Software defined radio based Synthetic Aperture noise and OFDM (WiFi) RADAR mapping

J.-M Friedt[1], W. Feng[2]

[1] FEMTO-ST/Time & Frequency department, Besançon, France
[2] National Laboratory of Radar Signal Processing, Xidian University, Xi'an, China

jmfriedt@femto-st.fr

Slides at `http://jmfriedt.free.fr/grcon2020_radar.pdf`

sequel to `http://jmfriedt.free.fr/sdra_radar.pdf` SDRA2020

August 15, 2020

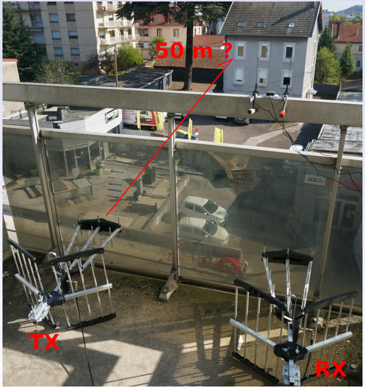
Why a SDR-based RADAR ?

Can we range the house opposite to our balcony using available hardware, namely

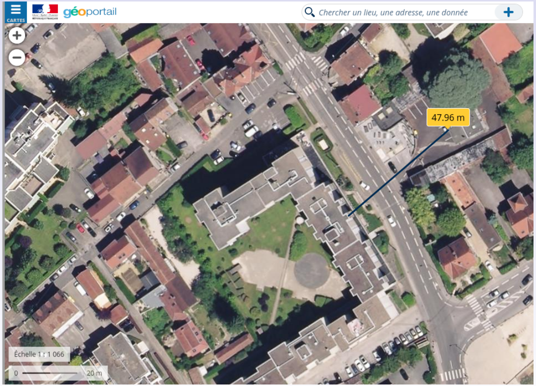
- two DVB-T and two WiFi antennas
- PlutoSDR + Ettus Research B210
- splitter and attenuator

$\Rightarrow$ OFDM or noise, frequency swept RADAR[1]

[1] RAdio Detection And Ranging



View from the balcony



Aerial map of the area: balcony to house=48 m ?

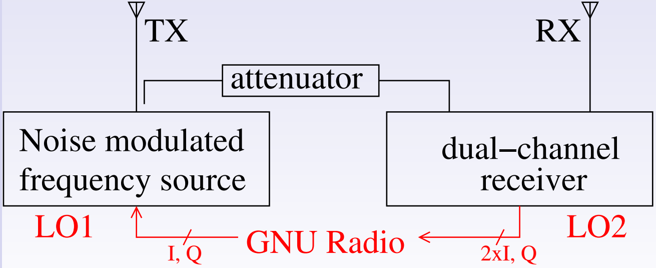
French Geographic Institute (`geoportail`) claims 48 m: is that correct ?

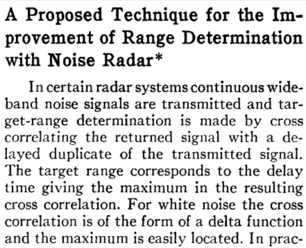
RADAR design – general principles

Spectrum spreading the emission: $\Delta R = c/(2B)$

- could be chirp (LO frequency sweep),
- OFDM[1]/WiFi (sum of well defined adjacent frequency components),
- noise RADAR: pseudo random phase fluctuation while keeping amplitude constant (see GPS/CDMA for PRN phase spread spectrum)
- Correlate reference signal with measured (delayed) signal=sum of echoes:

$$\boxed{FT(xcorr(r,s)) = FT(r)\cdot FT^*(s) \sim FT(r)/FT(s)}$$





A Proposed Technique for the Improvement of Range Determination with Noise Radar*

In certain radar systems continuous wide-band noise signals are transmitted and target-range determination is made by cross correlating the returned signal with a delayed duplicate of the transmitted signal. The target range corresponds to the delay time giving the maximum in the resulting cross correlation. For white noise the cross correlation is of the form of a delta function and the maximum is easily located. In prac-

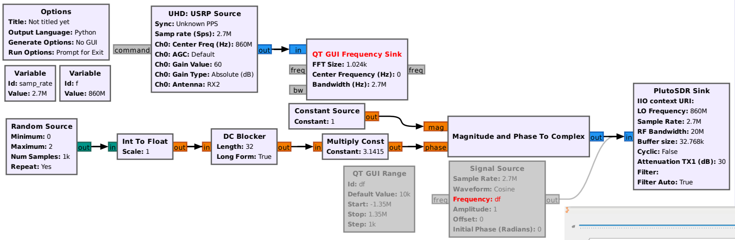
R. Bourret, *A proposed technique for the improvement of range determination with noise radar*, Proc IRE **45** (12) 1744–1744 (1957)

[1] M. Braun, *OFDM Radar Algorithms in Mobile Communication Networks* (2015) at `publikationen.bibliothek.kit.edu/1000038892/2987095`

