Software defined radio based Global Navigation Satellite System real time spoofing detection and cancellation

J.-M Friedt, D. Rabus, G. Goavec-Merou

FEMTO-ST/Time & Frequency department, Besançon, France

jmfriedt@femto-st.fr

Slides at `http://jmfriedt.free.fr/grcon2020_gps.pdf`

sequel to FOSDEM2019: `https://archive.fosdem.org/2019/schedule/event/sdr_gps/`

August 15, 2020

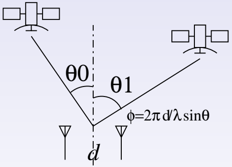
SDR based GNSS receiver

- UK study [1]: GPS disruption = 1 billion pounds/day
- Notifications of GPS jamming exercices: affect up to 112 km range at 30000 feet AMSL https://www.ofcom.org.uk/spectrum/information/gps-jamming-exercises
- Demonstrated[2] GPS spoofing using pluto-gps-sim (https://github.com/Mictronics/pluto-gps-sim) with a proper reference clock.

⇒ spoofing detection & mitigation solutions: technology can be fooled, **physics** much harder

Software Defined Radio approach: access raw I/Q physical characteristics

- Initially tested with low-cost DVB-T receivers (1.023 MHz bandwidth, sample at 1.123 MS/s complex)
- **Direction of arrival** measurement: dual channel coherent receiver AD9361 as radiofrequency frontend of the Ettus Research B210



[1] London Economics, *The economic impact on the UK of a disruption to GNSS* (June 2017) at https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/619544/17.3254_Economic_impact_to_UK_of_a_disruption_to_GNSS_-_Full_Report.pdf

[2] G. Goavec-Merou, J.-M Friedt, F. Meyer, *Spoofing GPS – is it really the time we think it is, and are we really where we think we are ?*, FOSDEM 2019

PUBLIC MONEY
PUBLIC CODE

Spoofing detection

Assumption: a genuine satellite constellation is distributed in space, while a single spoofer will show all satellites located at the same place.

Computationnally efficient codeless processing: antenna a receives satellite i signal

$$s_a(t) = A_a(t)\exp(j(\underbrace{\delta\omega_i t}_{Doppler} + \underbrace{\varphi_{PRN,i}}_{BPSK} + \underbrace{\varphi_{a,i}}_{geometric}))$$

with Doppler shift $\delta\omega_i$, BPSK modulated code $\varphi_{PRN,i} \in [0,\pi]$, geometrical phase $\varphi_{a,i}$

$$\Rightarrow s_a^2(t) = A_a^2(t)\exp(j(2\delta\omega_i t + 2\varphi_{a,i}))$$

since $2\varphi_{PRN,i} \in [0,2\pi] = 0$

$$\Rightarrow \frac{FFT(s_1^2(t))}{FFT(s_2^2(t))} = \frac{A_1^2(t)}{A_2^2(t)}\exp(2j(\varphi_{1,i} - \varphi_{2,i})) \text{ computed at bin } \delta\omega_i \text{ of FFT}$$

If all $\varphi_{1,i} - \varphi_{2,i}$ $\forall i$ are **close: spoofing attack** occuring (standard deviation on $\varphi_1 - \varphi_2$)

Real time spoofing detection

If all phase differences $\varphi_{ant1,j} - \varphi_{ant2,j}\ \forall\ j$ close: all sources at same location = spoofing.

