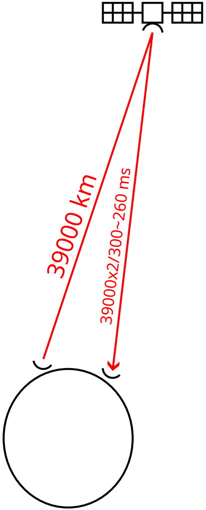
An opensource framework for prototyping Two Way Satellite Time and Frequency Transfer using Software Defined Radio

J.-M. Friedt, G. Goavec-Merou (FEMTO-ST, Besançon)
E. Meyer, F. Meyer (Besançon Observatory)
J. Achkar, M. Lours, M. Dupont, B. Chupin, O. Chiu (SYRTE, Paris Observatory)
with invaluable support from Claudio Calosso (INRIM, Italy)

slides at `http://jmfriedt.free.fr/`
project repository at `https://github.com/oscimp/amaranth_twstft`

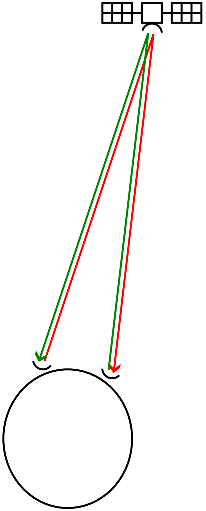
November 8, 2023

Principle of TWSTFT: two-way comparison of the time of flight



Principle of TWSTFT: two-way comparison of the time of flight

- Coordinated Universal Time (UTC): comparison of (atomic) clocks disseminated worldwide
- Various means of time comparison: optical fiber links, global navigation satellite systems (spaceborne atomic clocks), **Two Way Satellite Time and Frequency Transfer** (TWSTFT)
- Exchange timing signals for recovering frequency and ime informations
- **Objective**: sub-200 ps accuracy with 5 MHz available bandwidth and resolution at 1-s integration time ...
- ... using a Software Defined Radio transmitter and receiver



Easy: generate pseudo-random sequence [a], binary phase-modulate the carrier, uplink on a 14-GHz microwave carrier to a geostationnary satellite and receive the 11-GHz downlink, correlate to recover time

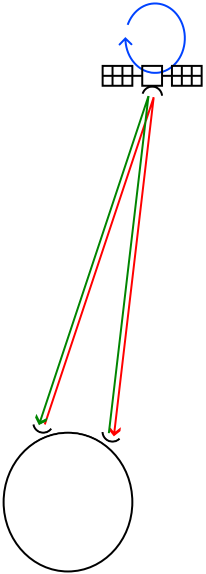
[a] same spectrum spreading technique as used in noise RADAR, see J.-M Friedt, *Software defined radio for noise and passive RADAR processing*, GNU Radio Conference (2021) at `https://pubs.gnuradio.org/index.php/grcon/article/view/74`

Principle of TWSTFT: two-way comparison of the time of flight

- Coordinated Universal Time (UTC): comparison of (atomic) clocks disseminated worldwide
- Various means of time comparison: optical fiber links, global navigation satellite systems (spaceborne atomic clocks), **Two Way Satellite Time and Frequency Transfer** (TWSTFT)
- Exchange timing signals for recovering frequency and ime informations
- **Objective**: sub-200 ps accuracy with 5 MHz available bandwidth and resolution at 1-s integration time ...
- ... using a Software Defined Radio transmitter and receiver