RADAR design – general principles

Spectrum spreading the emission: $\Delta R = c/(2B)$

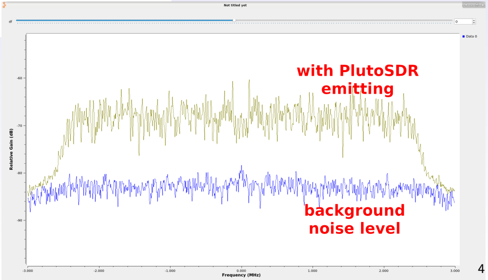
- noise RADAR: pseudo random phase fluctuation while keeping amplitude constant (see GPS/CDMA for PRN phase spread spectrum)



Fixed magnitude (power) and pseudo-random definition of phase in the Magnitude and Phase to Complex block

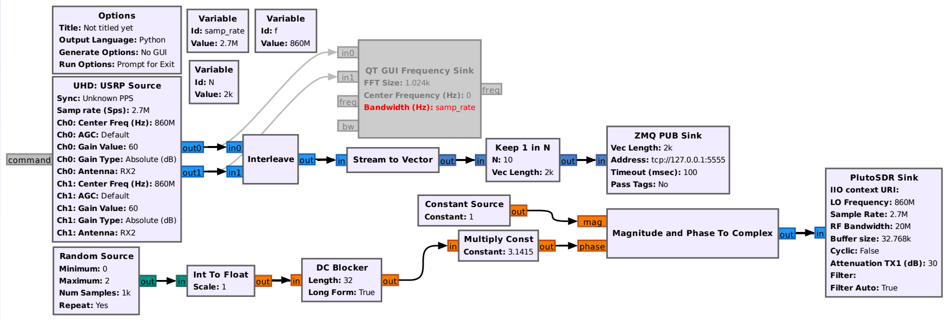
↑ Flowchart for checking the spectrum spreading capability of the PlutoSDR and simultaneous data collection by the B210 ($\Rightarrow$ max samp_rate with no sample loss)

Spectrum collected by the B210 ⟶



RADAR design – GNU Radio implementation

B limited by USB to about 2.7 MHz=56 m range resolution:
sweep LO to concatenate multiple spectra and extend B following multiple sequential sweeps [1]



- Collect time series → program next LO frequency → repeat until full B has been swept
- No known way of synchronizing data collection with LO sweep in GNU Radio Companion

⇒ benefit from external data collection and processing program (GNU Octave): streaming using ZeroMQ from GNU Radio to Octave

[1] S. Prager *& al.*, *Ultrawideband Synthesis for High-Range-Resolution Software-Defined Radar*, IEEE Trans. Instrumentation and Measurement **69**(6), 3789-3803 (2019): "frequency stacking"

RADAR design – GNU Radio implementation [2]

Python generated by GNU Radio Companion

```
#!/usr/bin/env python3
class t(gr.top_block)
[...]
    def set_f(self, f):
        self.f = f
        self.iio_pluto_sink_0.set_params(self.f, self.→
             ↪samp_rate, 20000000, 30.0, '', True)
        self.uhd_usrp_source_0.set_center_freq(self.f, 0)
        self.uhd_usrp_source_0.set_center_freq(self.f, 1)

def main(top_block_cls=t, options=None):
    tb = top_block_cls()
    def sig_handler(sig=None, frame=None):
        tb.stop()
        tb.wait()
        sys.exit(0)

    signal.signal(signal.SIGINT, sig_handler)
    signal.signal(signal.SIGTERM, sig_handler)

    tb.start()
```

Callback functions defining PlutoSDR and B210 LO frequency `f`

GNU Octave & ZeroMQ toolbox

```
total_length=140000; % [samples/measurement]
Nfreq=50;
pkg load zeromq
for frequency=1:Nfreq
 sock1 = zmq_socket(ZMQ_SUB);
 zmq_connect   (sock1,"tcp://127.0.0.1:5555");
 zmq_setsockopt(sock1, ZMQ_SUBSCRIBE, "");
 recv=zmq_recv(sock1, total_length*8*2, 0);
 value=typecast(recv,"single complex");
 x(:,frequency)=value(1:2:length(value));
 m(:,frequency)=value(2:2:length(value));
 zmq_close (sock1);
end
```