No spoofing (all `angle` different)

```
pos=936: angle=0.054160 - mag=3958. angle=0.047418
pos=1089: angle=1.676572 - mag=1183. angle=1.662526
pos=1099: angle=2.326184 - mag=118. angle=2.355659
pos=3: angle=-0.758771 - mag=1285. angle=-0.437436
pos=987: angle=-1.240077 - mag=11951. angle=-1.202965
pos=1049: angle=-2.437160 - mag=108. angle=-2.439323
pos=171: angle=-0.957273 - mag=291. angle=-0.657624
Current receiver time: 44 s
New GPS NAV message from satellite GPS PRN 12 (IIR-M)
New GPS NAV message from satellite GPS PRN 25 (IIF)
New GPS NAV message from satellite GPS PRN 32 (IIF)
New GPS NAV message from satellite GPS PRN 14 (IIR)
pos=936: angle=0.020945 - mag=3633. angle=0.015451
pos=1089: angle=1.669953 - mag=1792. angle=1.674325
pos=1099: angle=2.238233 - mag=89. angle=2.262804
pos=3: angle=-0.473037 - mag=959. angle=-0.490393
pos=987: angle=-1.549491 - mag=256. angle=1.402446
pos=1049: angle=-2.245952 - mag=40. angle=-2.155030
pos=1136: angle=0.412245 - mag=75. angle=0.546567
Current receiver time: 45 s
pos=936: angle=0.006864 - mag=3539. angle=0.006116
pos=1089: angle=1.667449 - mag=1654. angle=1.603043
pos=1099: angle=2.348760 - mag=99. angle=2.339042
pos=3: angle=-0.677564 - mag=582. angle=-0.541820
pos=987: angle=-1.190132 - mag=666. angle=-1.815525
pos=1049: angle=-2.388442 - mag=52. angle=-2.598973
pos=1136: angle=1.080694 - mag=82. angle=1.049687
pos=0: angle=0.334906 - mag=194. angle=0.829775
pos=237: angle=1.722518 - mag=397. angle=1.911799
```

With spoofing (`angle`$\simeq$ -0.45 rad)

```
pos=1011: angle=-0.479624 - mag=415. angle=-0.480036
pos=1166: angle=-0.462592 - mag=430. angle=-0.465509
pos=825: angle=-0.469689 - mag=404. angle=-0.467373
pos=1071: angle=-0.488331 - mag=429. angle=-0.483574
pos=964: angle=-0.458709 - mag=408. angle=-0.460694
pos=994: angle=-0.473683 - mag=434. angle=-0.472869
pos=22: angle=-0.472729 - mag=463. angle=-0.468133
pos=870: angle=-0.449495 - mag=442. angle=-0.434457
pos=795: angle=-0.543935 - mag=415. angle=-0.519226
pos=1014: angle=-0.490846 - mag=391. angle=-0.489497
pos=1008: angle=-0.544428 - mag=396. angle=-0.535370
pos=813: angle=-0.475915 - mag=416. angle=-0.447317
pos=1002: angle=-0.446111 - mag=377. angle=-0.423784
pos=12: angle=-0.375477 - mag=380. angle=-0.372678
pos=1017: angle=-0.386707 - mag=973. angle=-0.361714
GPS L1 C/A tracking bit synchronization locked GPS PRN 10 (IIF)
GPS L1 C/A tracking bit synchronization locked GPS PRN 17 (IIR-M)
GPS L1 C/A tracking bit synchronization locked GPS PRN 13 (IIR)
GPS L1 C/A tracking bit synchronization locked GPS PRN 15 (IIR-M)
GPS L1 C/A tracking bit synchronization locked GPS PRN 12 (IIR-M)
Current receiver time: 17 s
pos=1011: angle=-0.473406 - mag=430. angle=-0.473470
pos=1166: angle=-0.473563 - mag=437. angle=-0.473421
pos=825: angle=-0.475702 - mag=444. angle=-0.475105
pos=1071: angle=-0.478151 - mag=416. angle=-0.477901
pos=994: angle=-0.474488 - mag=428. angle=-0.473118
pos=964: angle=-0.466985 - mag=431. angle=-0.464038
pos=22: angle=-0.461783 - mag=430. angle=-0.463075
pos=870: angle=-0.451118 - mag=424. angle=-0.435480
pos=795: angle=-0.470265 - mag=426. angle=-0.475780
```

Spoofing cancellation

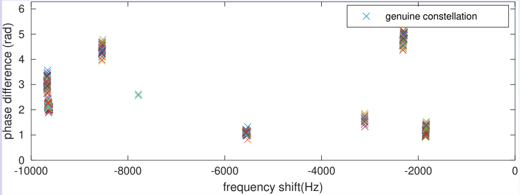
CRPA (Controlled Reception Pattern Antenna): null forming to cancel the spoofing signal in the direction of arrival (in our case, 2 antennas $\Rightarrow$ 1 null)

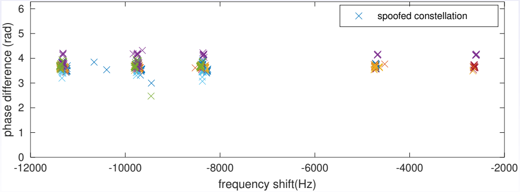
How to identify the weight of the signal detected on one antenna to subtract its contribution on the second antenna ?

