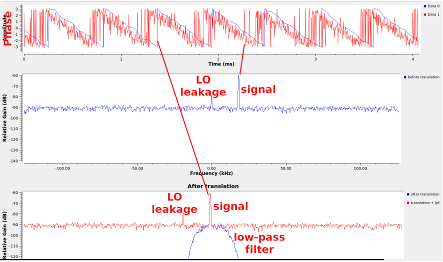


B. Maréchal (now Mikron) & S. Denis (now CSEM), FEMTO-ST, Besançon, France

https://github.com/oscimp/app/tree/f6e739e13248a8ec5a39b4fede420d57ed3552aa/redpitaya/double_iq_pid_vco

Drawbacks of digital: LO leakage





- LO leakage of the homodyne receiver[8]: super heterodyne [9]solution by introducting an intermediate frequency (historically designed to avoid amplifier oscillation)
- `Xlating FIR Filter` introduces intermediate frequency

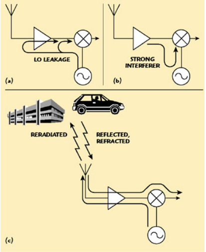


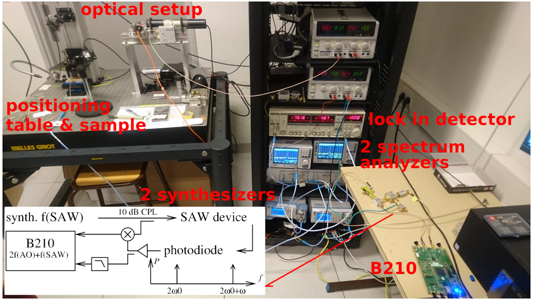
Fig. 8 DC offset mechanisms.

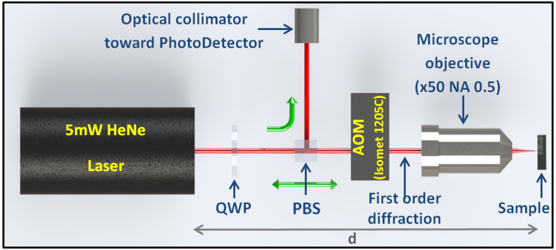
[9]A. Mashhour, W. Domino, N. Beamish, *On the Direct Conversion Receiver – A Tutorial*, Microwave Journal (2001), at `http://www.microwavejournal.com/articles/3226-on-the-direct-conversion-receiver-a-tutorial`

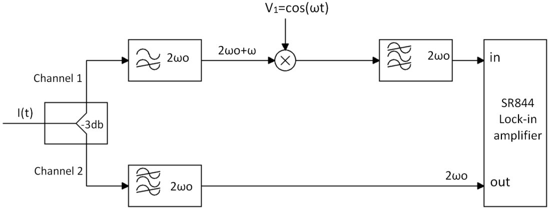
[9]E.H. Armstrong, *A new system of short wave amplification*, Proc. IRE. **9** (1), 3-–11 (1921)

SDR instrumentation: acoustic wave field mapping [10]

- $LO_{RX} \simeq 2\omega_0 + \omega$ on both channels CH_1 and CH_2 – ω_0 acousto-optic modulator frequency and ω SAW frequency
- send $|CH_1|$, $|CH_2|$, $|CH_1/CH_2|$, $arg(CH_1/CH_2)$
- Replace dedicated laboratory hardware with general purpose SDR receiver (Ettus Research **B210**)







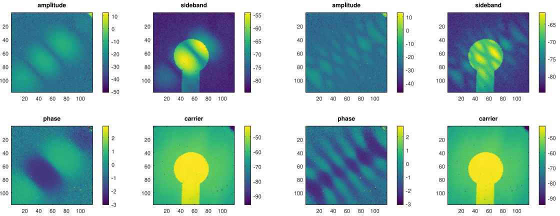
D. Teyssieux, FEMTO-ST, Besançon, France

- Flexibility: **70–6000** MHz RF frontend (AD9361)
- **Coherent dual** input (get rid of LO contribution between both channels)
- Software processing with GNU Radio: challenge of **synchronizing** positioning table & sampling (0-MQ)

