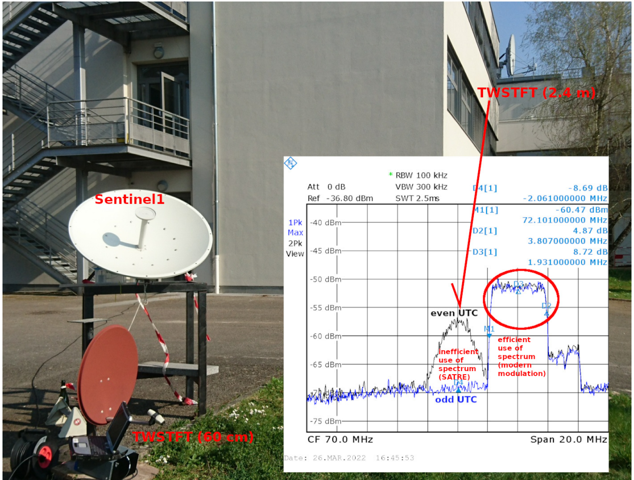
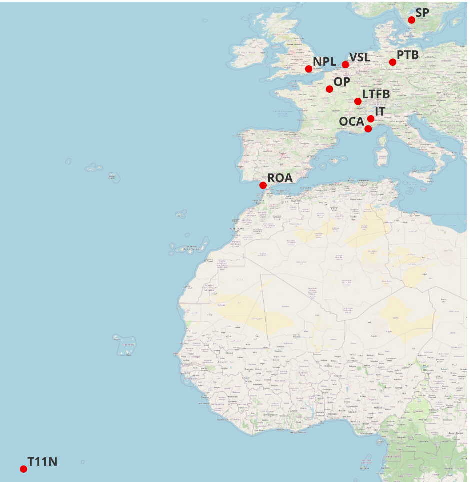
Signaux et méthodes de distributions de temps (historique et actuel), sol et espace

J.-M. Friedt (FEMTO-ST, Time & Frequency, Besançon, France)

references at `http://jmfriedt.free.fr` for reproducible and accessible experiments

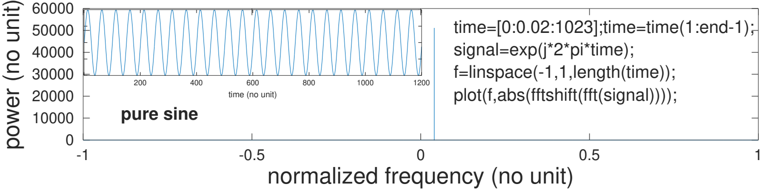




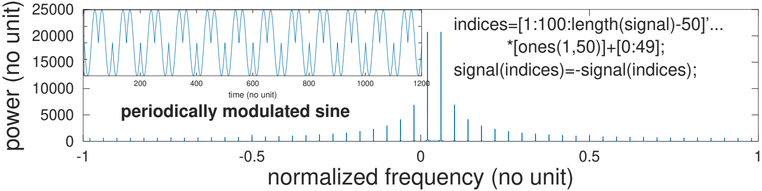
Spectrum spreading numerical experiments

Carrier frequency and bandwidth are two unrelated quantities which can be tuned independently for **matching each sensor spectral characteristics**

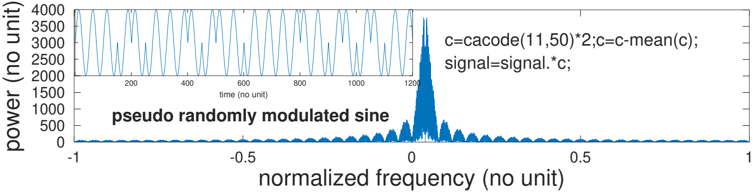
Binary Phase shift keying: $\varphi \in [0;\pi]$ for spectrum spreading



```
time=[0:0.02:1023];time=time(1:end-1);
signal=exp(j*2*pi*time);
f=linspace(-1,1,length(time));
plot(f,abs(fftshift(fft(signal))));
```

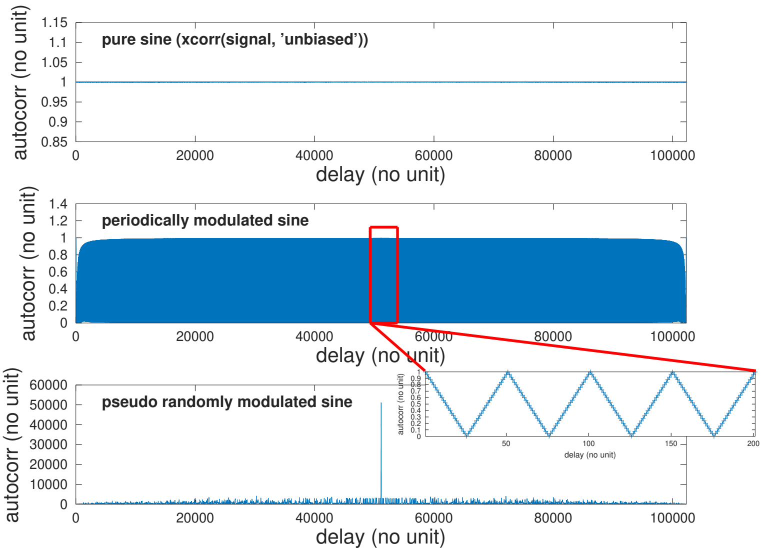


```
indices=[1:100:length(signal)-50]'...
        *[ones(1,50)]+[0:49];
signal(indices)=-signal(indices);
```



```
c=cacode(11,50)*2;c=c-mean(c);
signal=signal.*c;
```

Spectrum spreading numerical experiments

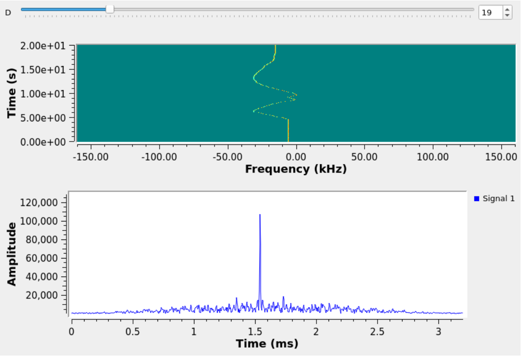


Carrier frequency and bandwidth are two unrelated quantities which can be tuned independently for **matching each sensor spectral characteristics**

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:

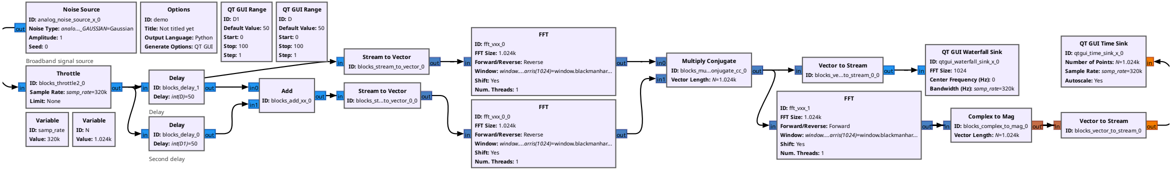
$$FT(conv(s,r))=FT(s)\cdot FT(r)$$

- Correlation:

$$corr(s,r)(\tau)=\int s(t)r^*(t+\tau)dt$$

- Convolution → correlation: time reversal
- since $\exp(j\omega t)^*=\exp(-j\omega t)$, we conclude

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



Noise Source
ID: analog_noise_source_x_0
Noise Type: *analo..._GAUSSIAN*=Gaussian
Amplitude: 1
Seed: 0

Broadband signal source

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

QT GUI Range
ID: D1
Default Value: 50
Start: 0
Stop: 100
Step: 1

QT GUI Range
ID: D
Default Value: 50
Start: 0
Stop: 100
Step: 1

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

Variable
ID: samp_rate
Value: 320k

Variable
ID: N
Value: 1.024k

Delay
ID: blocks_delay_1
Delay: *int(D)*=50

Delay

Delay
ID: blocks_delay_0
Delay: *int(D1)*=50

Second delay

Add
ID: blocks_add_xx_0

Stream to Vector
ID: blocks_stream_to_vector_0

Stream to Vector
ID: blocks_st...to_vector_0_0

FFT
ID: fft_vxx_0
FFT Size: 1.024k
Forward/Reverse: Reverse
Window: *window....arris(1024)*=window.blackmanhar...
Shift: Yes
Num. Threads: 1