$$\alpha = \left\langle \frac{FFT(s_1^2(t))}{FFT(s_2^2(t))} \right\rangle_i = \frac{A_1^2}{A_2^2}\exp(\varphi_{1,i} - \varphi_{2,i})$$

$\Rightarrow s_1 - \alpha \times s_2$ allows for recovering the genuine signal by cancelling spoofing signal

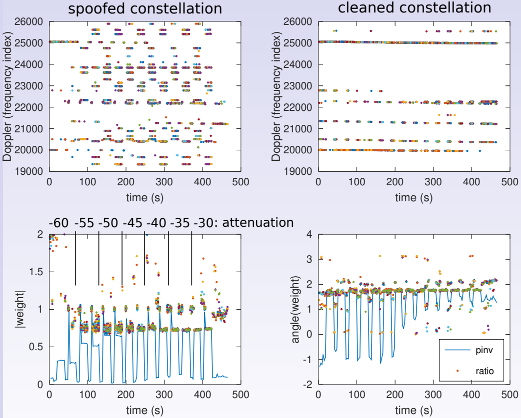
- average α over all spoofing satellites i whose $\varphi_{1,i} - \varphi_{2,i}$ is close to mean value

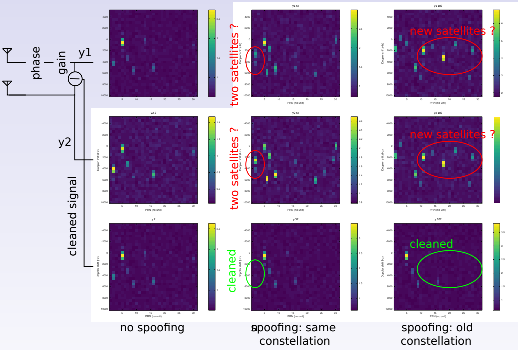




Post-processing spoofing cancellation: Doppler-PRN maps

Post-processing recorded data as spoofing power is increased [3]

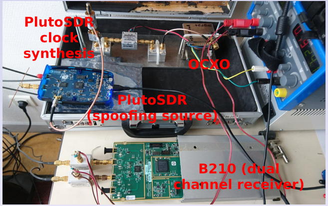




[3] indicated attenuation is with respect to the PlutoSDR output power measured as 0 dBm CW output when no attenuation is introduced

Post-processing spoofing cancellation: setup

- Demonstrate spoofing cancellation by **decoding** using `gnss-sdr` [4] (post-processing and real time)



- added a custom[5] Signal Source processing step for cancelling spoofing.

src/algorithms/signal_source/adapters/uhd_signal_source.cc

```
UhdSignalSource::UhdSignalSource(const →
    ↪ConfigurationInterface* configuration,
  const std::string& role, unsigned int in_stream, →
    ↪unsigned int out_stream, ...)
{...
 spoofing_detect_=gnss_sdr_make_spoof(item_size_, queue);
 top_block->connect(uhd_source_, i, spoofing_detect_, i);
 ...
}

gr::basic_block_sptr UhdSignalSource::get_right_block(int →
    ↪RF_channel)
{return spoofing_detect_;}
```

src/algorithms/signal_source/libs/spoofing_detection.cc

```
Gnss_Spoofing_Protect::Gnss_Spoofing_Protect(size_t →
    ↪sizeof_stream_item, Concurrent_Queue<pmt::pmt_t>*→
    ↪ queue) : gr::sync_block("spoofing_detection",
    gr::io_signature::make(1, 20, sizeof_stream_item),
    gr::io_signature::make(1, 1, sizeof_stream_item)),
    d_ncopied_items(0),
    d_queue(std::move(queue))
{...}

int Gnss_Spoofing_Protect::work(int noutput_items,
    gr_vector_const_void_star &input_items,
    gr_vector_void_star &output_items)
{...}
```

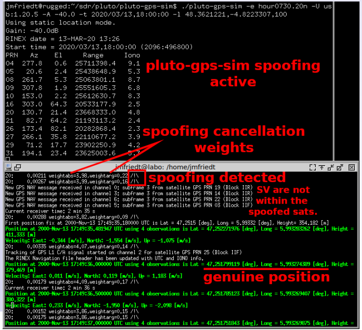
[4] C. Fernandez, *An Open Source Global Navigation Satellite Systems Software-Defined Receiver*, SDRA 2019 at https://www.youtube.com/watch?v=4uHdu-iGsgI