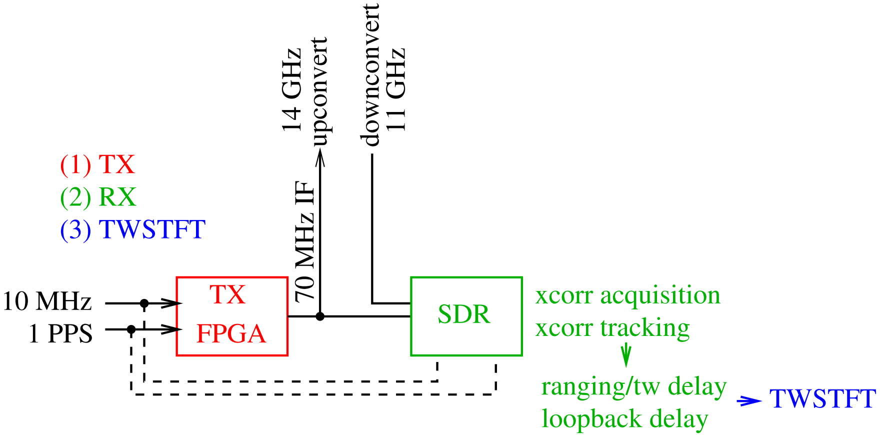
Easy: generate pseudo-random sequence [a], binary phase-modulate the carrier, uplink on a 14-GHz microwave carrier to a geostationnary satellite and receive the 11-GHz downlink, correlate to recover time

- **Geostationary satellite** = fixed location in space?



[a] same spectrum spreading technique as used in noise RADAR, see J.-M Friedt, *Software defined radio for noise and passive RADAR processing*, GNU Radio Conference (2021) at `https://pubs.gnuradio.org/index.php/grcon/article/view/74`

Outline

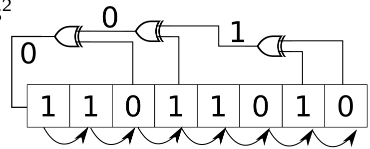


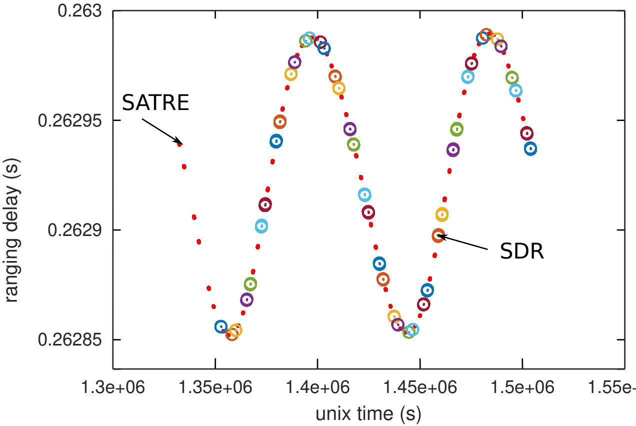
1. Reverse engineer the proprietary TimeTech SATRE modem
2. FPGA based emitter
3. SDR based receiver
4. Signal processing for ranging and time of flight calculation

All steps must be successful to qualify the system

Motion of the satellite

- Satellite allocated bandwidth: 4.x MHz ⇒ 2.5 Mchips/s
- Intermediate Frequency feeding the upconverter: 70 MHz
- 1-PPS timing reference ⇒ 1-s long code (22-bit long)
- Pseudo random sequence generated from a Linear Feedback Shift Register[1] with wisely selected taps[2]





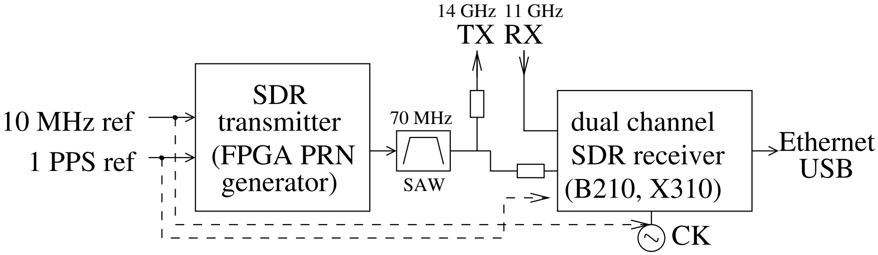
Result: time of flight (ranging) measurement consistent with proprietary SATRE modem communication, but the **satellite is moving by** $\pm 75\ \mu$s (150 μs=45 km) !