FFT
ID: fft_vxx_0_0
FFT Size: 1.024k
Forward/Reverse: Reverse
Window: *window....arris(1024)*=window.blackmanhar...
Shift: Yes
Num. Threads: 1

Multiply Conjugate
ID: blocks_mu...onjugate_cc_0
Vector Length: *N*=1.024k

Vector to Stream
ID: blocks_ve...to_stream_0_0

FFT
ID: fft_vxx_1
FFT Size: 1.024k
Forward/Reverse: Forward
Window: *window....arris(1024)*=window.blackmanhar...
Shift: Yes
Num. Threads: 1

QT GUI Waterfall Sink
ID: qtgui_waterfall_sink_x_0
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): *samp_rate*=320k

Complex to Mag
ID: blocks_complex_to_mag_0
Vector Length: *N*=1.024k

QT GUI Time Sink
ID: qtgui_time_sink_x_0
Number of Points: *N*=1.024k
Sample Rate: *samp_rate*=320k
Autoscale: Yes

Vector to Stream
ID: blocks_vector_to_stream_0

Code orthogonality

- The **correlation** is a linear function:
  $xcorr(ax+by,c) = \int (ax(t)+by(t))\cdot c^*(t+\tau)dt = a\int x(t)c^*(t+\tau)dt + b\int y(t)c^*(t+\tau)dt$
  $a\cdot xcorr(x,c) + b\cdot xcorr(y,c)$
- Correlation will propagate a phase (digital modulation on CDMA: $\varphi \in \{0,\pi\}$ in BPSK

$$xcorr(x\cdot \exp(j\varphi),c) = \exp(j\varphi)\cdot xcorr(x,c)$$

- Correlation can generate an orthogonal basis if selecting the codes wisely

$$xcorr(c_i,c_j) = \delta_{i,j} = \left\{ \begin{array}{l} \neq 0 \text{ if } i == j \\ = 0 \text{ if } i \neq j \end{array} \right.$$

$\Rightarrow$ ability to detect simultaneous communication by multiple speakers using different **known** codes

Spectrum spreading numerical experiments

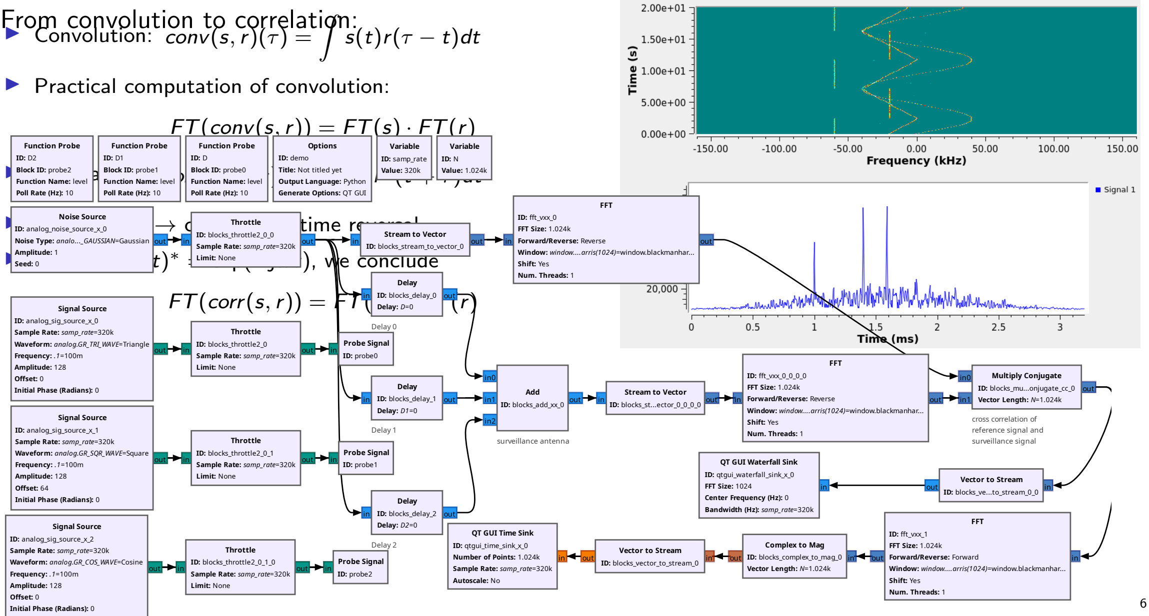
From convolution to correlation:

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

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

- $\to$ c time re
- $t)^*$), we conclude

$$FT(corr(s,r)) = FT \quad (r)$$



Function Probe
ID: D2
Block ID: probe2
Function Name: level
Poll Rate (Hz): 10

Function Probe
ID: D1
Block ID: probe1
Function Name: level
Poll Rate (Hz): 10

Function Probe
ID: D
Block ID: probe0
Function Name: level
Poll Rate (Hz): 10

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

Variable
ID: samp_rate
Value: 320k

Variable
ID: N
Value: 1.024k

Noise Source
ID: analog_noise_source_x_0
Noise Type: *analo..._GAUSSIAN*=Gaussian
Amplitude: 1
Seed: 0

Throttle
ID: blocks_throttle2_0_0
Sample Rate: *samp_rate*=320k
Limit: None

Stream to Vector
ID: blocks_stream_to_vector_0

FFT
ID: fft_vxx_0
FFT Size: 1.024k
Forward/Reverse: Reverse
Window: *window....arris(1024)*=window.blackmanhar...
Shift: Yes
Num. Threads: 1

Delay
ID: blocks_delay_0
Delay: *D*=0

Delay 0

Signal Source
ID: analog_sig_source_x_0
Sample Rate: *samp_rate*=320k
Waveform: *analog.GR_TRI_WAVE*=Triangle
Frequency: *.1*=100m
Amplitude: 128
Offset: 0
Initial Phase (Radians): 0

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

Probe Signal
ID: probe0

Delay
ID: blocks_delay_1
Delay: *D1*=0

Delay 1

Add
ID: blocks_add_xx_0

surveillance antenna

Stream to Vector
ID: blocks_st...ector_0_0_0_0

FFT
ID: fft_vxx_0_0_0_0
FFT Size: 1.024k
Forward/Reverse: Reverse
Window: *window....arris(1024)*=window.blackmanhar...
Shift: Yes
Num. Threads: 1

Multiply Conjugate
ID: blocks_mu...onjugate_cc_0
Vector Length: *N*=1.024k

cross correlation of
reference signal and
surveillance signal

Signal Source
ID: analog_sig_source_x_1
Sample Rate: *samp_rate*=320k
Waveform: *analog.GR_SQR_WAVE*=Square
Frequency: *.1*=100m
Amplitude: 128
Offset: 64
Initial Phase (Radians): 0

Throttle
ID: blocks_throttle2_0_1
Sample Rate: *samp_rate*=320k
Limit: None

Probe Signal
ID: probe1

QT GUI Waterfall Sink
ID: qtgui_waterfall_sink_x_0
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): *samp_rate*=320k

Vector to Stream
ID: blocks_ve...to_stream_0_0

Delay
ID: blocks_delay_2
Delay: *D2*=0

Delay 2

Signal Source
ID: analog_sig_source_x_2
Sample Rate: *samp_rate*=320k
Waveform: *analog.GR_COS_WAVE*=Cosine
Frequency: *.1*=100m
Amplitude: 128
Offset: 0
Initial Phase (Radians): 0

Throttle
ID: blocks_throttle2_0_1_0
Sample Rate: *samp_rate*=320k
Limit: None

Probe Signal
ID: probe2

QT GUI Time Sink
ID: qtgui_time_sink_x_0
Number of Points: 1.024k
Sample Rate: *samp_rate*=320k
Autoscale: No

Vector to Stream
ID: blocks_vector_to_stream_0

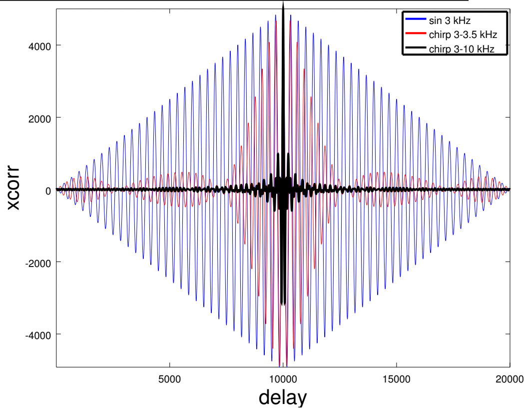
Complex to Mag
ID: blocks_complex_to_mag_0
Vector Length: *N*=1.024k