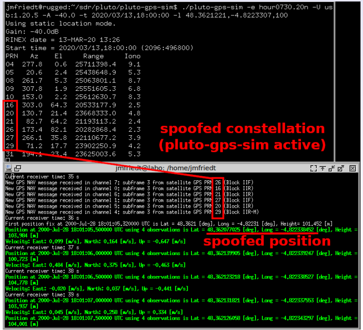
[5] https://github.com/oscimp/gnss-sdr-custom

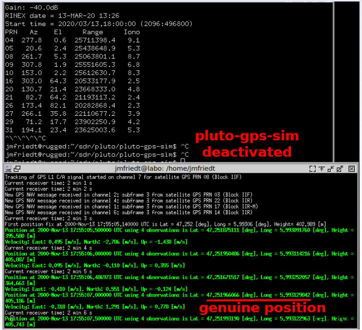
Post-processing spoofing cancellation: decoding

gnss-sdr messages:

1. `Tracking of GPS L1 C/A signal started on channel 3 for satellite GPS PRN 11 (Block IIR)`
   Internal state machine on which channel tracks which SV (little interest)
2. `GPS L1 C/A tracking bit synchronization locked in channel 3 for satellite GPS PRN 11 (Block I`
   SV signal was successfully acquired
3. `New GPS NAV message received in channel 3:  subframe 2 from satellite GPS PRN 11 (Block IIR)`
   SV signal was successfully tracked and decoded
4. `Position at 2000-Dec-22 10:13:35.500000 UTC using 4 observations is Lat = 47.251720167 [deg], Long = 5.993348221 [deg], Height = 326.550 [m]`







Post-processing spoofing cancellation: decoding

Pair of antennas exposed to clear-sky views are subject to varying power of spoofing signal.
Non-deterministic result of `gnss-sdr`: 100-runs on the same dataset and statistical result in %

Attenuation (dB)	Constellation	Correct pos. (%)	Wrong pos. (%)	No solution (%)
none	current	100/100/100/96	not relevant	0/0/0/4
35	current	0/90/100/99	57/0/0/0	43/10/0/1
40	current	0/93/100/99	96/0/0/0	4/7/0/1
45	current	0/2/100/100	61/1/0/0	39/97/0/0
50	current	0/3/100/99	31/7/0/0	69/90/0/1
55	current	52/23/100/0	0/0/0/0	48/77/0/100
60	current	88/64/100/13	0/0/0/0	12/36/0/87
40	-6 h	7/100/100/100	44/0/0/0	49/0/0/0
50	-6 h	6/4/100/32	90/96/0/0	4/0/0/68

raw collected data[6]/cleaned using least-square method[7]/cleaned using FFT ratio method/cleaned using [8]

[6] one of the two antennas

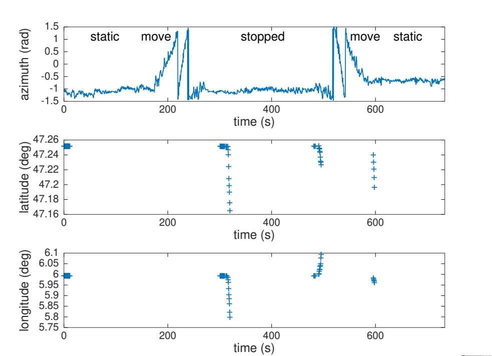
[7] see later when discussing jamming cancellation

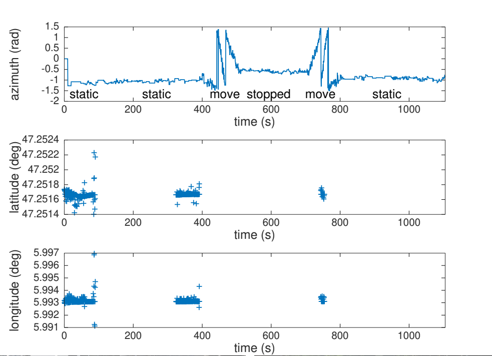
[8] S. Daneshmand & al., *A low-complexity GPS anti-spoofing method using a multi-antenna array*, Proc. 25th International Tech. Meeting of the Satellite Division of The Inst. of Nav. (ION GNSS 2012), pp.1233–1243 (2012) introduces $\int s1(t) \cdot s_2^*(t)dt$ implemented (VOLK) as `volk_32fc_x2_multiply_conjugate_32fc(out,(const gr_complex*)input_items[0],(const gr_complex*)input_items[1],SIZE); sum={0.,0.}; for (ch=0;ch<SIZE;ch++) sum+=out[ch];`