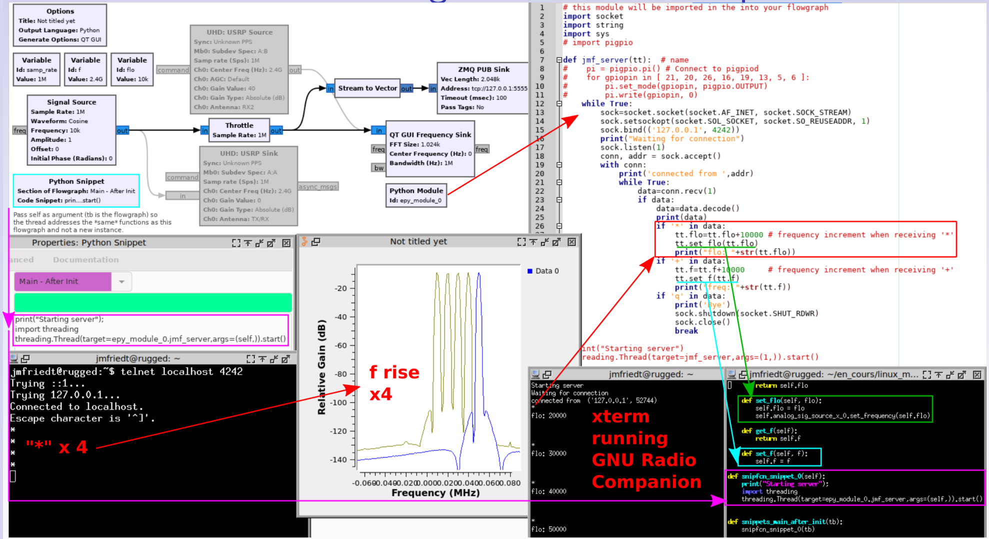
- `zmq_recv`: number of bytes (*8 complex, *2 interleaved channels)
- `typecast`: char → complex float
- socket-connect-opt-close = 130 us

TCP client: `sockets` toolbox

```
pkg load sockets
sc=socket(AF_INET, SOCK_STREAM, 0);
s=struct("addr","127.0.0.1","port",4242);
connect(sc,s);
for frequency=1:Nfreq
 [...]
 send(sc,'+'); % wait PlutoSDR
 pause(1.0)    % to stabilize
end
send(sc,'q');
```

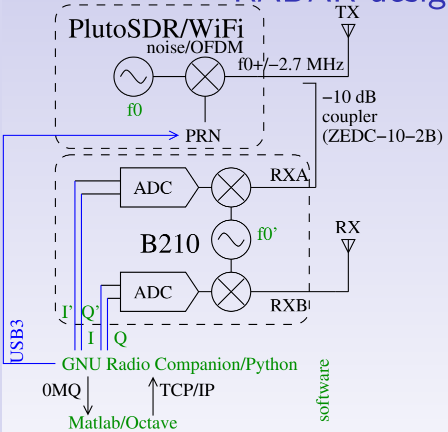
[2] J.-M. Friedt, W. Feng, *Noise RADAR implementation using software defined radio hardware*, SDRA 2020

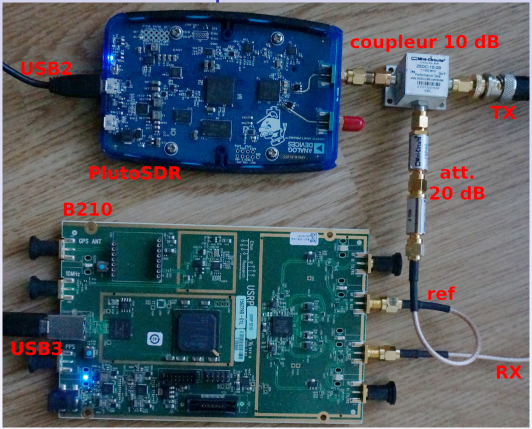
RADAR design – GNU Radio implementation



Python Module in GNU Radio Companion to add TCP server receiving commands from Octave (LO reset, LO increment, antenna position with PyGPIO)

RADAR design – GNU Radio implementation

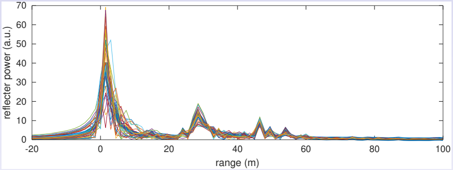




- ⇒ functional frequency sweep ...
- ... but need to **wait** for unknown delay between LO programming and settling ⇒ 1 s between programming the PlutoSDR [3] and recording
- if LO not settled, no pseudo-random modulation ⇒ single carrier carrying all energy with no range resolution (threshold test)

Range measurement (noise, 2450±50 MHz)

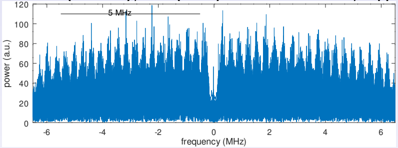
- Well resolved multiple echoes: 24, 28.5, 46.5, 49.5, 54 m (30 measurements, bold=average)

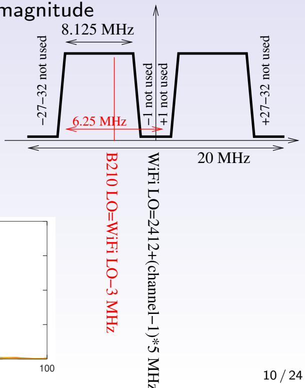


- However, hard to identify the source of the reflections:
- $\simeq$50 m is **probably** associated with the house/roof (red arrow)
- $\simeq$25 m **probably** associated with the nearby covered parking boxes (yellow arrow)