[2] Y. Guidon, *Galois et les nombres pseudo-aléatoires*, GNU/Linux Magazine France **261** (Jan. 2023)

[2] `https://users.ece.cmu.edu/~koopman/lfsr/`

SDR emitter and receiver hardware

- Inputs: reference 1-PPS and 10 MHz from metrological source
- FPGA implements Linear Feedback Shift Register pseudo-random sequence (PRN) at 2.5 Mchips/s
- PRN generation controlled by Reset signal and 1-PPS edge
- a 70 MHz sine wave is BPSK modulated (XOR) with the PRN sequence
- GPIO output is filtered by a Surface Acoustic Wave 70 MHz filter (IF)
- IF feeds microwave upconverter and power amplifier on transmitter
- microwave downconverter and low noise amplifier returns 70 MHz IF signal ...
- ... sampled by **dual-channel coherent** SDR board.



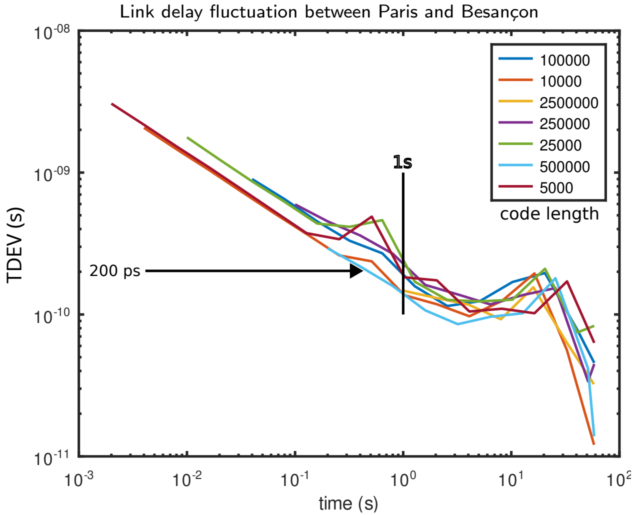
FPGA is either on the SDR board or **external** (faster synthesis)

Code length impact

Code length:

- matched filter for finding a pattern x in y: $xcorr(x,y)(\tau) = \int_N x(t)\cdot y(t+\tau)dt \propto$ power with noise dropping as N
- ... but fixed local copy of the code c in y: $xcorr(c,y)(\tau) = \int_N c(t)\cdot y(t+\tau)dt \propto$ voltage with noise dropping as $\sqrt{N}$
- Shorter code $\Rightarrow$ more averages within 1 s $\Rightarrow$ noise dropping as $\sqrt{N}$

Conclusion: SNR **independent** of code length (longer code increases averaging duration but fewer codes/s) as verified experimentally [a]



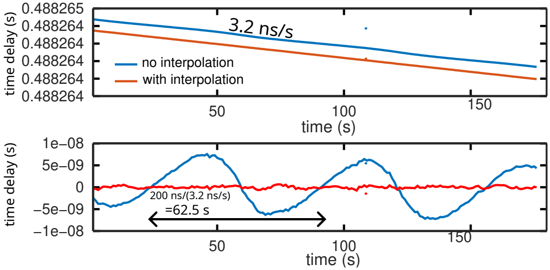
[a] J.-M. Friedt & al., *Development of an opensource, openhardware, software-defined radio platform for two-way satellite time and frequency transfer*, Proc. IFCS (2023)

Correlation

How can we achieve 200 ps delay measurement when sampling at 5 MS/s (200 ns/sampling period)?