PUBLIC MONEY
PUBLIC CODE

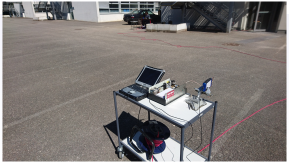
Real time spoofing cancellation

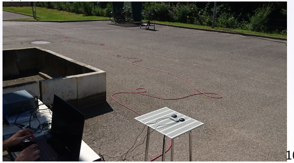
Challenge to keep `gnss-sdr` locked to the constellation as receiver is static wrt a moving spoofing source





If no anti-spoofing filter is applied, the receiver is constantly fooled into the erroneous 48.36°N, 4.822°W (western France) location.

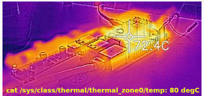




Buildroot for embedded GNU Radio/gnss-sdr

Porting GNU Radio and gnss-sdr to Buildroot:

- gnss-sdr (GNU Radio) uses intensively VOLK (SIMD instruction sets) to improve performance
- A general purpose binary distribution (Raspbian on Raspberry Pi) will not provide efficiency (supports the lowest grade CPU) even with 64 bit kernel (RPi3 v.s RPi4)
- Buildroot for custom, optimized software using NEON instructions of Raspberry Pi 4 CPU
- https://github.com/oscimp/PlutoSDR/tree/master/doc



Buildroot, powersave	Buildroot, performance	Raspbian, ondemand
volk_64u_popcntpuppet_64uu generic completed in 7103.62 ms neon completed in 4038.24 ms Best aligned arch: neon Best unaligned arch: neon	**volk_64u_popcntpuppet_64u** generic completed in 3089.73 ms neon completed in 1897.77 ms Best aligned arch: neon Best unaligned arch: neon	**volk_64u_popcntpuppet_64u** no architectures to test
volk_64u_popcntpuppet_64u generic completed in 7154.26 ms neon completed in 4106.08 ms Best aligned arch: neon Best unaligned arch: neon	**volk_64u_popcntpuppet_64u** redgeneric completed in 3157.41 ms neon completed in 2081.84 ms Best aligned arch: neon Best unaligned arch: neon	**volk_64u_popcntpuppet_64u** no architectures to test
volk_16ic_deinterleave_real_8i generic completed in 1745.19 ms neon completed in 254.155 ms Best aligned arch: neon Best unaligned arch: neon	**volk_16ic_deinterleave_real_8i** generic completed in 697.845 ms neon completed in 105.462 ms Best aligned arch: neon Best unaligned arch: neon	**volk_16ic_deinterleave_real_8i** generic completed in 420.678ms u_orc completed in 391.035ms Best aligned arch: u_orc Best unaligned arch: u_orc
volk_16ic_s32f_deinterleave_32f_x2 generic completed in 2258.27 ms neon completed in 1274.83 ms Best aligned arch: neon Best unaligned arch: neon	**volk_16ic_s32f_deinterleave_32f_x2** generic completed in 2185.24 ms neon completed in 728.173 ms Best aligned arch: neon Best unaligned arch: neon	**volk_16ic_s32f_deinterleave_32f_x2** generic completed in 2211.99ms u_orc completed in 4766.13ms Best aligned arch: generic Best unaligned arch: generic
volk_16i_s32f_convert_32f generic completed in 2181 ms neon completed in 697.446 ms a_generic completed in 2181.02 ms Best aligned arch: neon Best unaligned arch: neon	**volk_16i_s32f_convert_32f** generic completed in 870.3 ms neon completed in 310.137 ms a_generic completed in 870.304 ms Best aligned arch: neon Best unaligned arch: neon	**volk_16i_s32f_convert_32f** generic completed in 749.928ms a_generic completed in 750.233ms Best aligned arch: generic Best unaligned arch: generic
volk_16i_convert_8i generic completed in 1745.56 ms neon completed in 134.038 ms a_generic completed in 1745.59 ms Best aligned arch: neon	**volk_16i_convert_8i** generic completed in 696.289 ms neon completed in 75.7975 ms a_generic completed in 696.28 ms Best aligned arch: neon	**volk_16i_convert_8i** generic completed in 457.922ms a_generic completed in 458.445ms Best aligned arch: generic Best unaligned arch: generic
volk_32f_cos_32f generic_fast completed in 51036.2 ms generic completed in 13673.1 ms Best aligned arch: generic Best unaligned arch: generic	**volk_32f_cos_32f** generic_fast completed in 19325.9 ms generic completed in 4678.62 ms Best aligned arch: generic Best unaligned arch: generic	**volk_32f_cos_32f** generic_fast completed in 22240.9ms generic completed in 5470.72ms Best aligned arch: generic Best unaligned arch: generic

