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)

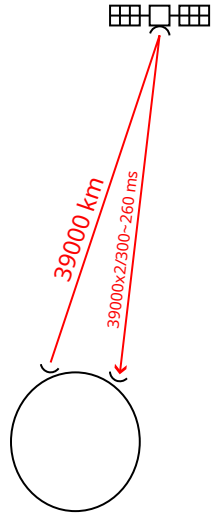
slides at `http://jmfriedt.free.fr/`
project repository at `https://github.com/oscimp/amaranth_twstft`

March 28, 2023

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 time informations
- **Objective**: sub-200 ps accuracy 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

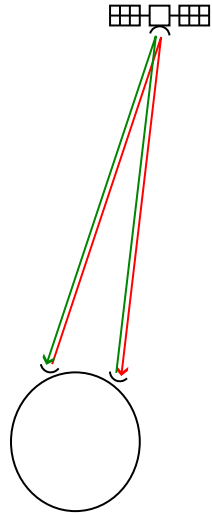


[a] same spectrum spreading technique as used in noise RADAR, see J.-M Friedt, *Software defined radio for noise and passive RADAR processing*, GNU Radio Conference (2021) at `https://pubs.gnuradio.org/index.php/grcon/article/view/74`

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 time informations
- **Objective**: sub-200 ps accuracy 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



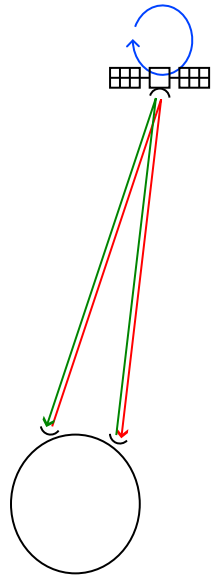
[a] same spectrum spreading technique as used in noise RADAR, see J.-M Friedt, *Software defined radio for noise and passive RADAR processing*, GNU Radio Conference (2021) at `https://pubs.gnuradio.org/index.php/grcon/article/view/74`

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 time informations
- **Objective**: sub-200 ps accuracy 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

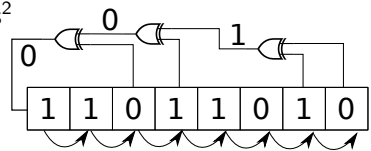
- **Geostationary satellite** = fixed location in space?

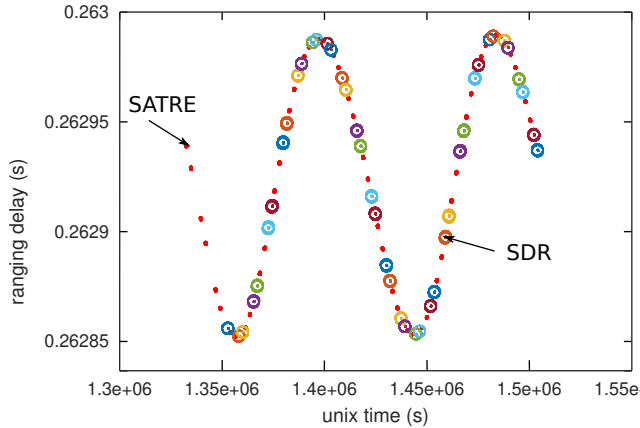


[a] same spectrum spreading technique as used in noise RADAR, see J.-M Friedt, *Software defined radio for noise and passive RADAR processing*, GNU Radio Conference (2021) at `https://pubs.gnuradio.org/index.php/grcon/article/view/74`

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]



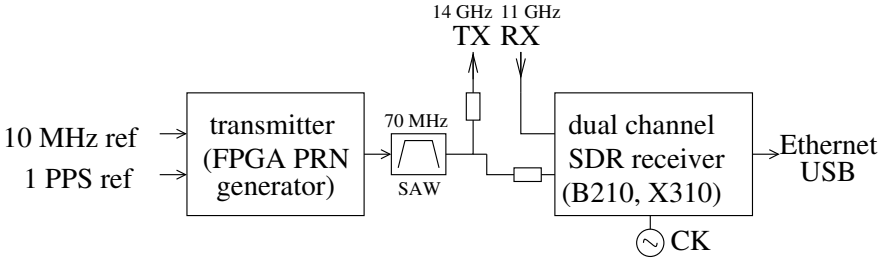


Result: time of flight (ranging) measurement consistent with proprietary SATRE modem communication, but the **satellite is moving by** ±75 μs (150 μs=45 km) !