FFT
ID: fft_vxx_1
FFT Size: 1.024k
Forward/Reverse: Forward
Window: *window....arris(1024)*=window.blackmanhar...
Shift: Yes
Num. Threads: 1

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 $1/B$

$\boxed{\textbf{pulse compression ratio (PCR)} = B \cdot T}$ and for digital systems, $\mathbf{B \cdot T = N}$ **code length**

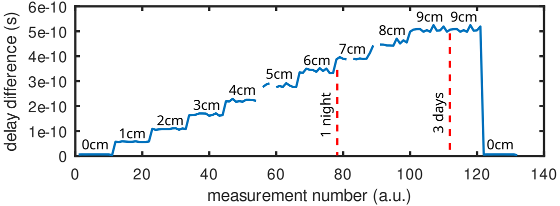


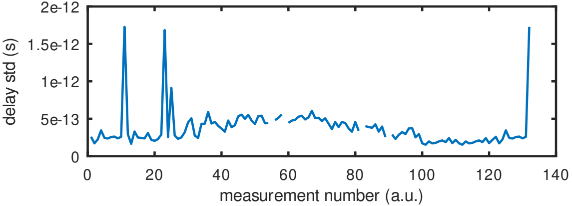
```
time=[0:1e-6:1e-2]; %samp. rate=1 us

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-');
```

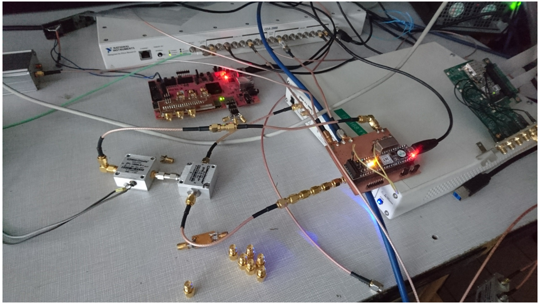
Remember: PRN code is designed for **timing signals** with better than one "chip" resolution. **Pulse Repetition Interval** (PRI): maximum unambiguous delay measurement

Correlation parabolic peak fitting for superresoution

5 MSamples/s SDR measurement [1] or 200 ns sampling period T_s: $5 \cdot 10^{-13} = T_s/(4 \cdot 10^5)$







Sub-ps measurement resolution (1 cm=50 ps @ 20 cm/ns)

While RADAR range resolution states $\Delta R \geq \frac{c}{2B}$, correlation peak fitting leads to **time resolution** $\ll 1/B$

---

[1] J.-M. Friedt & G. Goavec-Merou, *Time of flight measurement with sub-sampling period resolution using Software Defined Radio*, Proc. GNU Radio Conf. (2023) à `pubs.gnuradio.org/index.php/grcon/article/view/142`

PRN spectrum spreading [6] for time transfer

Used in many fields of wireless radiofrequency communication

- VLF (DCF77) with phase modulation over AM modulation [2] reaching 50 μs after 5 min integration ($2 \cdot 10^{-7}$) with 646 Hz bandwidth ($1/645 = 1.5$ ms)
- GNSS (GPS L1 C/A Gold codes)
- "noise RADAR" [3] [4]
- over optical fibers [5]
- Two Way Time Transfer to **compensate for time of flight**: Two Way Satellite Time and Frequency Transfert (TWSTFT) using GEO satellite, Wi-Wi for ground based, White Rabbit for computer networks

Light travels at 300 m/μs or 30 cm/ns or 3000 km/10 ms

---

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

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

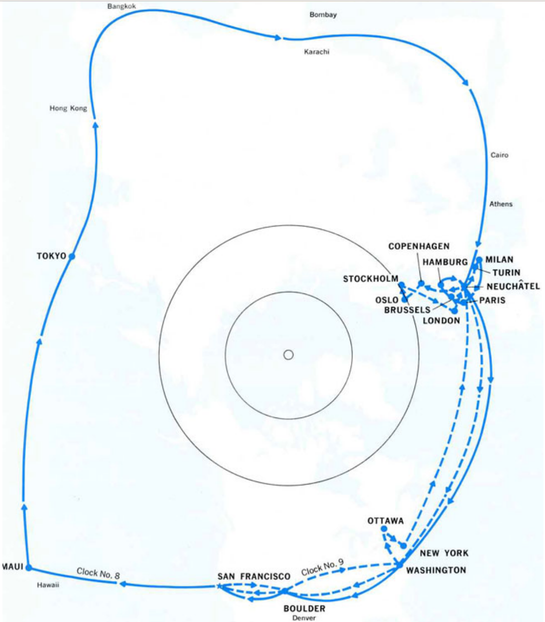
[4] B.M. Horton, *Noise-Modulated Distance Measuring Systems*, Proc. IRE **47** 821–828 (1959)

[5] O. Lopez *& al.*, *Simultaneous remote transfer of accurate timing and optical frequency over a public fiber network*, Applied Physics B **110** 3–6 (2013)

[6] `https://www.analog.com/en/resources/technical-articles/introduction-to-spreadspectrum-communications--maxim-integrated.html`

Traveling clock...

- L.N. Bodily *& al.*, "*World-Wide Time Synchronization, 1966*", Hewlett-Packard Journal (Aug. 1966) at `https://www.hpmemoryproject.org/timeline/alan_bagley/hpj_aug66_01.htm` →
- NPL, *Time transfer Information Film* at `https://www.youtube.com/watch?v=SXV4c5eVkE4` ↓



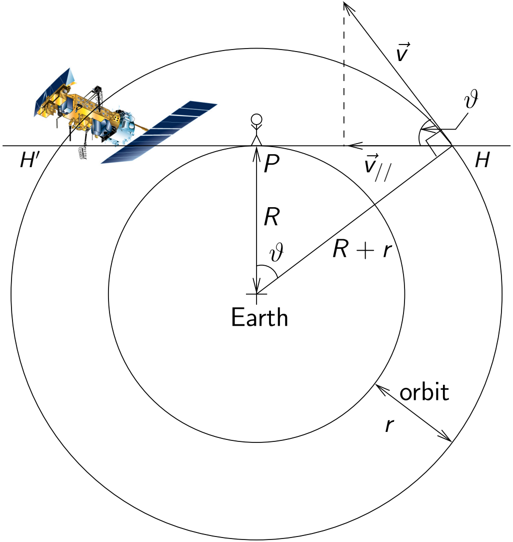
`https://time-nuts.febo.narkive.com/nCHwTARZ/time-transfer-internationally-before-gps`

Spaceborne time and frequency transfer

A satellite at altitude h can illuminate an area of radius $d = R \arccos\left(\frac{R}{r+R}\right)$ with $R \simeq 6400$ km

- GEO: $r = 36000$ km $\Rightarrow d = 9100$ km or $\pm 80^\circ$ latitude
- MEO: $r = 20000$ km $\Rightarrow d = 8500$ km
- LEO: $r = 800$ km $\Rightarrow d \simeq 3000$ km
- ISS: $r \simeq 400$ km $\Rightarrow d \simeq 2200$ km
- Puy de Dôme: $r \simeq 1.5$ km $\Rightarrow d \simeq 140$ km

Before satellite communication: tropospheric scatter. Also **meteor** scatter [a] [b] and moonbounce [c].



---

[a] V.V. Sidorov & al, *Two-way time transfer experiments via meteor burst communication*, Proc. Conf. on Precision Electromagnetic Measurements Digest (1994)

[b] V.A. Korneev & al. *Information protection based on nanosecond synchronization of time scales in meteor burst channel*, Automation and Remote Control **69**(6) 1065–1076 (2008)

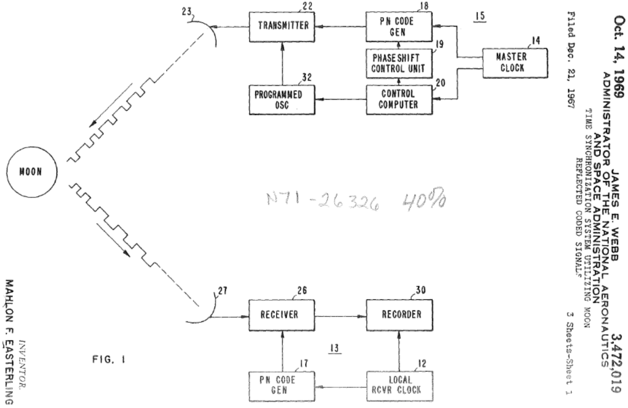
[c] `https://ntrs.nasa.gov/api/citations/19710016850/downloads/19710016850.pdf`

Spaceborne time and frequency transfer

A satellite at altitude h can illuminate an area of radius $d = R \arccos\left(\frac{R}{r+R}\right)$ with $R \simeq 6400$ km

- GEO: $r = 36000$ km $\Rightarrow d = 9100$ km or $\pm 80^\circ$ latitude
- MEO: $r = 20000$ km $\Rightarrow d = 8500$ km
- LEO: $r = 800$ km $\Rightarrow d \simeq 3000$ km
- ISS: $r \simeq 400$ km $\Rightarrow d \simeq 2200$ km
- Puy de Dôme: $r \simeq 1.5$ km $\Rightarrow d \simeq 140$ km

Before satellite communication: tropospheric scatter. Also **meteor** scatter [a] [b] and moonbounce [c].



