Time of flight measurement with sub-sampling period resolution using Software Defined Radio

J.-M Friedt, G. Goavec-Merou JMFRIEDT@FEMTO-ST.FR
FEMTO-ST Time & Frequency, Besançon, France

Abstract

Timing signal delay difference with sub-sampling period resolution is achieved by spectrum spreading using a pseudo-random sequence generator and fitting the cross-correlation peak of the received signal. An experimental demonstration on various radiofrequency software defined radio (SDR) receivers demonstrates sub-ps synchronization on the Ettus Research X310 BasicRX frontend, picosecond synchronization of the AD9361 fitted on the Ettus Research B210, and a drift of a few hundred picoseconds during measurements lasting a few seconds with the Lime Microsystems LMS7002 fitted on the Fairwaves XTRX. All measurements are achieved with 200 ns sampling period or a rate of 5 Msamples/s, emphasizing the timing resolution improvement of at least 1000-fold over the sampling period.

As is known from the classical RADAR range resolution equation, the time of flight resolution dt is solely determined by the signal bandwidth B as $dt = 1/B$, leading to the range resolution $\Delta R = c_0/(2B)$ with $c_0 = 300$ m/μs the speed of light in free space and the $1/2$ term representing the two-way trip in a monostatic system. In this generic case where no assumption is made on the structure of the signal backscattered by targets, the delay resolution is determined by the broadcast signal spectrum occupation. The pulsed RADAR provides an intuitive illustration of the range resolution limited by the sampling rate since the backscattered echoes will be detected either as one sample or the next, and hardly any signal processing can improve the resolution below the sampling period. However by spreading the spectrum using frequency sweep (Frequency Swept Continuous Wave RADAR – FSCW) or by broadcasting a pseudo-random sequence (noise RADAR), the cross-correlation accumulates energy coherently only at the time delay representative of the time of flight to a target.

Proceedings of the 13th GNU Radio Conference, Copyright 2023 by the author(s).

The magnitude of the cross-correlation between the transmitted signal and received signal exhibits a shape that hints at oversampling by fitting the peak under the assumption of a unique isolated reflector.

In the numerical experiment of Fig.1, this concept is tested by generating a *known* random sequence – the autocorrelation of noise being a Dirac function at 0-delay – and spreading the energy within each cell to adjacent samples by convolution. Energy spreading allows for assessing the behavior of the cross-correlation when time delays smaller than the sampling period, now equal to the convolution length, are introduced between the reference and measurement channels.

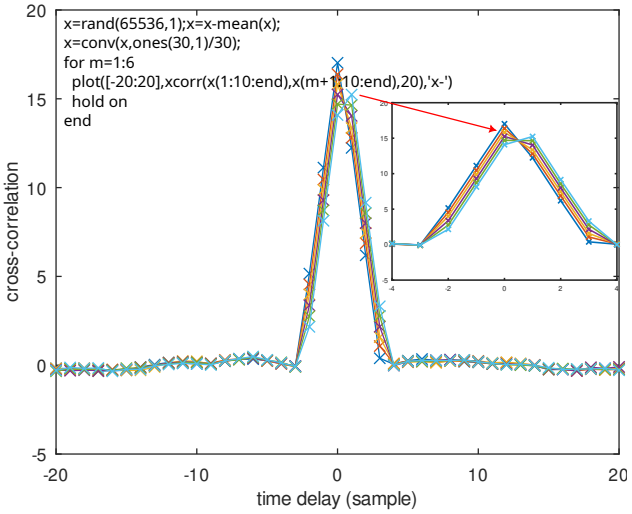

Figure 1. Cross-correlation peak shape evolution as the measurement signal is shifted by sub-sampling period delays: the code inset on the top-left illustrates the energy spreading using the convolution to numerically simulate sub-sampling period time delay.

The code inset of Fig.1 (top-left) running in GNU/Octave highlights how time delays `m` ranging from 1 to 6 lead to the shape of the cross-correlation peak magnified in the inset to the right: while a coarse maximum detection would always identify a delay index of 0 except for the light blue curve when the maximum is shifted to index 1, the continuous shape change of the correlation magnitude around its max-

imum, with samples adjacent to the cross-correlation peak maximum slowly rising and falling, hints at the ability to fit the peak and finely oversample to improve the delay estimate. Indeed it can be demonstrated that the **time delay resolution is improved by using a second-order polynomial (parabola) peak fitting by a factor equal to the signal to noise ratio**, with noise being the cross-correlation output noise on each sample and the signal being the magnitude difference between the cross-correlation peak and its two adjacent neighbors (Friedt et al., 2010).