- Improve resolution with correlation peak fitting:
- Search for correlation magnitude $|x|$ maximum at position n
- Parabolic fit with samples at position $n-1$, n and $n+1$:

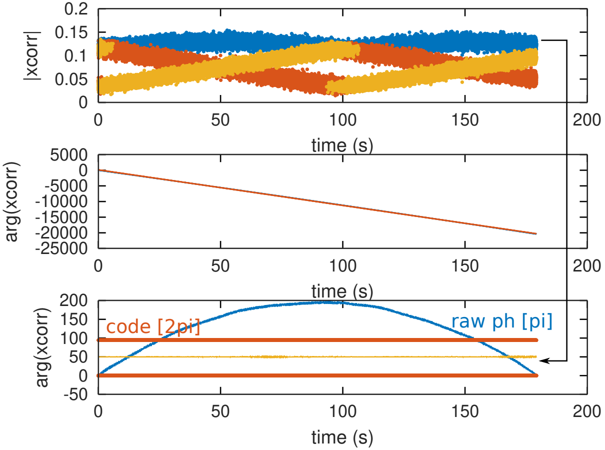
$$dn = \frac{1}{2} \cdot \frac{|x_{n-1}| - |x_{n+1}|}{|x_{n-1}| + |x_{n+1}| - 2|x_n|}$$

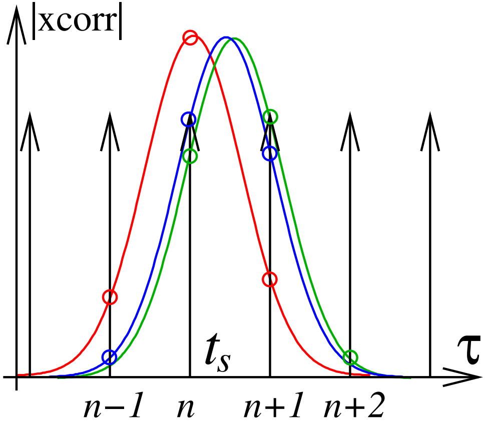


- Correlation peak position improvement = measurement SNR *after correlation* $(+10 \times log_{10}(N))$
- BUT the satellite is moving as we correlate! At 3.2 ns/s, the correlation peak shifts from one sampling period to the next (200 ns) in 62.5 s
- **Solution**: oversampling by interpolation (here 3-fold oversampling)

Low SNR signal fitting issue

Challenge with low SNR signals or drifting time scale: parabolic fit of correlation peak [3] might become unstable.





[3] J.-M Friedt, C. Droit, G. Martin, and S. Ballandras, *A wireless interrogation system exploiting narrowband acoustic resonator for remote physical quantity measurement* Rev. Sci. Instrum. **81**, 014701 (2010)

Correlation with peak fitting and oversampling

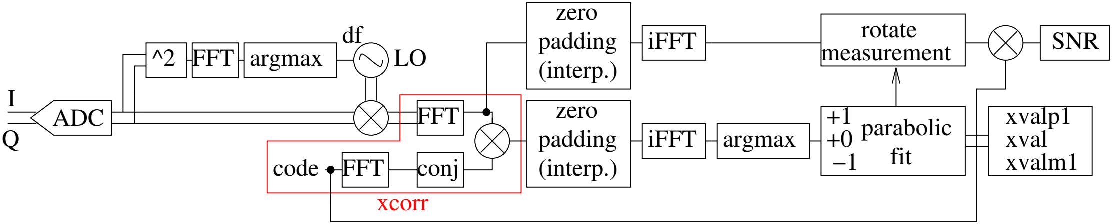
Convolution theorem

$$conv(x,y)(\tau) = \int x(t)\cdot y(\tau - t)dt \rightarrow FT(conv(x,y)) = FT(x)\cdot FT(y)$$

Application to correlation

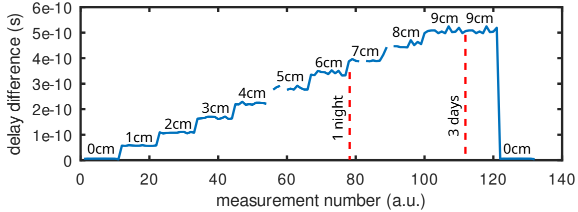
$$xcorr(x,y)(\tau) = \int x(t)\cdot y^*(t+\tau)dt \rightarrow FT(xcorr(x,y)) = FT(x)\cdot FT^*(y)$$