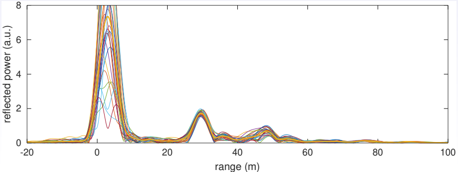
Range measurement (WiFi, ch 1 to 11=55 MHz)

- 20 MHz-wide OFDM with 64 sub-carrier structure: unused 0 (avoid DC after downconversion), -32 to -27 and 27 to 32
- 20 MHz/64 = 0.3125 MHz subcarrier spacing
- USB bandwidth: offset RX LO by 3 MHz wrt WiFi center TX frequency, sample at 6.25 MS/s, keep 5 MHz
- OFDM structure creates sidelobes: replace $FT(meas) \cdot FT(ref)^*$ with $FT(meas)/FT(ref)$ to cancel varying magnitude





- ability to combine digital communication with ranging:



Azimuth measurement

Wikipedia: ASR-9

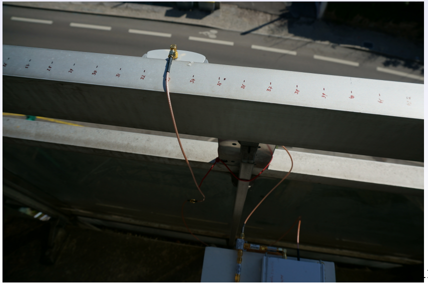
- Focus narrow beam in a known direction to identify azimuth of target
- Beamwidth $\propto$ wavelength λ/antenna diameter D
- Antenna array: replace single wide antenna with array of small antennas
- Synthetic Aperture RADAR (SAR): simulate D my moving a single small antenna along a path of length D
- For an N antenna array with elements separated by distance d:

$$\vartheta_{3dB} = \frac{0.89\lambda}{Nd} = \frac{0.89 \times 0.12}{1.40} = 80 \text{ mrad} = 4.5^\circ$$

or 8 m azimuth resolution at 100 m at 2.45 GHz ($\lambda = 12$ cm) with rail length 1.4 m

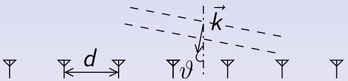
Replace one-dimension analysis (range compression by correlation) with two-dimensional analysis (range and azimuth compression)

Scale for positioning the antenna every $\lambda/4$ $\rightarrow$



Signal processing basics

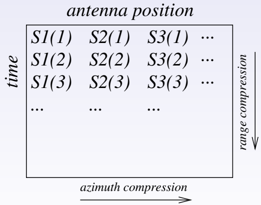
- Uniform Linear Array: a plane wave reaches each antenna with an additional phase delay $\vec{k}\cdot\vec{d} = \frac{2\pi}{\lambda} d \sin(\vartheta)$
- For the nth antenna, the received time series $S_t(nd)$ has become



$$S_t(nd)\cdot\exp\left(j\frac{2\pi nd}{\lambda}\right)$$

which can be read as the Fourier transform of S_t with the phase being the translation

- $\Rightarrow$ azimuth compression is inverse Fourier Transform (FT) along the antenna position
- Cross-correlation is inverse FT of product of transmitted signal FT with complex conjugate received signal FT
- $\Rightarrow$ range/azimuth compression is inverse Fourier transform of the matrix with column=time series and line=antenna position
- **range resolution** only determined by signal **bandwidth**
- **azimuth resolution** only determined by antenna **displacement range**: no degree of freedom in mapping to real scale [4]

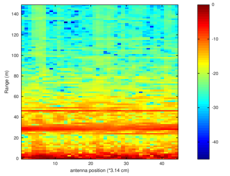


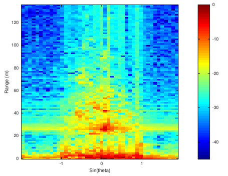
[4]see last appendix slide for antenna motion step demonstration meeting the sampling theorem