[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.



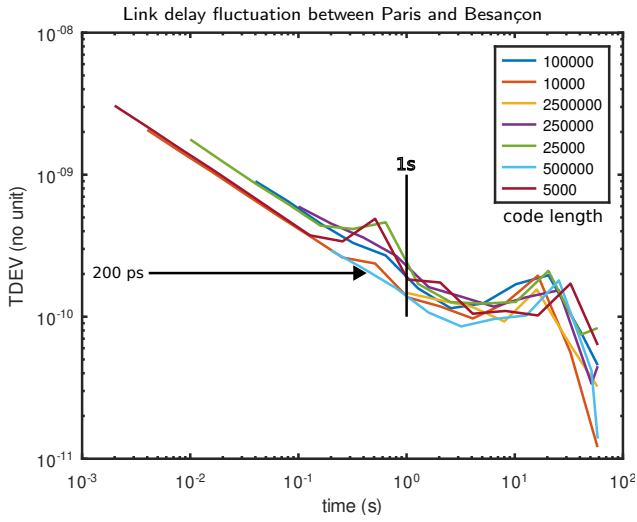
FPGA is either on the SDR board or external (faster synthesis)

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]



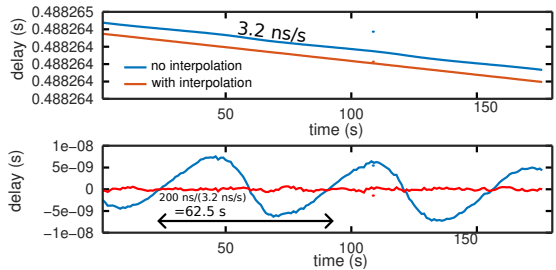
[a] J.-M. Friedt *& al.*, *Development of an opensource, openhardware, software-defined radio platform for two-way satellite time and frequency transfer*, Proc. IFCS (2023)

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)

Correlation with peak fitting and oversampling

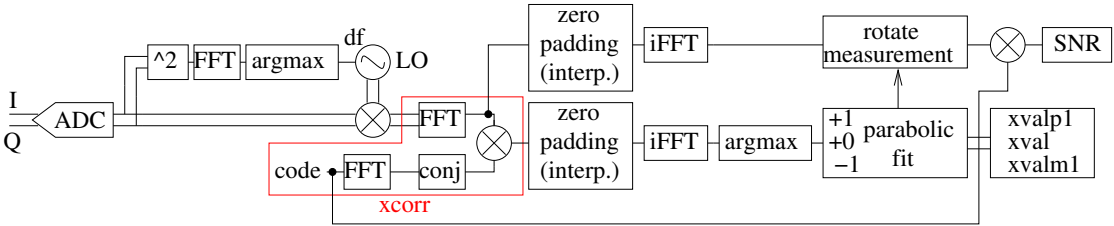
Convolution theorem

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

Application to correlation

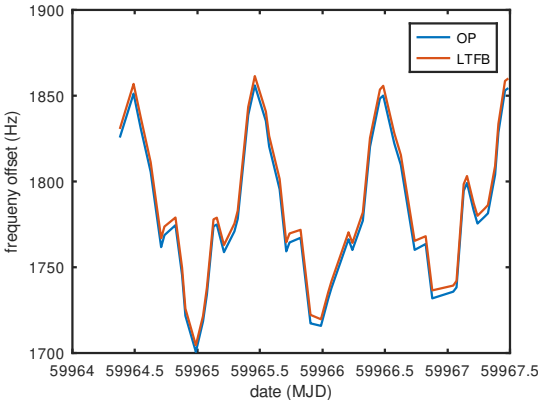
$$xcorr(x,y)(\tau) = \int x(t) \cdot y^*(t + \tau)dt \rightarrow FT(xcorr(x,y)) = FT(x) \cdot FT^*(y)$$

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



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);
```

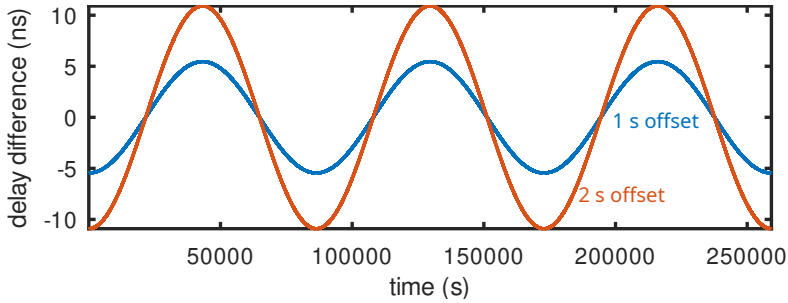
2 static phase cases and 4 frequency offset cases