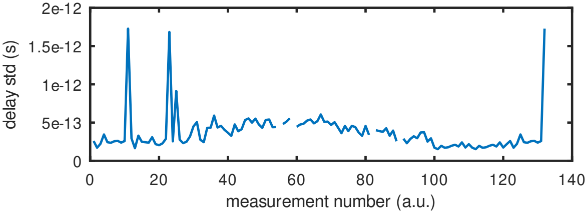
Once we are in the Fourier domain: interpolate by zero-padding $FT(x)\cdot FT^*(y)$ before returnig to time-domain by iFT

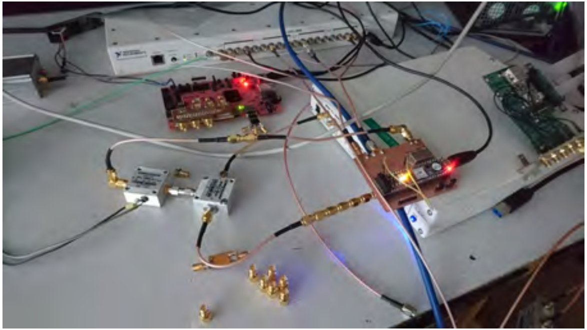


X310 platform delays between channels (external clock)

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





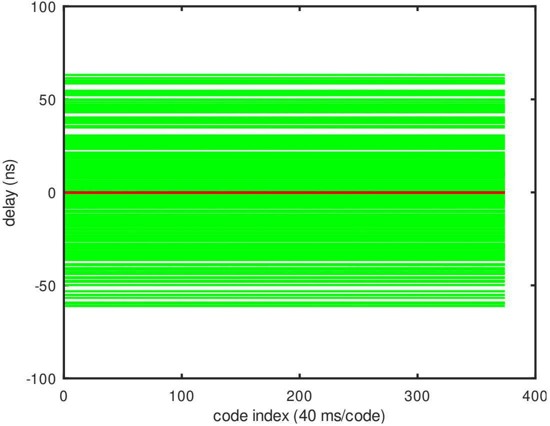


Sub-ps measurement resolution
(1 cm=50 ps @ 20 cm/ns)

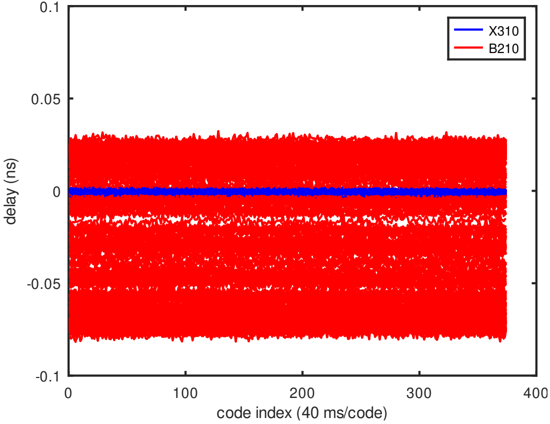
Synchronizing B210 and X310, external clock (Octoclock)

- $\langle \sigma(X310_{1-2}) \rangle = 0.6$ ps, $\langle \sigma(B210_{1-2}) \rangle = 1.5$ ps, $\langle \sigma(B210_1 - X310_1) \rangle = 2.5$ ps within each trace
- $\sigma(\langle X310_{1-2}) = 0.6$ ps, $\sigma(\langle B210_{1-2} \rangle) = 38$ ps, $\sigma(\langle B210_1 - X310_1 \rangle) = 28$ ns within each trace
  $\Rightarrow$ no synchro between X310 and B210 during successive measurements

Green=B210-X310 ; red=B210-B210 ; (blue=X310-X310)