Jamming cancellation

Challenge of jamming: no known structure of the interfering signal (cannot rely on squaring to identify phase and magnitude of weight)

- obvious optimization problem: find the jammer $n(t)$ on antenna 1 signal $s1(t)$ and subtract its contribution from antenna 2 signal $s(2)$ with weight α

$$cleaned = s1 - \alpha \times s2$$

- problem of identifying α: least-square (LS) optimization problem with solution

$$\alpha = pinv(s1) \cdot s2 \text{ where } pinv(X) = (X^t \cdot X)^{-1} \cdot X^t$$

- again demonstrate with `gnss-sdr` decoding (post-processing and real time)
- weight identification α using LS v.s inverse-filtering (IF) $iFFT(FFT(s1)/FFT(s2))[0]$:

With jamming

```
IF 0.1809+i*0.3856        LS (0.2127,0.4668)
LS (0.2191,0.4710)        IF 0.1781+i*0.3618
LS (0.2196,0.4684)        LS (0.2121,0.4662)
IF 0.1779+i*0.3678        LS (0.2129,0.4649)
LS (0.2125,0.4629)        LS (0.2138,0.4657)
```

No jamming

```
IF 0.0606+i*-0.0125       LS (0.00073,0.00029)
LS (0.00027,0.00011)      LS (0.00082,0.00033)
LS (0.00037,0.00014)      LS (0.00092,0.00036)
LS (0.00046,0.00018)      IF 0.06576+i*0.01101
IF 0.06623+i*0.00309      LS (0.00111,0.00044)
LS (0.00064,0.00026)
```

Threshold on $|\alpha|$ is indicator of jamming (beyond loss of service)

Jamming cancellation: post-processing

Pair of antennas exposed to clear-sky views are subject to varying power of jamming signal (varying distance jammer [9]-receiver).

Non-deterministic result of `gnss-sdr`: 100-runs on the same dataset and statistical result in %

Distance	Correct pos.(%)	No sol.(%)
no jamming	100	0
10m00	100/100	0/0
9m00	100/100	0/0
8m00	100/100	0/0
7m50	0/94	100/6
6m50	0/49	100/51
6m00	0/79	100/21
5m50	0/0	100/100
5m00	0/0	100/100
4m50	0/0	100/100

no cancellation/cancellation using least square solution

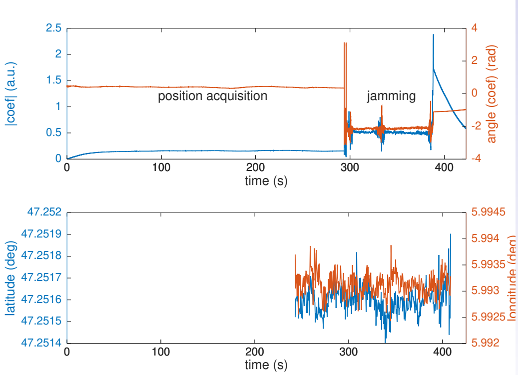
[9] frequency swept VCO jammer bought at `https://www.amazon.fr/IrahdBowen-Bloqueur-Bouclier-Brouilleur-Disjoncteur/dp/B07KSC5LLD` whose antenna was removed and replaced with an SMA connector

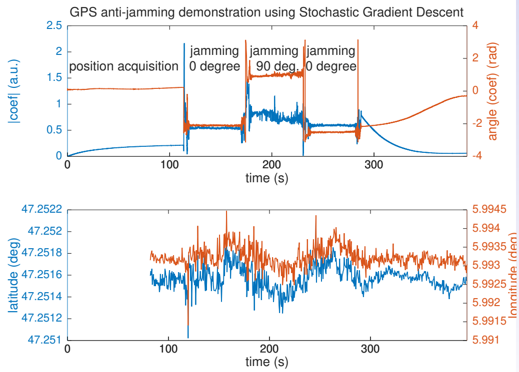
Jamming cancellation: real time

Pseudo invese too slow and computationnally intensive for real time processing of datastream