---

[a] V.V. Sidorov & al, *Two-way time transfer experiments via meteor burst communication*, Proc. Conf. on Precision Electromagnetic Measurements Digest (1994)

[b] V.A. Korneev & al. *Information protection based on nanosecond synchronization of time scales in meteor burst channel*, Automation and Remote Control **69**(6) 1065–1076 (2008)

[c] `https://ntrs.nasa.gov/api/citations/19710016850/downloads/19710016850.pdf` ⟶

Spaceborne time and frequency transfer

A satellite at altitude h can illuminate an area of radius $d = R \arccos\left(\frac{R}{r+R}\right)$ with $R \simeq 6400$ km

- GEO: $r = 36000$ km $\Rightarrow d = 9100$ km or $\pm 80^\circ$ latitude
- MEO: $r = 20000$ km $\Rightarrow d = 8500$ km
- LEO: $r = 800$ km $\Rightarrow d \simeq 3000$ km
- ISS: $r \simeq 400$ km $\Rightarrow d \simeq 2200$ km
- Puy de Dôme: $r \simeq 1.5$ km $\Rightarrow d \simeq 140$ km

but LEO requires many **more satellites** than MEO + high quality ephemeris despite **atmospheric drag**

- **MEO**: GNSS (GPS, GLONASS, Galileo, Beidou)

  > A GPS satellite broadcasts 50 W from 20000 km[7] away: received power is 10 dB **below** thermal noise

- **GEO**: TWSTFT [a] and regional GNSS (India NavIC, EGNOS, WAAS, QZSS); DVB-S ($\lesssim 100$ ns) [b]
- **LEO**[c]: Iridium (STL [d]), Xona, Starlink using signal of opportunity [8]

Iridium GSM signals can be decoded (`gr-iridium`[9]) but STL might be encrypted

---

[a] commercial service by NIST: *Time over Satellite Special Test, SKU: 78500S* at `https://shop.nist.gov/ccrz__ProductDetails?sku=78500S&cclcl=en_US`

[b] Y. Xiang *et al*, *A method of high precision time transfer based on DVB-S*, Joint EFTF/IFCS (2014)

[c] *Is LEO PNT the Next Big Thing?*, ION (Winter 2023)

[d] Satellite Time and Location by Satelles

---

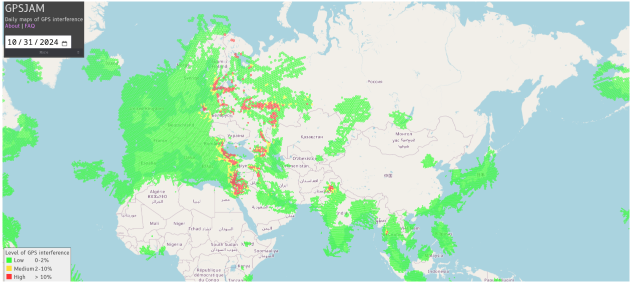
[7] *The economic impact on the UK of a disruption to GNSS* (18 October 2023), `https://www.gov.uk/government/publications/report-the-economic-impact-on-the-uk-of-a-disruption-to-gnss`

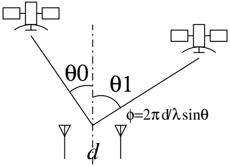
[8] T.E. Humphreys *& al.*, *Signal Structure of the Starlink Ku-Band Downlink*, IEEE Trans. Aerospace and Electronic Systems **59**(5) 6016-6030 (2023)

[9] J.-M Friedt, *À l'écoute des messages transmis par satellite en orbite basse : Iridium*, MISC Hors Série 29 (2024)

GNSS vulnerabilities

- jamming: loss of service, little or no interest (`gpsjam.org`)
- spoofing: injection of unwanted signals in a receiver (CRPA protection when using multiple antennas)
- NTRIP[10] casters for "secure" GNSS
  - NTRIP is internet-based RTCM (Radio Technical Commission for Maritime Services) message transfer
  - a GNSS receiver measures the raw time of flight from GNSS satellites (*pseudo-ranges*)
  - a differential GNSS receiver (rover) compensates for unwanted time delays (ionosphere, troposphere) by subtracting the time of flight from pseudo-ranges from a reference time station (base)
- UBlox Zed-F9P is NTRIP compatible
- freely available network of NTRIP casters: `https://centipede.fr/`





$$\frac{FFT(s_1^2(t))}{FFT(s_2^2(t))} = \frac{A_1^2(t)}{A_2^2(t)} \exp(2j(\varphi_{1,i} - \varphi_{2,i}))$$

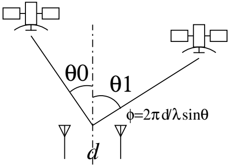
computed at bin $\delta\omega_i$ of FFT

---

[10] J.-M Friedt, *Mise en œuvre d'un serveur NTRIP pour la mesure de position centimétriqe : qu'est-ce que l'altitude ?*, Hackable (à paraître)

GNSS vulnerabilities

- jamming: loss of service, little or no interest (`gpsjam.org`)
- spoofing: injection of unwanted signals in a receiver (CRPA protection when using multiple antennas)
- NTRIP[10] casters for "secure" GNSS
  - NTRIP is internet-based RTCM (Radio Technical Commission for Maritime Services) message transfer
  - a GNSS receiver measures the raw time of flight from GNSS satellites (*pseudo-ranges*)
  - a differential GNSS receiver (rover) compensates for unwanted time delays (ionosphere, troposphere) by subtracting the time of flight from pseudo-ranges from a reference time station (base)
- UBlox Zed-F9P is NTRIP compatible
- freely available network of NTRIP casters: `https://centipede.fr/`



$$\frac{FFT(s_1^2(t))}{FFT(s_2^2(t))} = \frac{A_1^2(t)}{A_2^2(t)} \exp(2j(\varphi_{1,i} - \varphi_{2,i}))$$

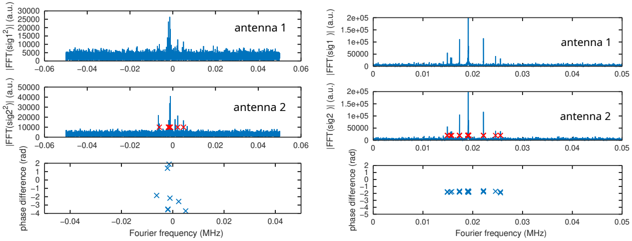
computed at bin $\delta\omega_i$ of FFT

---

[10] J.-M Friedt, *Mise en œuvre d'un serveur NTRIP pour la mesure de position centimétriqe : qu'est-ce que l'altitude ?*, Hackable (à paraître)

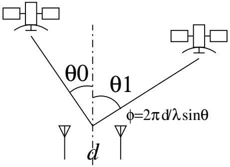
GNSS vulnerabilities

- jamming: loss of service, little or no interest (`gpsjam.org`)
- spoofing: injection of unwanted signals in a receiver (CRPA protection when using multiple antennas)
- NTRIP[10] casters for “secure” GNSS
  - NTRIP is internet-based RTCM (Radio Technical Commission for Maritime Services) message transfer
  - a GNSS receiver measures the raw time of flight from GNSS satellites (*pseudo-ranges*)
  - a differential GNSS receiver (rover) compensates for unwanted time delays (ionosphere, troposphere) by subtracting the time of flight from pseudo-ranges from a reference time station (base)
- UBlox Zed-F9P is NTRIP compatible
- freely available network of NTRIP casters: `https://centipede.fr/`



Genuine

Spoofed



$$\frac{FFT(s_1^2(t))}{FFT(s_2^2(t))} = \frac{A_1^2(t)}{A_2^2(t)}\exp(2j(\varphi_{1,i} - \varphi_{2,i}))$$

computed at bin $\delta\omega_i$ of FFT

`https://github.com/oscimp/gnss-sdr-1pps`

---

[10] J.-M Friedt, *Mise en œuvre d’un serveur NTRIP pour la mesure de position centimétriqe : qu’est-ce que l’altitude ?*, Hackable (à paraître)

NTRIP casters as a source of GNSS security?