```
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$= 5 \cdot 10^{-6} \Rightarrow 200\ \text{ns} \cdot 10^{-5} = 2\ \text{ps}$

Impact of time synchronization

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

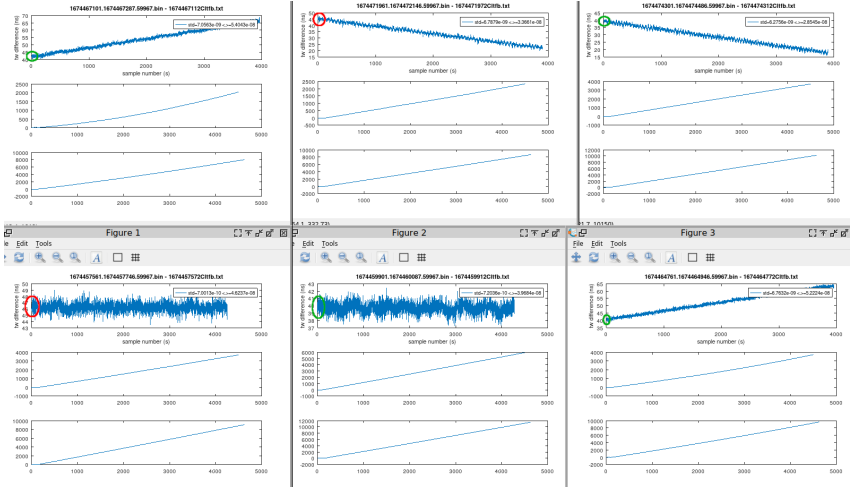


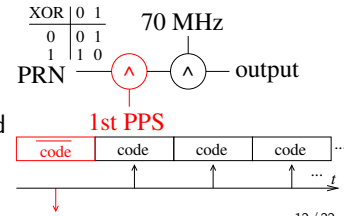

```
temps=[1:86400*3];
sinus=75e-6*sin(2*pi*temps/86400);
plot((sinus(1:end-1)-sinus(2:end))*1e9);
hold on
plot((sinus(1:end-2)-sinus(3:end))*1e9);
xlabel('time (s)')
ylabel('delay difference (ns)')
```

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

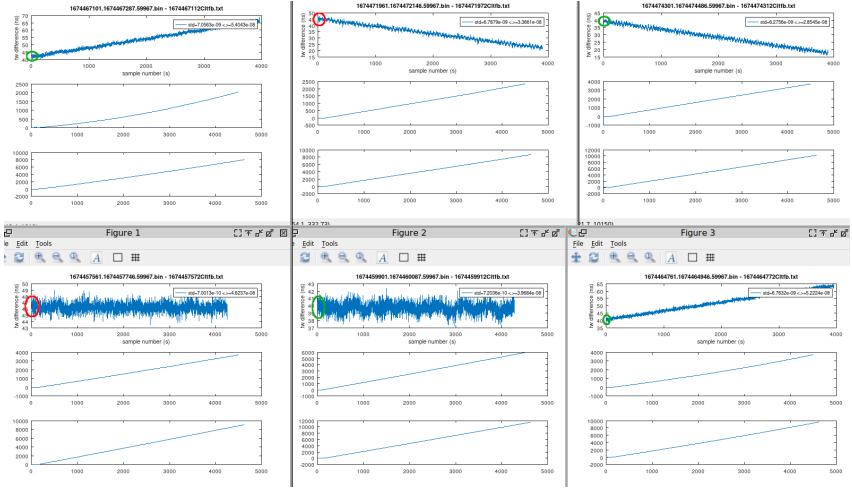
If the satellite is moving at X ns/s, then a 1 s error is observed as X ns offset, $X \ll 1/f_s$

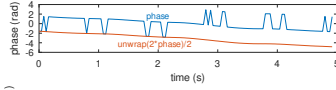


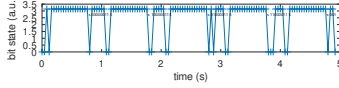


- Need to encode **NTP timestamp (at least seconds)** in the transmitted signal
- Shorter code allows for more accurate time alignement on both ends