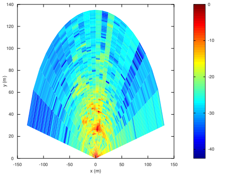
Full demonstration

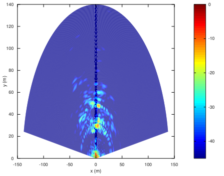
1. 2D-FT of $\exp(j2\pi f_y \cdot y_0) \times \exp(j2\pi f_x \cdot x_0) \xrightarrow{\text{2D FT}} \delta(\underbrace{x_0, y_0}_{target})$ so[5] we must separate f_q (frequency $\rightarrow$ range r_0) and x_p (antenna position $\rightarrow$ azimuth ϑ_0)
2. $s(p,q) \underbrace{\propto}_{RCS} \exp\left(j\frac{4\pi}{c}f_q \cdot R_p(\underbrace{r_0, \vartheta_0}_{target})\right)$ with $R_p = \sqrt{(x_p - r_0\sin\vartheta_0)^2 + (r_0\cos\vartheta_0)^2}$ where $x_0 = r_0 \sin\vartheta_0$ and $y_0 = r_0\cos\vartheta_0$ (cartesian $\rightarrow$ polar coordinates)
3. $\frac{\partial}{\partial x_p}\left(\sqrt{(x_p-a)^2+b^2}\right) = \frac{x_p - a}{\sqrt{(x_p-a)^2+b^2}} = \frac{-a}{\sqrt{a^2+b^2}}$ at $x_p \simeq 0 \Rightarrow$ Taylor expansion $R_p \simeq r_0 - x_p \sin\vartheta_0$ since $a^2+b^2 = r_0^2$
4. $s(p,q) \propto \exp\left(j\frac{4\pi}{c}f_q \cdot (r_0 - x_p\sin\vartheta_0)\right) \simeq \exp\left(j2\pi(\underbrace{2f_q \cdot r_0/c}_{range\ \alpha} - \underbrace{2x_p\sin\vartheta_0/\lambda_c}_{azimuth\ \beta})\right)$ assuming that $f_q/c = 1/\lambda \simeq 1/\lambda_c$ the wavelength at center frequency since $\frac{1}{x} = \sum_n (-1)^n \cdot (x-1)^n \simeq 1$ around $x \simeq 1$ (keep only $n=0$)
5. $r_0 = \alpha \times c/(2f_q)$ and $\sin\theta_0 = \beta/(2\lambda_c)$ in polar coordinates or
6. $x_0 = r_0\sin\vartheta_0 = \alpha\beta \cdot c \cdot \lambda_c/4$ and $y_0 = r_0\cos\vartheta_0 = c\alpha/2\cos(\mathrm{asin}(\lambda_c\beta/2))$ in cartesian coordinates: conversion from (f_q, x_p) to (x_0, y_0) using 2D-FT thanks to variable separation.

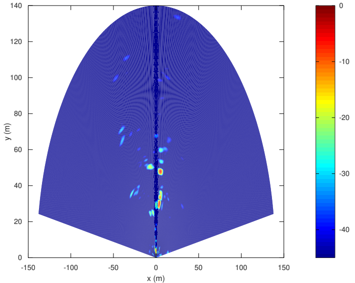
[5] https://hforsten.com/synthetic-aperture-radar-imaging.html











Azimuth compression

Map {angle-range} → X-Y:

```
Nf=100      % number of frequencies
Na=73       % number of ant. pos,
c=3e8       % speed of light
f=2.45e9    % center frequency (azimuth compression)
df=1e6;     % freq. sweep step => total bw is 100*1=100 MHz
lambda=c/f
dx=lambda/4; % antenna moved by quarter wavelength steps

fs_r = 1/df;
r = (0:Nf-1)*fs_r/Nf*c/2;

fs_a = 1/dx;
alpha = (0:Na-1)*fs_a/Na-fs_a/2;
sin_thta=alpha*lambda/2;

[R,ST]=meshgrid(r,sin_thta(abs(sin_thta)<=1));
X = R.*ST;Y = R.*sqrt(1-ST.^2);
Z=Img_focus(:,(abs(sin_thta)<=1));
pcolor(X.',Y.',10*log10(Z));
```

Top left: range-position map

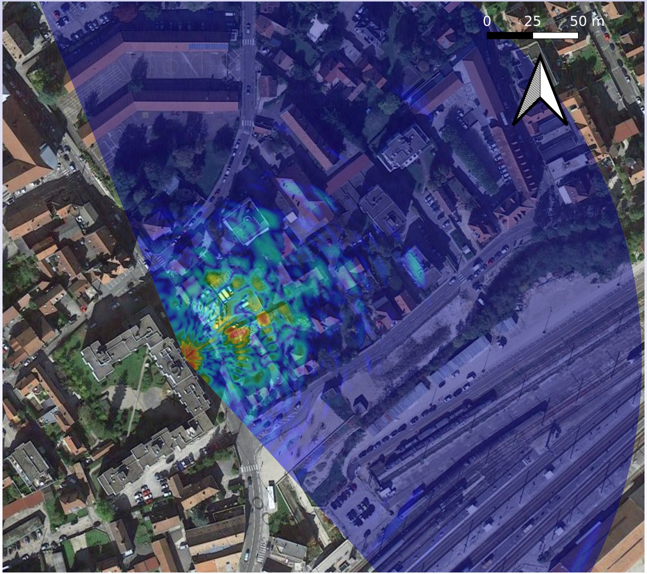
Top right: range-angle map (iFFT along antenna pos.)