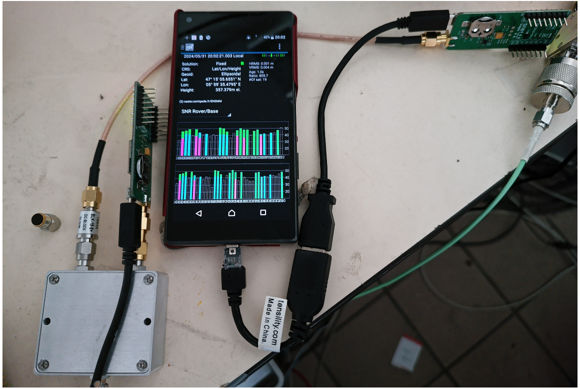
M. Spanghero, P. Papadimitratos, *Testing network-based RTK for GNSS receiver security* (2024) (`https://arxiv.org/pdf/2405.10906`)

"Nevertheless, neither the GNSS receiver nor the RTKLib implementation seems to be aware of the spoofed reference station and instead of rejecting meaningless corrections, tries to reach convergence degrading the RTK solution quality which goes from full RTK fix (where the carrier phase information is fully resolved) to Differential GNSS (considering only **double differences on the pseudoranges**). Instead, the receiver should reject any correction that does not improve the accuracy achievable in stand-alone positioning mode. Additional investiga- tions in these regards are ongoing."

⇒ inconsistent phase does not prevent PVT convergence [a]

Failed to demonstrate spoofing of UBlox Zed-F9P with `https://github.com/Mictronics/pluto-gps-sim`
Expected behaviour:

- RTK solution with **genuine** constellation (phase + code of base and rover)
- Differential GPS with **spoofing** signal (code only)



42% 20:02

rtk

2024/05/31 20:02:21.003 Local

Solution: Fixed
CRS: Lat/Lon/Height
Geoid: Ellipsoidal
Lat: 47° 15' 05.6551" N
Lon: 05° 59' 35.4795" E
Height: 357.379m el.

HRMS: 0.001 m
VRMS: 0.004 m
Age: 1.0s
Ratio: 805.7
#Of sat: 19

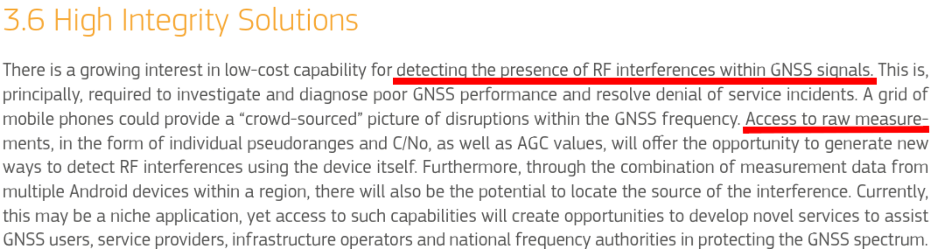
(2) caster:centipede.fr/ENSMM

---

[a] `https://github.com/jancelin/RtkGps/issues/15`

GNSS vulnerabilities: Android

Raw Measurements Task Force, *Using GNSS raw measurements on Android devices* (2017) [11]



3.6 High Integrity Solutions

There is a growing interest in low-cost capability for detecting the presence of RF interferences within GNSS signals. This is, principally, required to investigate and diagnose poor GNSS performance and resolve denial of service incidents. A grid of mobile phones could provide a "crowd-sourced" picture of disruptions within the GNSS frequency. Access to raw measurements, in the form of individual pseudoranges and C/No, as well as AGC values, will offer the opportunity to generate new ways to detect RF interferences using the device itself. Furthermore, through the combination of measurement data from multiple Android devices within a region, there will also be the potential to locate the source of the interference. Currently, this may be a niche application, yet access to such capabilities will create opportunities to develop novel services to assist GNSS users, service providers, infrastructure operators and national frequency authorities in protecting the GNSS spectrum.

Android (since version 7) publishes raw pseudo-range measurements. Only **raw data (pseudoranges + phase)** can be used to assess spoofing, the processed PVT solution is too late.

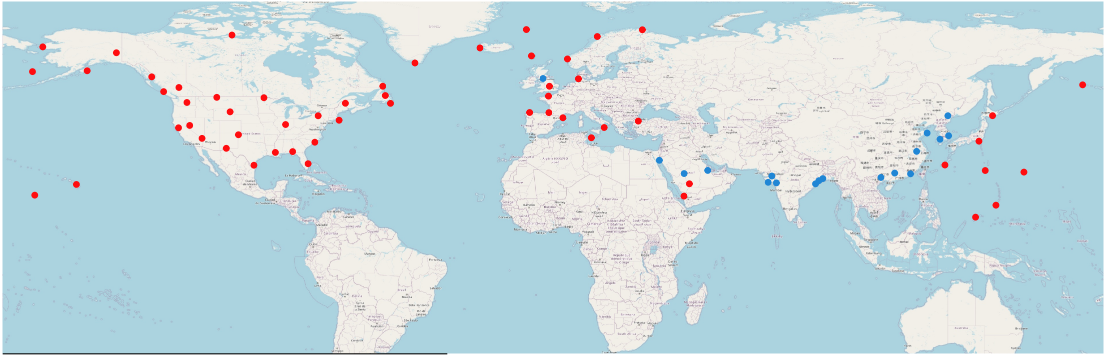
---

[11] `https://galileognss.eu/wp-content/uploads/2018/05/Using-GNSS-Raw-Measurements-on-Android-devices.pdf`

Ground based time and frequency transfer: wireless

Long range communication ⇒ very low frequency, using the ionosphere as a waveguide

- dedicated signals: DCF77 (Mainflingen, DE: 77.5 kHz, AM and PM modulation), MSF (Anthorn, UK: 60 kHz), ALS162 (Allouis, FR: 162 kHz), JJY (JP: 40 & 60 kHw), WWVB (USA: 60 kHz), WWVH (Hawaii: 2.5, 5, 10 et 15 MHz), Polskie Radio (225 kHz) also on DAB+
- Germany EFR[12]: Mainflingen (DE) 129.1 kHz, Burg (DE) 139.0 kHz, Lakihegy (HU) 135.6 kHz.
- positioning: OMEGA (1971–1997), (e)LORAN (100 kHz) ↓ (Saudi Arabia, India, China, Corea), R-Mode [13]: 300 kHz



---

[12] B. Sbick, K. Katzmann, E. Staliuniene, D. Piester, A. Bauch, *Monitoring of Time Signals Broadcast by EFR Long-Wave Transmitters at PTB* (2017)

[13] `https://interreg-baltic.eu/project/r-mode-baltic/`: Germany, Poland, Sweden, Denmark

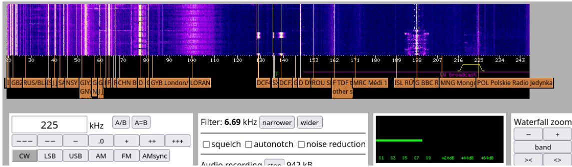
VLF timing analysis using Software Defined Radio (SDR)

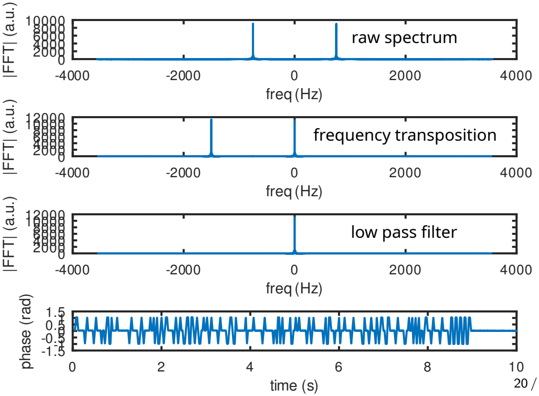
Signal reception: websdr in Twente (NL) at `http://websdr.ewi.utwente.nl:8901/`

```
pkg load signal
d=dir('*wav'); x=audioinfo(d(1).name);
fs=x.SampleRate;       # sampling frequency
x=audioread(d(1).name); L=length(x);
f=linspace(-fs/2,fs/2-fs/L,L);  # freq axis
t=[0:L-1]/fs;           # discrete time (s)
fIF=751.41;             # IF frequency (Hz)
nco=exp(j*2*pi*fIF*t);     # Numerical Osc.
x=x.*nco';        # frequency transposition
N=256;           # low pass filter taps vvv
b=firls(N,[0 300 500 fs/2]*2/fs,[1 1 0 0]);
x=filter(b,1,x);          # low pass filter
x=unwrap(angle(x));       # unwrapped phase
[a,b]=polyfit([0:length(x)-1],x,1);
x=(x-b.yf)(N:end);
```