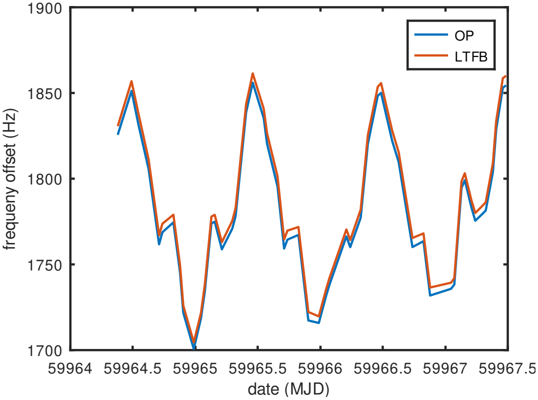


Zoom on B210-B210 ; X310-X310



Frequency offset impact

- Only time delay between correlation peaks can be used for time and frequency transfer: the satellite introduces a frequency offset when transposing from 14 GHz upling to 11 GHz downlink
- $\int c(t) \cdot x(t+\tau) \exp(j\delta\omega(t+\tau))dt$: τ maximizing xcorr is not dependent on $\delta\omega$ *except* through loss of SNR



Frequency offset from the satellite transponder: carrier frequency **cannot** be used for frequency transfer

```
function traite(ref,in)
  s=abs(xcorr(ref,in));
  [~,u]=max(s)
  (s(u-1)-s(u+1))/2/(s(u-1)+s(u+1)-2*s(u))
end

f=fopen('../221207_twoway_codes/codes/noiselen100000_bitlen17_taps09.bin');
code=fread(f,inf,'int8');
code=repelems(code,[[1:length(code)] ; ones(1,length(code))*3]);
a=2*code'-1;
b=a(2:end).*exp(j*0.8);                        traite(a(1:end-2),b);
b=a(2:end).*exp(-j*0.2);                       traite(a(1:end-2),b);
b=a(2:end).*exp(j*2*pi*1e-6*[0:length(a)-2]'); traite(a(1:end-2),b);
b=a(2:end).*exp(j*2*pi*4e-6*[0:length(a)-2]'); traite(a(1:end-2),b);
b=a(2:end).*exp(j*2*pi*12e-6*[0:length(a)-2]');traite(a(1:end-2),b);
b=a(2:end).*exp(j*2*pi*40e-6*[0:length(a)-2]');traite(a(1:end-2),b);
```

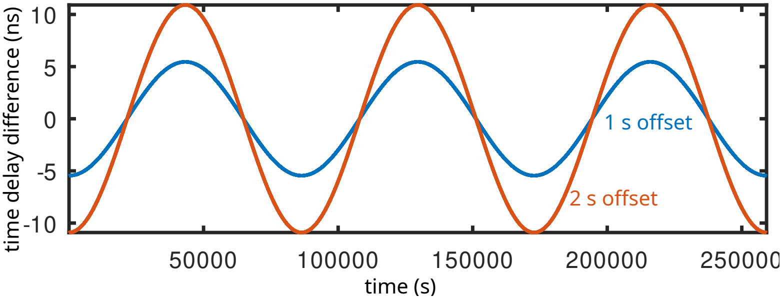
2 static phase cases and 4 frequency offset cases

```
u = 300000 ans = -5.0108e-06
u = 300000 ans = -5.0108e-06
u = 300000 ans = -3.4308e-06
u = 300000 ans = -2.6150e-05
u = 300000 ans = 1.8744e-05
u = 290580 ans = 0.031081
```

25 Hz accuracy at 5 MS/s$= 5 \cdot 10^{-6} \Rightarrow 200$ ns $\cdot 10^{-5} = 2$ ps

Impact of time synchronization (NTP synchronized computers)

What if the two-way processing is not applied to the same time-transfer delay measurement?




```
temps=[1:86400*3];
sinus=75e-6*sin(2*pi*temps/86400);
plot((sinus(1:end-1)-sinus(2:end))*1e9);
hold on
plot((sinus(1:end-2)-sinus(3:end))*1e9);
xlabel('time (s)')
ylabel('delay difference (ns)')
```