Middle left: range-azimuth map

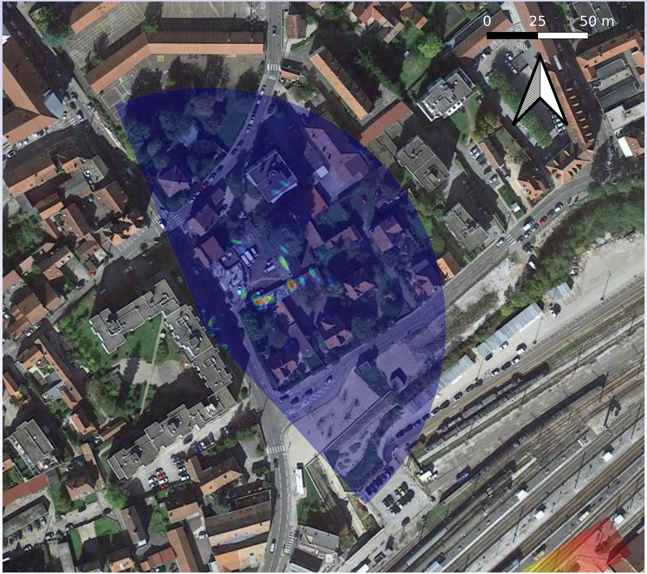
Middle right: range-azimuth backprojection

Bottom: range-azimuth backprojection with windowing

Azimuth compression

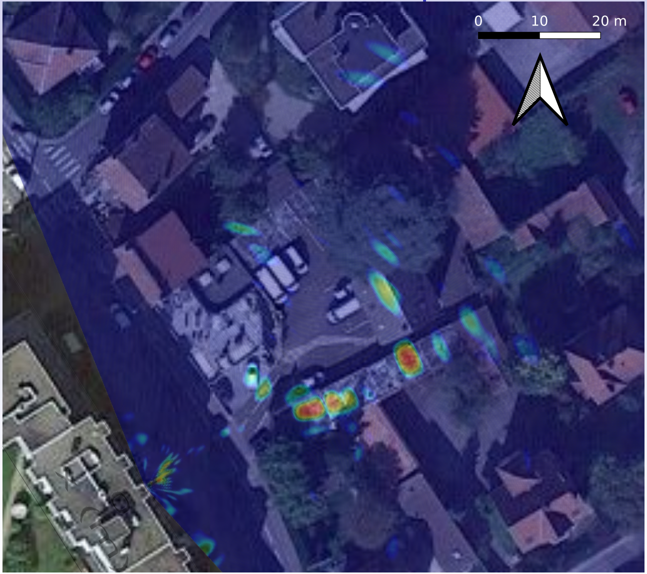


Azimuth compression

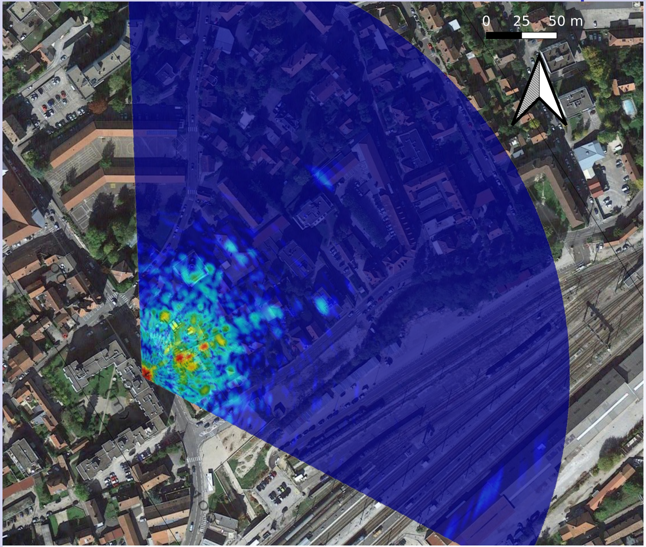


Azimuth compression

- Targets match geographic features: roof edges & cars
- No freedom in scaling image:
    - azimuth fully defined by Nd/λ + orientation of balcony rail
    - range given by frequency span
- QGis *Freehand raster georeferencer* plugin to scale, rotation and translate raster picture over georeferenced aerial images (Google Maps)



Azimuth compression (WiFi emitter)



- Replace PlutoSDR configured as noise generator with WiFi emitter
- Each WiFi channel is 20 MHz wide with center OFDM sub-carrier unused
- Frequency offset between emitted and received signal: 3 MHz
- Sampling rate : 5 MHz
- Scan 11 channels with 5 MHz steps: 55 MHz=3 m range resolution
- WiFi in Monitor mode & continuous emission using B. Bloessl's PacketSpammer [6]

[6] `github.com/bastibl/gr-ieee802-11/tree/maint-3.8/utils/packetspammer`

Interferometric displacement measurement (noise InSAR)

- Assuming the antenna is always positioned at the same location, the phase to the target is a fine ($\ll \lambda$) indicator of its position:

$$\varphi = 2\vec{k}\vec{r} = \frac{4\pi r}{\lambda_c}\cos\vartheta_0 = \frac{4\pi r \cdot f_c}{c}\cos\vartheta_0$$

$\Rightarrow$ varying $\vec{r}$ varies φ with sub-λ resolution BUT $\lambda/2$ (π) uncertainty

- Known moving target: corner reflector [7]

Static target (roof)

30 cm corner reflector

Positioned corner reflector

[7] Corner reflector and rail fabricated by P. Abbé (FEMTO-ST/Time & Frequency, Besançon, France)