If the satellite is moving at X ns/s, then a 1 s error is observed as X ns offset, $X \ll 1/f_s$

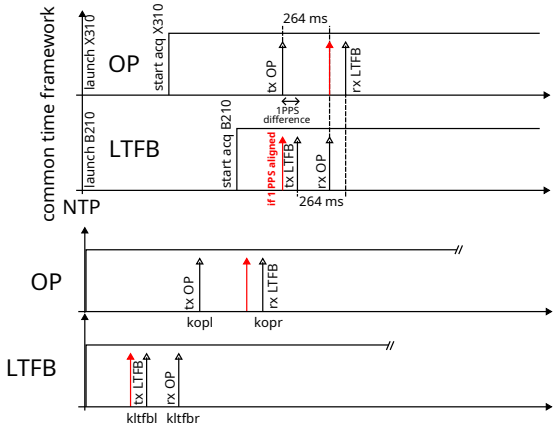






- Need to encode **NTP timestamp (at least seconds)** in the transmitted signal
- Shorter code allows for more accurate time alignement on both ends

Impact of time synchronization

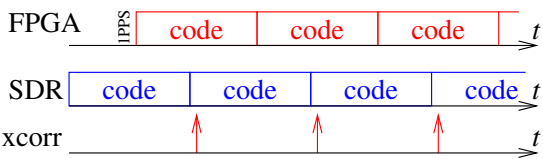


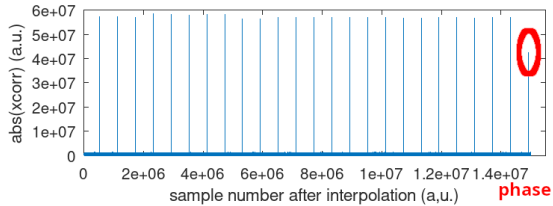
- On both ends the emission script is started by an NTP synchronized crontab
- On both ends UHD initializes their respective SDR boards (X310, B210), introducing a random delay
- On both ends the FPGA starts emitting their respective PRN code at the 1-PPS mark after the recording started
- On both ends the sampling duration (3 minutes) is known within one sample accuracy
- On both ends the end of the acquisition date is stored ("last access time" argument of the file)

Code alignement

Rather than observing the code position at a given index, align the received signal on the local copy of the code to avoid cyclic correlation between two sets of chips
$\Rightarrow$ removes correlation peak drift during a session

- One big correlation and search for correlation peaks every code length or small correlation on datasets incremented by code length samples? Correlation cyclicity hypothesis (thanks C. Calosso, INRIM)
- `real()` v.s. `abs()`?

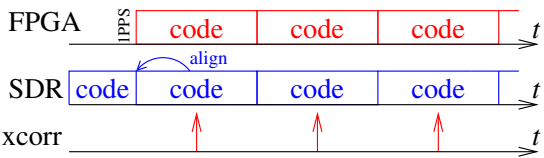


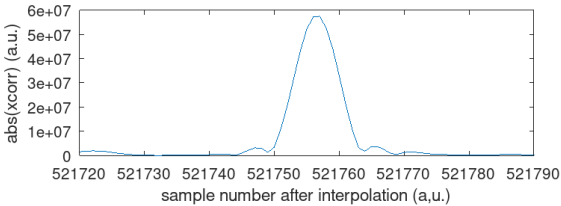


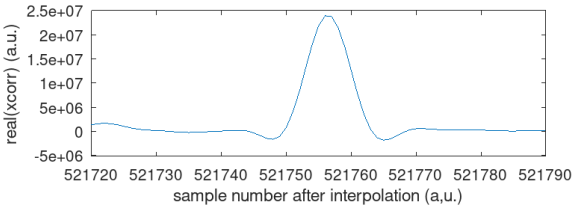
Code alignement

Rather than observing the code position at a given index, align the received signal on the local copy of the code to avoid cyclic correlation between two sets of chips
$\Rightarrow$ removes correlation peak drift during a session

- One big correlation and search for correlation peaks every code length or small correlation on datasets incremented by code length samples? Correlation cyclicity hypothesis (thanks C. Calosso, INRIM)
- `real()` v.s. `abs()`?