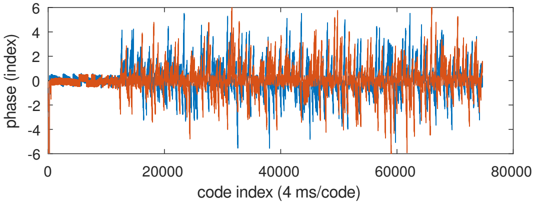
velocity $\leq$ 5 ns/s $\Rightarrow$ *dt* $<$ 20 ms for sub-100 ps accuracy achievable with NTP-synchronized transceivers

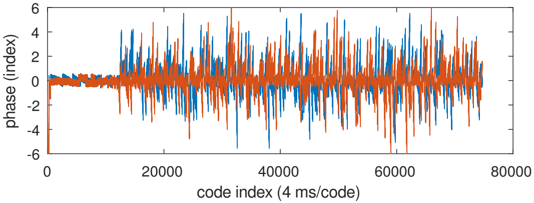
On both ends the sequence is generated by the metrological 1-PPS timing reference, but

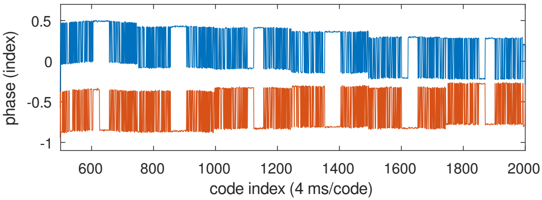
1. make sure we are comparing the same pseudo-random sequence (tag beginning of second if code length $<$ 1 s)
2. make sure we are comparing the same second

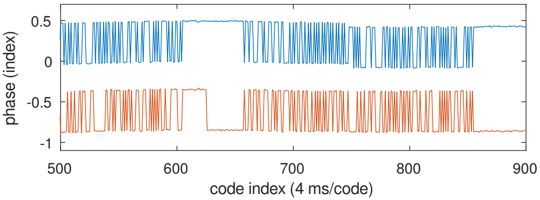
DLL and PLL (phase tracking)

- Initial approach was a basic signal acquisition using the cross correlation
- Time delay artifact $\Rightarrow \pi$ independent PLL (frequency offset) and DLL (sampling rate)
- DLL divergence after some time?



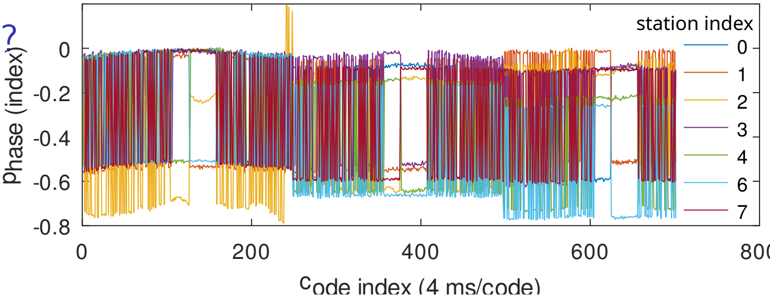


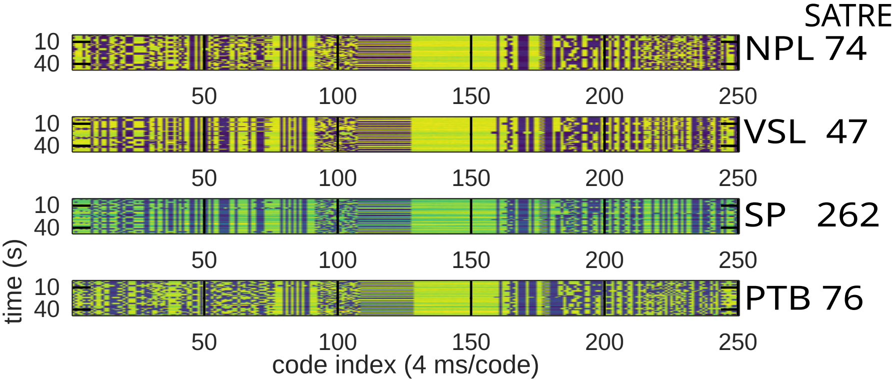




Digital bit payload content?