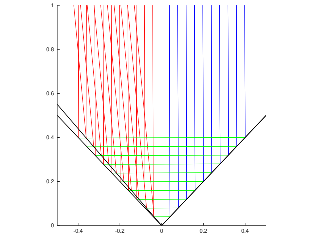
Interferometric displacement measurement (noise InSAR)

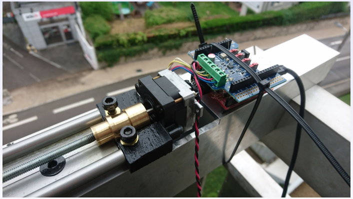
- Assuming the antenna is always positioned at the same location, the phase to the target is a fine ($\ll \lambda$) indicator of its position:

$$\varphi = 2\vec{k}\vec{r} = \frac{4\pi r}{\lambda_c}\cos\vartheta_0 = \frac{4\pi r \cdot f_c}{c}\cos\vartheta_0$$

$\Rightarrow$ varying $\vec{r}$ varies φ with sub-λ resolution BUT $\lambda/2$ (π) uncertainty

- Known moving target: corner reflector [7]



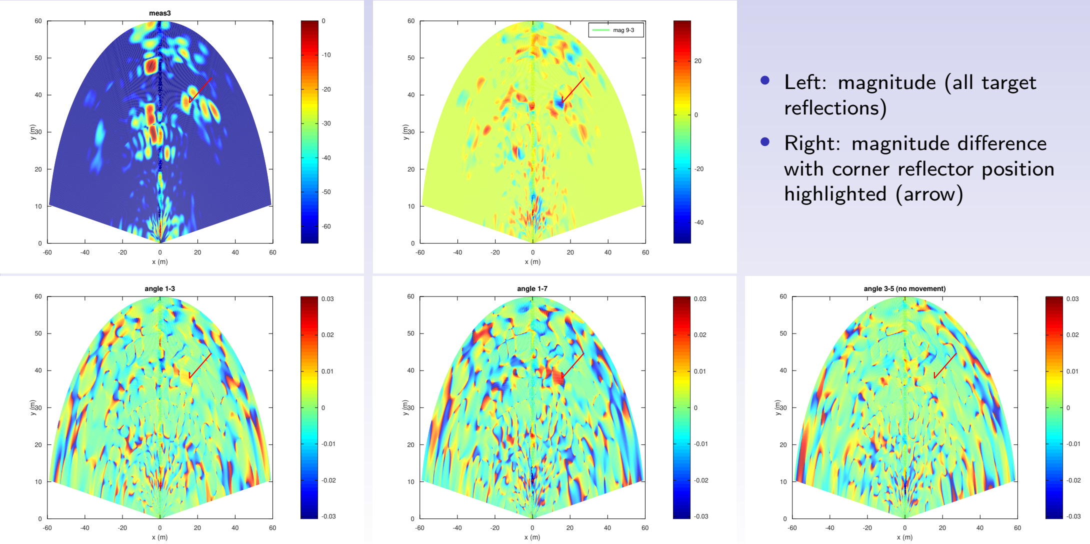


Rail for automated scanning — 90 v.s 95° angle — Motor

Microcontroller for motor control receiving cmds from computer (could be Raspberry Pi4) – 1 h/1.4 m sweep (3.2 cm/step: 30 min displacement/30 min acquisition)

[7] Corner reflector and rail fabricated by P. Abbé (FEMTO-ST/Time & Frequency, Besançon, France)

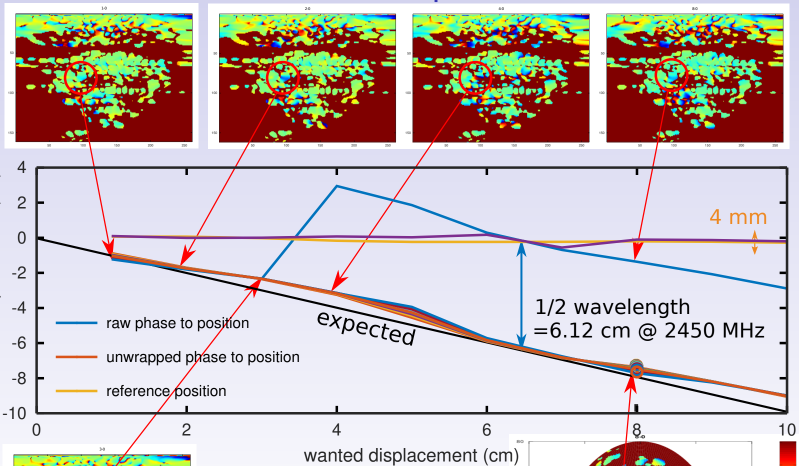
Interferometric displacement measurement

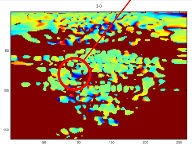


- Left: magnitude (all target reflections)
- Right: magnitude difference with corner reflector position highlighted (arrow)

Phase difference with arrow indicating corner reflector location: 1 cm, 2 cm and 0 cm