[a]

ALS162 [b] ⟶





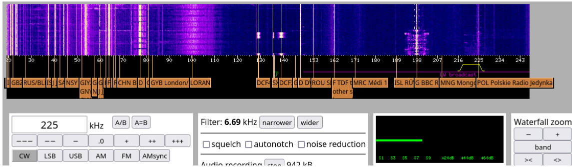
---

[a] 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) for DCF77

[b] H. Maier, *ALS162 Time Signal SDR Receiver for GNU Radio*, Proc. GNU Radio Conf. (2023)

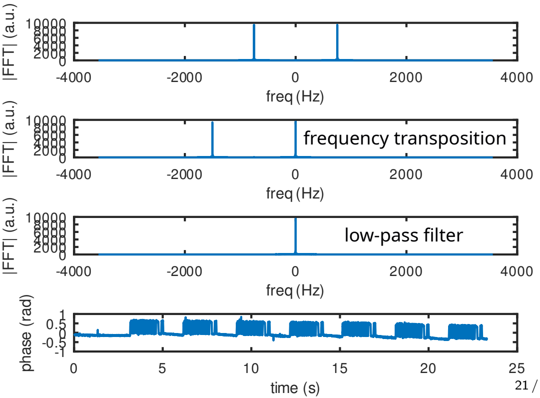
VLF timing analysis using Software Defined Radio (SDR)

Signal reception: websdr in Twente (NL) at `http://websdr.ewi.utwente.nl:8901/`



```
pkg load signal
d=dir(’*wav’); x=audioinfo(d(1).name);
fs=x.SampleRate;      # sampling frequency
x=audioread(d(1).name); L=length(x);
f=linspace(-fs/2,fs/2-fs/L,L); # freq axis
t=[0:L-1]/fs;          # discrete time (s)
fIF=751.41;            # IF frequency (Hz)
nco=exp(j*2*pi*fIF*t);    # Numerical Osc.
x=x.*nco’;       # frequency transposition
N=256;          # low pass filter taps vvv
b=firls(N,[0 300 500 fs/2]*2/fs,[1 1 0 0]);
x=filter(b,1,x);         # low pass filter
x=unwrap(angle(x));      # unwrapped phase
[a,b]=polyfit([0:length(x)-1],x,1);
x=(x-b.yf)(N:end);
```

[a] Polskie Radio Jedynka (225 kHz)⟶

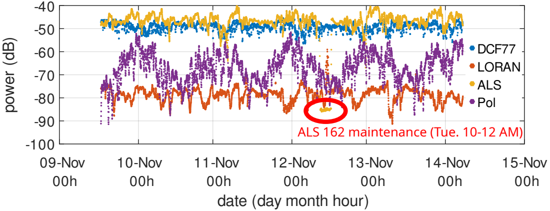


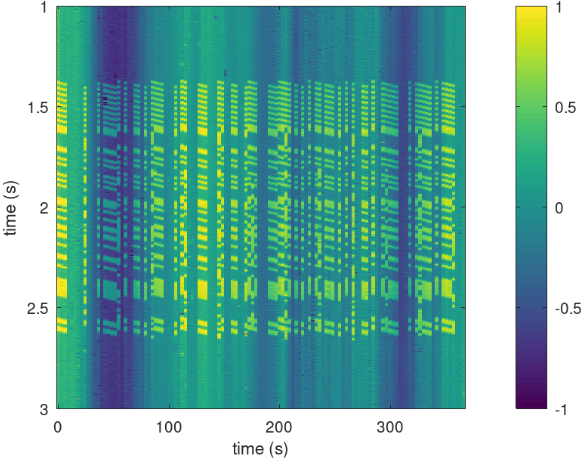
---

[a] 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) for DCF77

VLF timing analysis using Software Defined Radio (SDR)

Polish code (since 2023) includes Reed-Solomon and CRC for message integrity validation





GNU Radio receiver and C(++) decoders available [a][b][c]

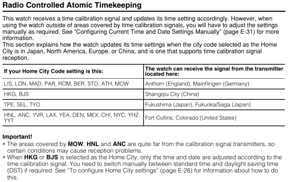
---

[a] https://pa3fwm.nl/signals/poland225kHz/

[b] https://github.com/sp6hfe/e-CzasPL

[c] https://github.com/sp5wwp/e-Czas

VLF timing analysis using Software Defined Radio (SDR)



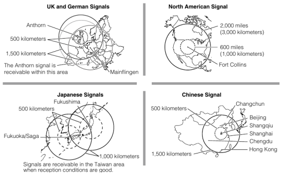
Radio Controlled Atomic Timekeeping

This watch receives a time calibration signal and updates its time setting accordingly. However, when using the watch outside of areas covered by time calibration signals, you will have to adjust the settings manually as required. See "Configuring Current Time and Date Settings Manually" (page E-31) for more information.
This section explains how the watch updates its time settings when the city code selected as the Home City is in Japan, North America, Europe, or China, and is one that supports time calibration signal reception.

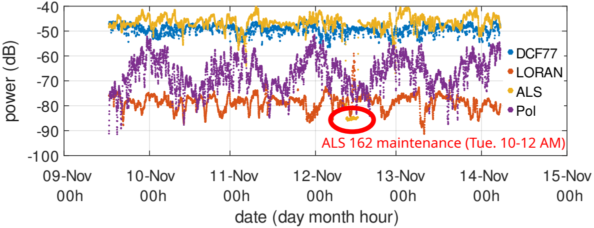
If your Home City Code setting is this:	**The watch can receive the signal from the transmitter located here:**
LIS, LON, MAD, PAR, ROM, BER, STO, ATH, MOW	Anthorn (England), Mainflingen (Germany)
HKG, BJS	Shangqiu City (China)
TPE, SEL, TYO	Fukushima (Japan), Fukuoka/Saga (Japan)
HNL, ANC, YVR, LAX, YEA, DEN, MEX, CHI, NYC, YHZ, YYT	Fort Collins, Colorado (United States)

Important!
- The areas covered by **MOW**, **HNL** and **ANC** are quite far from the calibration signal transmitters, so certain conditions may cause reception problems.
- When **HKG** or **BJS** is selected as the Home City, only the time and date are adjusted according to the time calibration signal. You need to switch manually between standard time and daylight saving time (DST) if required. See "To configure Home City settings" (page E-28) for information about how to do this.

Approximate Reception Ranges



"With Auto Receive, the watch performs the receive operation each day automatically up to six times (up to five times for the Chinese calibration signal) between the **hours of midnight and 5 a.m.** (according to the Timekeeping Mode time). When any receive operation is successful, none of the other receive operations for that day are performed."[14]

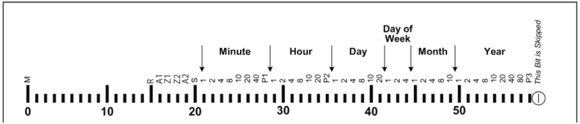


Error in DCF77 communication (Mainflingen-Besançon: 400 km)

DCF77 TIME CODE

DCF77 Time Signal Coding

The DCF77 time code provides a complete date/time string once every minute. Each minute, a pulse-string contains a BCD (Binary Coded Decimal) value for minute, hour, day, day of week, month, and year as well as other control parameters such as leap second and Daylight Saving Time (Summer Time) as shown below.