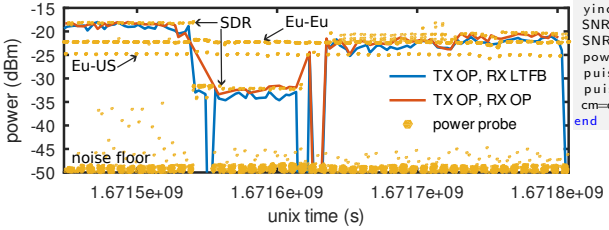






Signal to noise ratio calculation

- Signal is $\int x(t) \cdot code(t+\tau)dt$ for τ maximizing the correlation
- Noise is $\int x^2(t)dt$
- $\Rightarrow$ SNR $\in [-20 : -5]$ dB or C/N0 $\in [47 : 62]$ dB.Hz considering the 5 MHz sampling rate

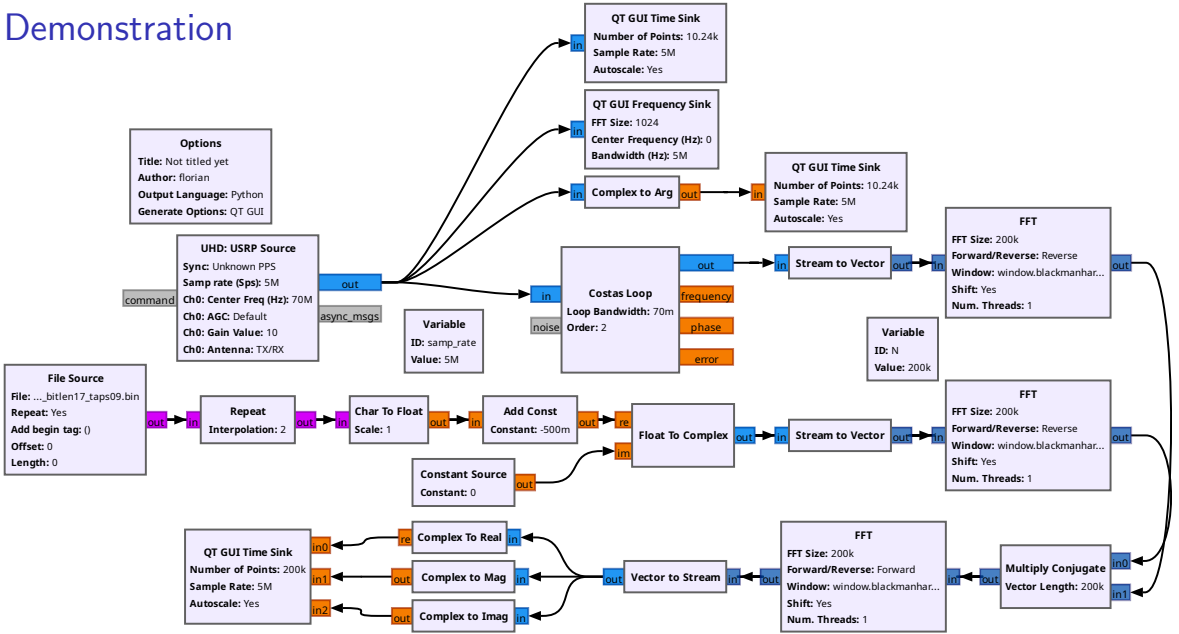


From equations to implementation (post-processing):

```
lo=exp(-j*2*pi*df*temps);                    % local oscillator
y=d.*lo;                                     % frequency transposition
ffty=fft(y);                                 % FFT(xcorr(x,y))=FFT(x).FFT*(y)
cx=fftshift(fcode.*conj(ffty));              % v 0-padding xcorr = interpolation
cx=[zeros(length(y)*(Nint),1) ; cx ; zeros(length(y)*(Nint),1)];
cx=(ifft(fftshift(cx)));                     % back to time-domain
yint=zeros(length(y)*(2*Nint+1),1);  % interpolate dataset by 0-padding
yint(1:length(y)/2)=ffty(1:length(y)/2); % for code-dataset alignement
yint(end-length(y)/2+1:end)=ffty(length(y)/2+1:end);
yint=ifft(yint);  % vv match code length and interpolated dataset length
ci=repelems(code,[[1:length(code)] ; ones(1,length(code))*(2*Nint+1)])';
cm=1;     % vv loop every code length elements and analyze xcorr peak/SNR
for cidx=1:length(code)*(2*Nint+1):length(cx)-length(code)*(2*Nint+1)+1
 [~,indice(cm)]=max(abs(cx(cidx:cidx+length(code)*(2*Nint+1)-1)));
 vl=cx(indice(cm)+cidx-1);                   % xcorr maximum
 vlm1=cx(indice(cm)+cidx-1-1);               % neigbours for parabolic fit