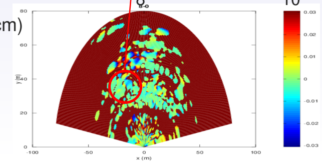
Interferometric displacement measurement





Corner reflector range: 40 m

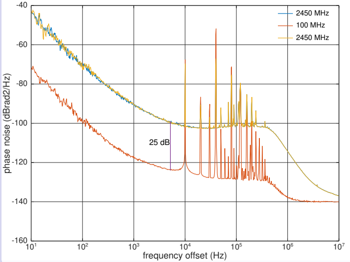
Reference=phase at roof position (47.5 m range)

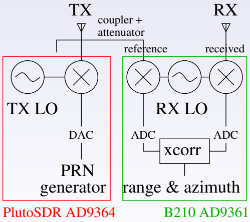


Tentative error budget (4 mm/day)

Electronics

- Impact of local oscillator phase noise: TX LO is cancelled but time delay between reference and measured signals allows for phase fluctuations.
- Minimum delay given by sampling rate ($1/b1 = 1/2.7$ MHz)
- Maximum delay given by correlation integration duration ($1/b2 = 20$ ms)
- Integrate oscillator phase noise (measured at *output* of AD9361) over this frequency offset interval



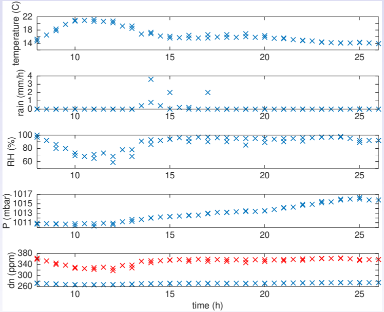


$$\sigma_\varphi = \sqrt{\int_{b1}^{b2} S_\varphi(f)df} \simeq 1.3^\circ \Rightarrow dr = \frac{\lambda}{720} d\varphi = 0.1 \text{ mm}$$

Weather/propagation

Impact of temperature and moisture on velocity of electromagnetic wave in air

$$N = (n-1)_{ppm} = 77,6\frac{p}{T} - 6\frac{p_e}{T} + 3,75\cdot10^5\frac{p_e}{T^2}$$

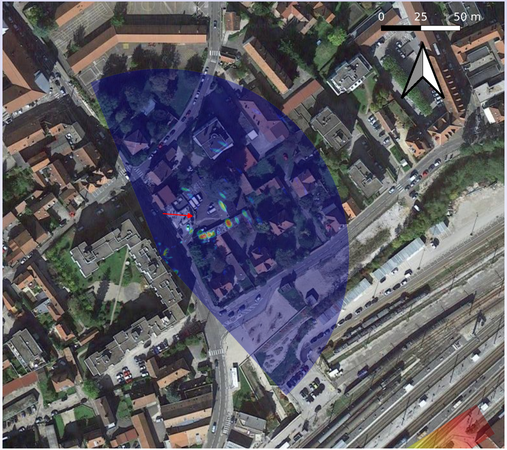


Ambiant pressure (1013 hPa=1013 mbar): temperature variation $dK_1 = 77,6 \times p/T = 77.6 \times 1013/273 = 287$ ppm and $dK_1/dT = 287/T \simeq 1$ ppm/K around ambiant temperature. 10 ppm=0.6 mm @ 60 m

Conclusion & perspective

Software Defined Radio for active RADAR prototyping (B210 receiver, PlutoSDR or WiFi emitter with carrier frequency sweep)

- **separate functions** and delegate to most efficient framework (GNU Radio, Python, Octave)
- Extended range measurement to azimuth measurement by implementing **Synthetic Aperture RADAR** (SAR) processing
- Extended SAR processing to **interferometric** displacement measurement on corner reflector
- 4-mm baseline stability on displacement measurement, mm accuracy with departure from expected displacement attributed to poor corner reflector positioning & weather
- github.com/jmfriedt/active_radar



Perspective: can we apply this knowledge to Copernicus satellite (Sentinel-1) datasets ?
https://scihub.copernicus.eu/dhus/

Antenna displacement step

Assumption: TX antenna is fixed and RX antenna is moved with x_p step:

1. Received signal phase is $\varphi = 2 \times \frac{2\pi f_c R}{c}$ where $R = \sqrt{(x_p - x_0)^2 + y_0^2}$
2. We have discussed far field $R \simeq r_0 + x_p \cdot \sin\vartheta$
3. $\Rightarrow \varphi \simeq \underbrace{2 \times 2\pi f_c \cdot r_0/c}_{static} + 2 \times 2\pi \underbrace{f_c/c}_{\lambda_c^{-1}} \cdot x_p \cdot \sin\vartheta_0$
4. avoid φ ambiguity with $2x_p/\lambda_c < 1 \Leftrightarrow$

$$\boxed{x_p < \lambda_c/2}$$

Notice that if **both TX and RX move** (as opposed to keeping TX fixed and moving RX), then

$$\boxed{x_p < \lambda_c/4}$$

Spatial equivalent to sampling theorem