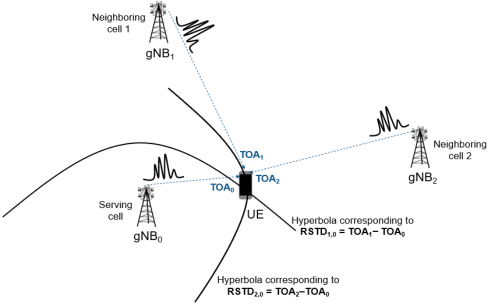
`https://www.cyber-sciences.com/wp-content/uploads/2019/01/TN-103_DCF77.pdf`

---

[14] `www.casio.com/content/dam/casio/global/support/manuals/watches/pdf/32/3258/qw3258_EN.pdf`

Ground based time and frequency transfer: wireless

- **TDMA** requires time synchronization, a core capability of **5G networks** leading to positioning capability, but mostly relying on GNSS



NR Positioning Using PRS at `https://www.mathworks.com/help/5g/ug/nr-prs-positioning.html`

J.-M Friedt, *Programmation USB sous GNU/Linux : application du FX2LP pour un récepteur de radio logicielle dédié aux signaux de navigation par satellite (2/2)*, Hackable (à paraître)

- Wi-Wi for two-way time transfer (experimental setup at NICT, Japan) `https://www.nict.go.jp/en/sts/wi-wi.html`

Signals of opportunity: passive reception of TWSTFT signals

- National Metrology Laboratories share their local atomic clock time and frequency signals through a satellite link over the Atlantic (Telstar11N)
- BIPM[a] collects these information and publishes the error between a weighted average and each clock $UTC(k) - UTC$
- add leap second to try and keep UTC synchronous with astronomical time
- sharing time between A and B in Two Way: $TWSTFT_{a,b} = \frac{1}{2}((RX_a - LO_a) - (RX_b - LO_b))))$
- anyone with a ≥ 60 cm parabola aimed at T11N can receive RX_i[b][c]
- need to compensate for satellite motion (± 75 μs) for time transfer using **known** NMI broadcasting position

Top-left: VSL, top-right: IT, bottom: ROA (Google)

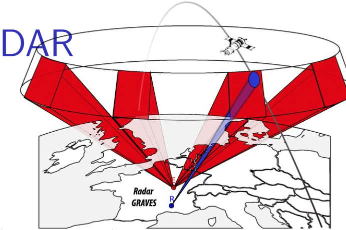
---

[a] `https://webtai.bipm.org/ftp/pub/tai/data/2023/time_transfer/twstft`

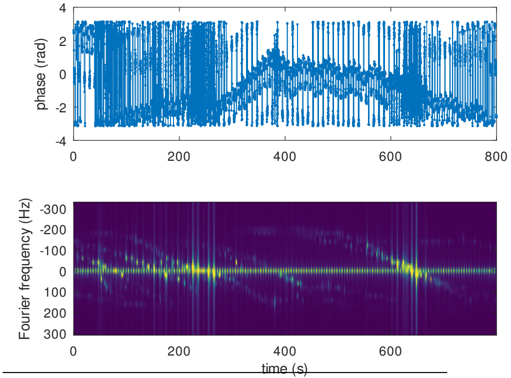
[b] C. Rieck & al., *Utilizing TWSTFT in a Passive Configuration*, Proc. 48th PTTI, 219–234 (2017)

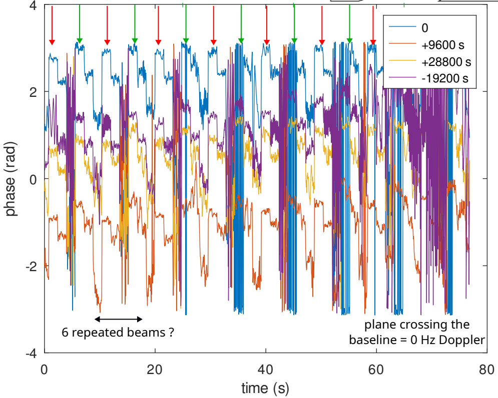
[c] M. Plumaris, T.T. Thai, D. Rovera, D. Dirkx, L. Iess & I. Sesia *Passive TWSTFT for UTC(k) Dissemination*, EFTF (2024)

Signal of opportunity: GRAVES space surveillance RADAR



- GRAVES[15] (Grand Réseau Adapté à la VEille Spatiale)
- CW at 143.05 MHz but electronic beam steering
- emitter located in Pesmes (Haute-Saône), 40 km West of Besançon beaming towards the sky (receiver at Plateau d'Albion)

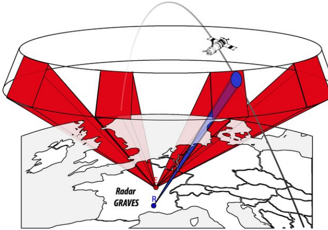




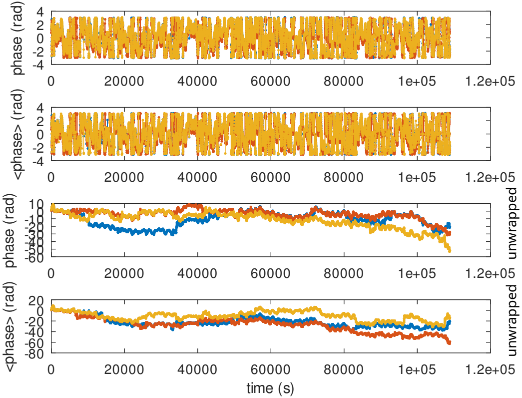
---

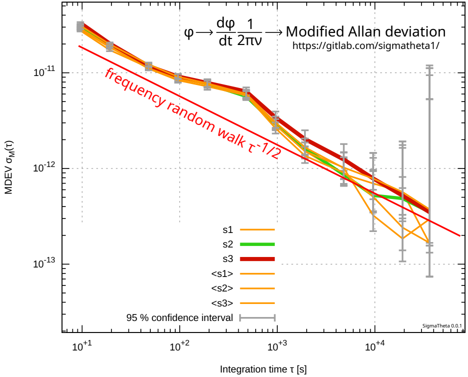
[15] J.-M Friedt, *Distributed coherent SDR systems: GNU Radio rides the White Rabbit*, European GNU Radio Days (2024) at `https://www.youtube.com/watch?v=iyUabco0z4A`

Frequency stability of GRAVES?



- Recording the aliased $200 - 143.05 = 56.95$ MHz … but still a remaining 19.18 Hz despite clocking the X310 with a hydrogen maser compared to the primary reference in Paris Observatory ($\Delta f/f < 10^{-13}$) see [16]
- Electronically steered beam to sweep the sky with the CW
- … need to extract each individual beam and observe phase evolution.

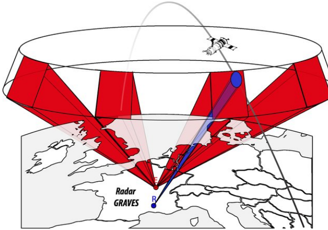




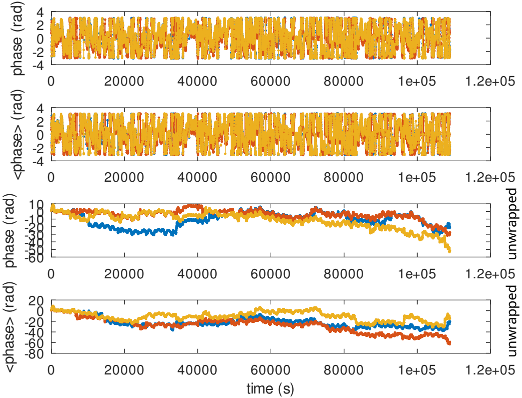
---

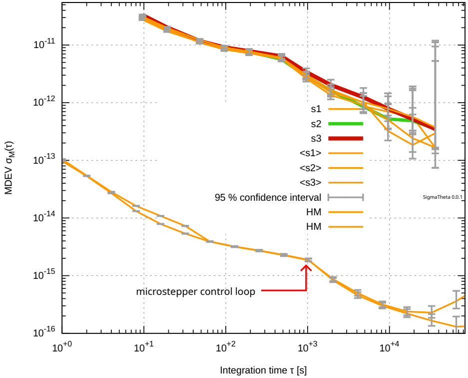
[16] Github issue `https://github.com/EttusResearch/uhd/issues/763`

Frequency stability of GRAVES?



- Recording the aliased $200 - 143.05 = 56.95$ MHz … but still a remaining 19.18 Hz despite clocking the X310 with a hydrogen maser compared to the primary reference in Paris Observatory ($\Delta f/f < 10^{-13}$)
- Electronically steered beam to sweep the sky with the CW [16]
- … need to extract each individual beam and observe phase evolution.