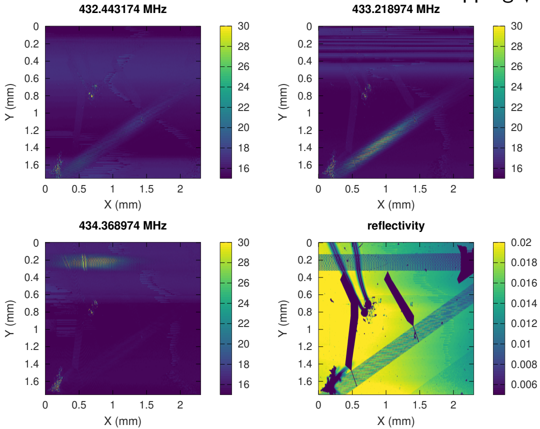
[10] D. Teyssieux *& al.*, *Absolute phase and amplitude mapping of surface acoustic wave fields*, Proc. IFCS (2013)

SDR instrumentation: acoustic wave field mapping

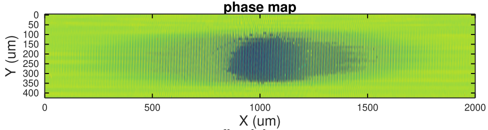
← Bulk acoustic resonator overtone out of plane acoustic field mapping

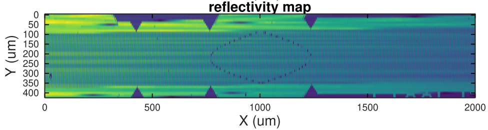


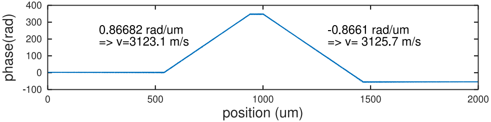
dual resonator SAW sensor acoustic field mapping ↓



← 434 MHz quartz resonator, 0.4 × 2 mm mapped 6.16×10^5 samples in 15100 s or 24.5 ms/sample limited by positioning table stabilization

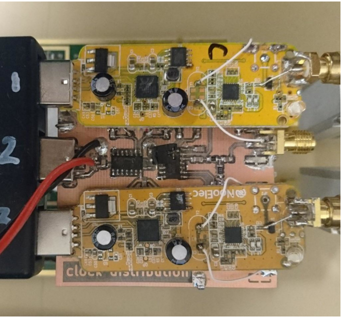


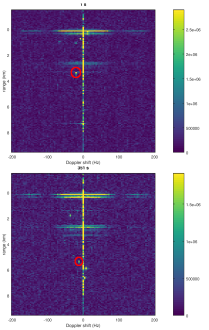




Passive RADAR: time & frequency maps

- Benefit from existing radiofrequency signals for detecting (moving) targets
- One reference channel (non-cooperative source) and one surveillance channel (targets)
- Doppler induced frequency shift + time delayed echoes = time-frequency (distance-velocity) maps
- only short-term (ms) phase coherence needed
- multi-DVB-T general purpose SDR demonstration [11] [12]





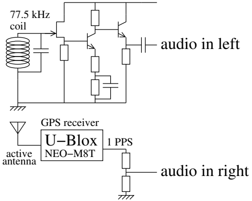
- challenge of random delay between channels (constant as long as datastream remains continuous): UDP stream
- frequency stacking for improved range resolution and surpass acquisition bandwidth f_s limitation

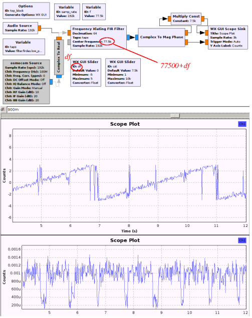
[11] W. Feng, J.-M Friedt, G. Cherniak, M. Sato, *Passive bistatic radar using digital video broadcasting–terrestrial receivers as general-purpose software-defined radio receivers*, Rev. Sci. Instrum. **89** 104701 (2018)

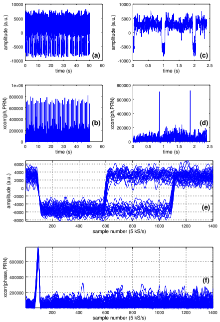
[12] J.-M Friedt *& al.*, *(Yet another) passive RADAR using DVB-T receiver and SDR*, FOSDEM 2018