 vlp1=cx(indice(cm)+cidx-1+1);               % v analytical solution of position
 crr(cm)=(abs(vlm1)-abs(vlp1))/(abs(vlm1)+abs(vlp1)-2*abs(vl))/2;
 yintmp=yint(cidx:cidx+length(code)*(2*Nint+1)-1); % select data
 yincode=[codetmp(indice(cm)-1:end) ; codetmp(1:indice(cm)-2)].*yintmp;
 SNRr(cm)=mean(real(yincode))^2/var(yincode);       % ^ rotate code
 SNRi(cm)=mean(imag(yincode))^2/var(yincode);
 power(cm)=var(y);
 puissancecode(cm)=mean(real(yincode))^2+mean(imag(yincode))^2;
 puissancenoise(cm)=var(yincode);
 cm=cm+1;
end
```

https://github.com/oscimp/gr-satre/hirate_digital_mode/claudio_aligned_code.m

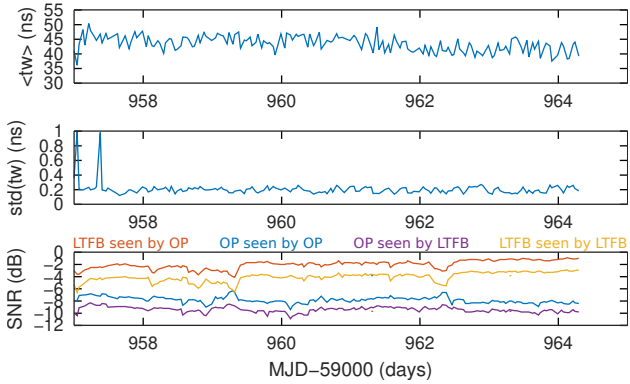
Demonstration



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`



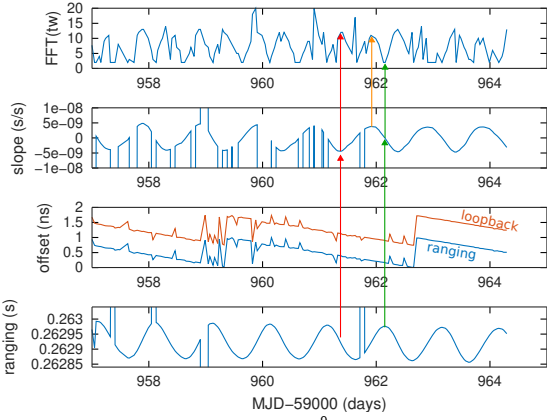
Work in progress

- 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 *pm*X ns from one measurement to the next

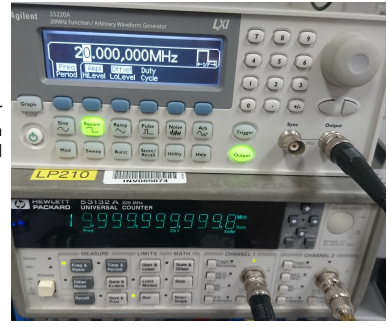
FIRST TF

PUBLIC MONEY
PUBLIC CODE

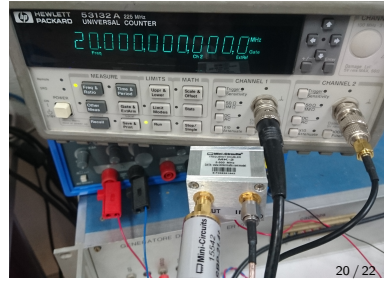
Conclusion



1 ns delay over 40 h is $\frac{10^{-9}}{40 \cdot 3600} = 7 \cdot 10^{-15}$ or at 20 MHz a frequency offset of 14 nHz on **both** the reference and measurement channels (since offset on FPGA reference clock), visible on the hydrogen maser disciplined B210



Direct digital synthesizer (square wave output with offset) and counter clocked by hydrogen maser



Counter clocked by hydrogen maser and doubled hydrogen maser (requires a sine wave to square wave conditioning circuit to clock the FPGA)

Conclusion

Experimental setup