Stochastic Gradient Descent [10]: iterative solution [11] to finding α by

$\nabla\left(\frac{1}{2}||y-A\alpha||_2^2\right)=\nabla\left(\frac{1}{2}\sum(y_i-a_i^t\alpha)^2\right)=-A^t(y-A\alpha)$: $\alpha^{(m+1)}=\alpha^{(m)}-\eta A^t(y-A\alpha^{(m)})$





[10] J.-M. Friedt & al., *Passive radar for measuring passive sensors: direct signal interference suppression on FPGA using orthogonal matching pursuit and stochastic gradient descent*, SPIE Optical Metrology 2019 – Multimodal Sensing and Artificial Intelligence: Technologies and Applications (Munich, Germany)

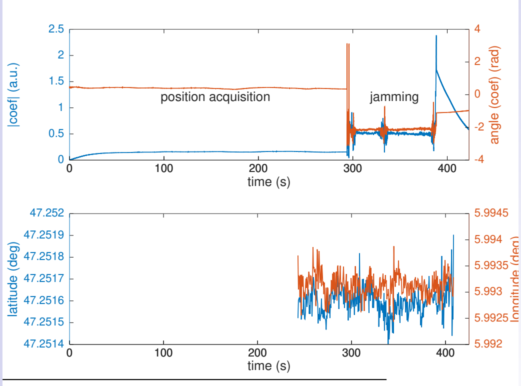
[11] `https://en.wikipedia.org/wiki/Stochastic_gradient_descent`

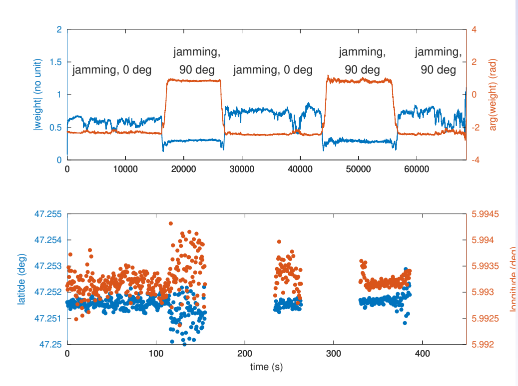
Jamming cancellation: real time

Pseudo invese too slow and computationnally intensive for real time processing of datastream

Stochastic Gradient Descent [10]: iterative solution [11] to finding α by

$$\nabla\left(\frac{1}{2}||y-A\alpha||_2^2\right)=\nabla\left(\frac{1}{2}\sum(y_i-a_i^t\alpha)^2\right)=-A^t(y-A\alpha):\ \alpha^{(m+1)}=\alpha^{(m)}-\eta A^t(y-A\alpha^{(m)})$$





[10] J.-M. Friedt *& al.*, *Passive radar for measuring passive sensors: direct signal interference suppression on FPGA using orthogonal matching pursuit and stochastic gradient descent*, SPIE Optical Metrology 2019 – Multimodal Sensing and Artificial Intelligence: Technologies and Applications (Munich, Germany)

[11] `https://en.wikipedia.org/wiki/Stochastic_gradient_descent`

Conclusion and perspectives

- Demonstrated real time GPS spoofing detection, cancellation and jamming cancellation running as custom processing blocks with `gnss-sdr` ...
- ... using computationally efficient techniques (FFT(squared signal), SGD) ...
- ... running on Raspberry Pi 3 or 4 single board computers with Buildroot software.
- Additional workload does not prevent real time (GPS L1) decoding
- Software available at `https://github.com/oscimp/gnss-sdr`

Perspectives: extend to L5/E5

- wider bandwidth (10-times longer PRN code) ⇒ faster sampling & processing
- requires adapting codeless analysis to BOC modulation [12] [13]
- challenge of moving jammer/spoofer: convince `gnss-sdr` to remember the tracked constellation even when a few signals are lost ?

[12] D. Borio *& al.*, *Codeless processing of binary offset carrier modulated signals*, IET Radar, Sonar & Navigation, **7**(2), pp143–152 (2013)

[13] C. O'Driscoll & J.T. Curran, *Codeless code tracking of BOC signals*, Proc. 29th Int. Technical Meeting of