Sound card for time of flight measurement

- Long range time dissemination: very low frequency signals bouncing off the ionosphere
- DCF77[13](77.5 kHz), MSF (60 kHz), eLORAN (100 kHz), TDF (162 kHz) ...
- again the broader the signal bandwidth [14], the better the timing accuracy (100 μs resolution @ VLF)







[14] J.-M Friedt, C. Eustache, É. Carry, E. Rubiola, *Software defined radio decoding of DCF77: time and frequency dissemination with a sound card*, Radio Science **53**(1) 48–61 (2018)

[14] P. Hetzel, *Time dissemination via the LF transmitter DCF77 using a pseudo-random phase-shift keying of the carrier*, 2nd EFTF 351-–364 (1988)

Beyond analog ...

- FM demodulation example

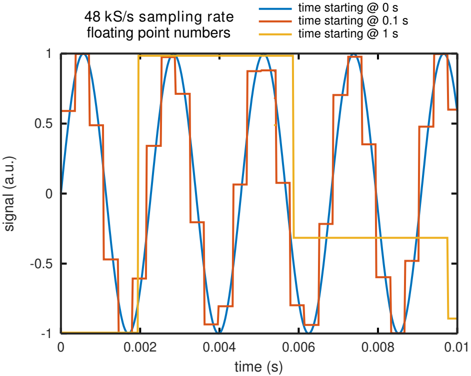
Beyond analog ...

Be aware of digital calculation limitations, especially with floating point number representation

- Discrete time: `t=[0:∞]'/f_s;` at sampling rate f_s
- Numerically controlled oscillator [15]: `lo=exp(j*2*pi*f*t);`
- Frequency transposition: `st=s.*lo;`

Example: keep trigonometric arguments in the $[-\pi : \pi]$ range where precision is maximized [16]

```
// increment current phase angle
void step(int n = 1)
{
  phase += phase_inc * n;
  if (fabs(phase) > GR_M_PI) {
     while (phase > GR_M_PI)
       phase -= 2 * GR_M_PI;

     while (phase < -GR_M_PI)
       phase += 2 * GR_M_PI;
  }
}

void nco<o_type, i_type>::sin(float* output, int noutput_items, double ampl)
{
    for (int i = 0; i < noutput_items; i++) {
        output[i] = (float)(sin() * ampl);
        step();
    }
}
```

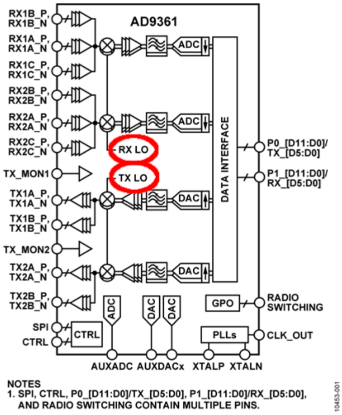


[15] transpose time to make a vector, or make sure to transpose `lo` with `.'` to avoid using the complex conjugate of `lo`

[16] `https://github.com/gnuradio/gnuradio/blob/master/gnuradio-runtime/include/gnuradio/nco.h#L50`

Hardware

- Open: HackRF, BladeRF, LimeSDR, Ettus Research hardware
- Cost: DVB-T receivers (<10 euros/$), ADi PlutoSDR AD936x **input and output are not coherent !** (different LO)
- VLF: computer sound card
- Size: Fairwaves XTRX (30 $\times$ 51 mm)
- General purspose: radiofrequency grade oscilloscope! (discontinuous stream but matches the ideal SDR definition)[17]



[17]https://github.com/jmfriedt/gr-oscilloscope38

Conclusion

Software Defined Radio for time& frequency analysis

- benefits of stability, flexibility and reconfigurability
- invest in hardware once, deploy for most investigations by tuning software
- challenging software combination (FPGA HDL, GP-CPU C++/Python, user interface & networking)

Not addressed in this presentation: SDR for **educational purposes** at the intersection between computer science, radiofrequency and digital signal processing

Selected bibliography

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

FOSDEM