- From the π dependent phase calculation (atan2(Q,I)): recover payload bit states $\longrightarrow$
- Stack sentences assuming 250 bps (1/4 ms) $\downarrow$

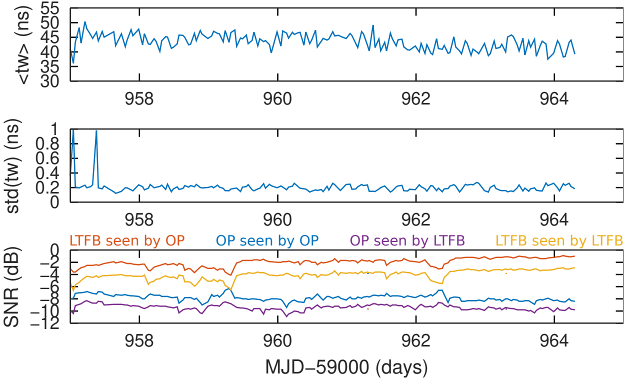




Conclusion

- Demonstrated two-way satellite time and frequency link using SDR transmitter and receiver between Paris Observatory (OP) and Besançon (LTFB)
- Identified key parameters (code length, timing accuracy)
- Functional implementation including first correlation inversion at the beginning of each second
- Implemented all post-processing analysis scripts

```
https://github.com/oscimp/amaranth_twstft
```



Work in progress

- What additional information to include in the transmitted sequences? (time? date? ID?)
- Code length selection: not driven by SNR considerations but by payload (short code = more bits)
- Complete TWSTFT post-processing sequence: at the moment, random fluctuations of ±X ns from one measurement to the next

“Scientific” production

1. J.-M Friedt, *Le temps et son transfert par satellite géostationnaire : réception avec une parabole de télévision et d’une radio logicielle*, GNU/Linux Magazine France Hors Série 121 (Oct-Nov. 2022)
2. 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)
3. J.-M Friedt, G. Goavec-Merou, *Passive reception of Two-Way Satellite Time and Frequency Transfer (TWSTFT) signals from a geostationary satellite, or GPS upside down*, GNU Radio Conference (2022)
4. J.-M Friedt, G. Goavec-Merou, *Measuring time delays with sub-sampling period resolution: qualification of some COTS SDR RF frontends with sub-100 ps resolution*, Software Defined Radio Academy (2023)
5. J.-M Friedt, G. Goavec-Merou, *Measuring time delays with sub-sampling period resolution: qualification of some COTS SDR RF frontends with sub-100 ps resolution*, GNU Radio Conference (2023)
6. J.-M Friedt, M. Lours, G. Goavec-Merou, M. Dupont, B. Chupin, O. Chiu, É. Meyer, F. Meyer, J. Achkar, *Development of an opensource, openhardware, software-defined radio platform for two-way satellite time and frequency transfer*, Proc. IEEE/IFCS (2023)

`https://iqengine.org/`: GNU Radio → space → Telstar11N

`https://github.com/oscimp/amaranth_twstft`

`https://github.com/oscimp/gr-satre`

+ 2 ESA projects (Galileo E6 time transfer validation) + Euramet project submission.

“Scientific” production

`https://iqengine.org/`: GNU Radio → space → Telstar11N

`https://github.com/oscimp/amaranth_twstft`

`https://github.com/oscimp/gr-satre`

+ 2 ESA projects (Galileo E6 time transfer validation) + Euramet project submission.