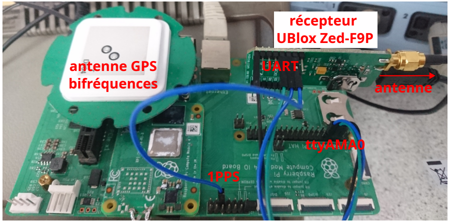




---

[16] A. Jouade & A. Barka, *Massively Parallel Implementation of FETI-2LM Methods for the Simulation of the Sparse Receiving Array Evolution of the GRAVES Radar System for Space Surveillance and Tracking*, IEEE Access (2019)

Ground based time and frequency transfer: wired



- Synchronization over computer networks
  - NTP (software only, $\simeq$ ms)
  - PTP (hardware synchronization, $\simeq \mu$s)
  - White Rabbit/PTP HA (hardware synchronization and syntonization, $\lesssim$ ns)
  - REFIMEVE(-T) optical signal (no digital communication)
- PTP timestamps received packets and leaves the local oscillator running $\neq$ WR disciplines slave oscillator on master and measures time delay with phase detector and time stretching (DDMTD)
- WR possible on non-dedicated board? See [17] and their presentation at 13th White Rabbit Workshop (March 2024)
- Raspberry Pi Compute Module 4 NIC is PTP compatible [18] (drivers sponsored by Meta)

---

[17] `https://www.missinglinkelectronics.com/company/news/mle-upstreamed-an-implementation-for-vcxo-less-white-rabbit-nodes-on-an-amd-zynq-ultrascale-mpsoc-zcu10` (October 21, 2024)

[18] J.-M Friedt, *Synchronisation d'ordinateurs par réseau informatique pour la datation sous GNU Linux : NTP, PTP et GPS sur Raspberry Pi Compute Module 4*, Hackable **51** (Nov/Dec 2023)

European analysis of A-PNT

Analysis of commercial solutions for alternatives to GNSS:
L. Bonenberg *& al.*, *Assessing Alternative Positioning, Navigation and Timing Technologies for Potential Deployment in the EU*, JRC (2023) [a]

- OPNT (WR)
- Seven Solutions (now Orolia, now Safran: WR)
- GMV Aerospace and Defence SAU (GNSS common view, DTM/WR)
- SCPTime (proprietary NTP/STS)
- Satelles Inc (LEO Iridium)
- Locata Corporation Pty Ltd (proprietary terrestrial radio)
- NextNav (proprietary terrestrial radio)

Conclusions: "The diversity in both the type of the assessed A-PNT technologies and the test conditions that were applied are such that a **fair benchmarking of the performances cannot be made in a fair manner.**"
"The main conclusion of the test campaign is that all A-PNT platforms fulfilled the minimum requirements set in the ITT..."
"EU companies have excellent record in time transfer and time generation. The test campaign highlighted the important role of the NMIs across Europe, ..."'

ISSN 1831-9424

European Commission

JRC SCIENCE FOR POLICY REPORT

Assessing Alternative Positioning, Navigation, and Timing Technologies for Potential Deployment in the EU

2023

Joint Research Centre

EUR 31450 EN

---

[a] `publications.jrc.ec.europa.eu/repository/handle/JRC132737`

Conclusion: time and frequency has become an ubiquitous need in daily activities

`https://www.dhs.gov/science-and-technology/pnt-program:`

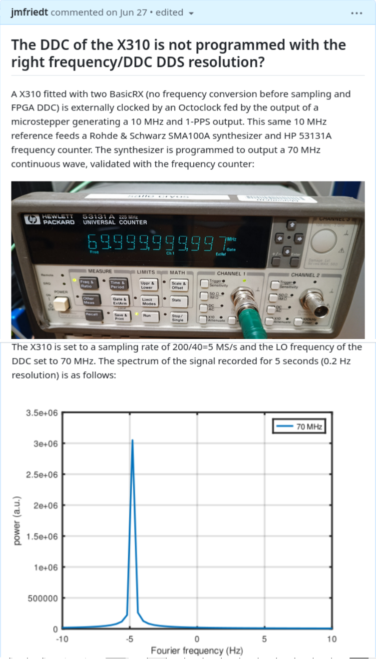
> "Accurate positioning, navigation, and timing (PNT) is **necessary for the functioning of many critical infrastructure** sectors. Precision timing is particularly important and is primarily provided through the Global Positioning System (GPS). However, GPS' space-based signals are low-power and unencrypted, making them susceptible to both intentional and unintentional disruption.
> To address GPS vulnerabilities in critical infrastructure, the Science and Technology Directorate (S&T) Positioning, Navigation, and Timing (PNT) Program has a **multi-pronged approach** of conducting vulnerability and impact assessments, developing mitigations, exploring **complementary timing technologies**, and engaging with industry through outreach events and meetings. Through these sustained efforts, the goal of the program is to increase the resiliency of critical infrastructure to GPS vulnerabilities in the future."

`https://www.gov.uk/government/news/`
`critical-services-to-be-better-protected-from-satellite-data-disruptions-through-new-position-navigation`

> "A resilient PNT system supports a **range of interactions in our daily lives**, from ensuring secure banking to uninterrupted television and radio broadcasts, and more widely stock market operations – ensuring the systems run smoothly, even in adverse conditions such as severe space weather events, which could disrupt PNT provided by Global Navigation Satellite Systems."

Many alternatives to GNSS for time transfer, accessible with minimal efforts.

Slides at
`http://jmfriedt.free.fr/241114_grenoble.pdf`
`https://first-tf.com/the-network/network-life/working-groups/impact-study-of-the-loss-of-gnss-signals/`



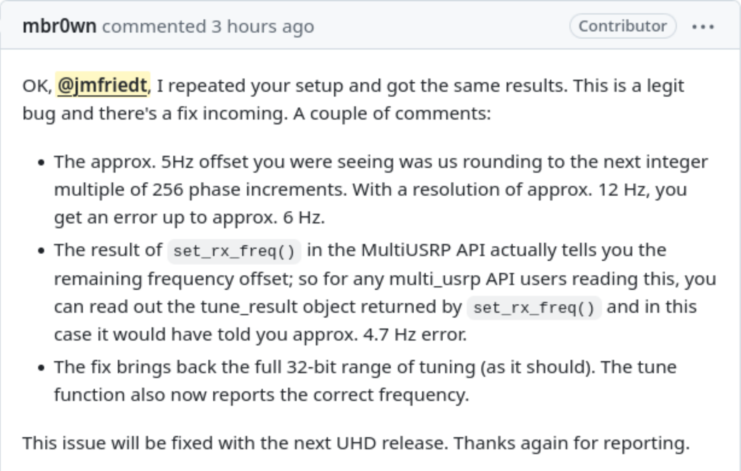
jmfriedt commented on Jun 27 • edited

The DDC of the X310 is not programmed with the right frequency/DDC DDS resolution?

A X310 fitted with two BasicRX (no frequency conversion before sampling and FPGA DDC) is externally clocked by an Octoclock fed by the output of a microstepper generating a 10 MHz and 1-PPS output. This same 10 MHz reference feeds a Rohde & Schwarz SMA100A synthesizer and HP 53131A frequency counter. The synthesizer is programmed to output a 70 MHz continuous wave, validated with the frequency counter:

The X310 is set to a sampling rate of 200/40=5 MS/s and the LO frequency of the DDC set to 70 MHz. The spectrum of the signal recorded for 5 seconds (0.2 Hz resolution) is as follows:

Answer 11/11/2024:



mbr0wn commented 3 hours ago Contributor

OK, @jmfriedt, I repeated your setup and got the same results. This is a legit bug and there's a fix incoming. A couple of comments:

- The approx. 5Hz offset you were seeing was us rounding to the next integer multiple of 256 phase increments. With a resolution of approx. 12 Hz, you get an error up to approx. 6 Hz.
- The result of `set_rx_freq()` in the MultiUSRP API actually tells you the remaining frequency offset; so for any multi_usrp API users reading this, you can read out the tune_result object returned by `set_rx_freq()` and in this case it would have told you approx. 4.7 Hz error.
- The fix brings back the full 32-bit range of tuning (as it should). The tune function also now reports the correct frequency.

This issue will be fixed with the next UHD release. Thanks again for reporting.

Technical issue solved... but some fundamental issue with calculating trigonometric function $\varphi = mod(2\pi \cdot f \cdot t, 2\pi) \in [0 : 2\pi)$
Notice: $\cos(2\pi \times (f/8) \times t) \neq \cos(2\pi \times f \times (t/8))$