This concept is demonstrated experimentally in Fig.2 in which a sound card sampling at 192 ksamples/s is used to record the 77 kHz timing signal broadcast from DCF77 emitter located in Mainflingen in Germany, 400 km from the receiver location, to assess the time delay with respect to GPS 1-PPS recorded by the second stereo channel. Although DCF77 phase-modulates a 645 Hz wide pseudo-random sequence over the classical amplitude modulation for improved resolution as described at `https://www.eecis.udel.edu/~mills/ntp/dcf77.html`, oversampling the correlation magnitude by peak fitting allows for assessing time delays well below the $1000/645 = 1.55$ ms inverse of the bandwidth (Friedt et al., 2018).

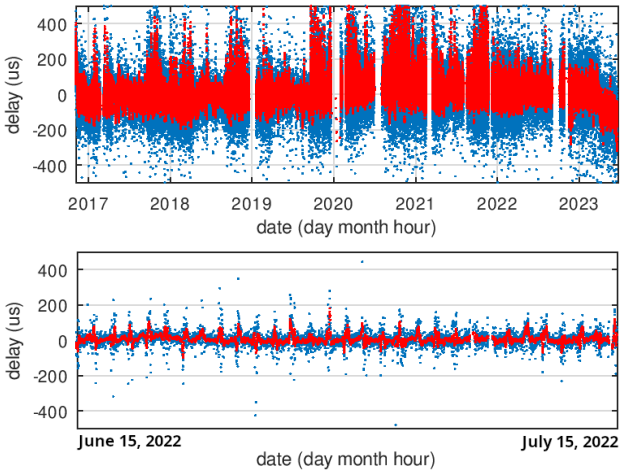

Figure 2. DCF77 time delay with respect to GPS 1-PPS reference signal recorded over multiple years exhibits a resolution well below the inverse of the bandwidth occupied by the phase modulation scheme (1.55 ms). Bottom: zoom on the time delay emphasizing the ability to observe daily fluctuations associated with sunrise and sunset impacting the ionosphere altitude and hence time of flight from Mainflingen to the receiver at a range of 400 km.

The outline of the paper is hence to assess the time delay measured between channels of the Ettus Research X310 fitted with BasicRX frontends (straight connection from input to the analog to digital converter (ADC) with no radiofrequency band processing), the Ettus Research B210 fitted with the Analog Devices AD9361 frontend performing frequency transposition by mixing with a local oscillator and variable gain amplifiers, and the Fairwaves XTRX fitted with the Lime Microsystems LMS7002 frontend providing functionalities similar to the AD9361. Results are obtained by using a dedicated pseudo-random sequence generator allowing to finely assess the behavior of each receiver, always focusing on a **differential** measurement, hence focusing on dual channel receivers.

1. Pseudo-random signal generator

A Field Programmable Gate Array (FPGA) is configured to run a Linear Feedback Shift Register (LFSR) Pseudo-Random Number (PRN) generator with tunable length: the longer the code, the better the pulse compression capability by rejecting noise over a longer averaging time and accumulating energy in the correlation peak, but the lower the datarate if a payload is to be broadcast over the Code Division Multiple Access (CDMA) signal. Throughout this investigation, a 10^5-long code is generated at a rate of 2.5 Mchips/s, or a 40 ms long code sequence repeated continuously. This PRN sequence defines the phase – 0 rad for a 0-bit state or flipped to π for a 1-bit state – of a 70 MHz numerically controlled oscillator generated by the FPGA. This binary phase shift keying intermediate frequency of 70 MHz – as classically used for space communication – is then split and feeds the two inputs of the dual-channel SDR receiver under investigation. The FPGA is clocked from an external 10 MHz frequency source and the sequence generation is triggered by the rising edge of the associated 1-PPS (Pulse Per Second) timing signal.

To comply with radiofrequency spectrum occupation regulations and allocated channel bandwidth when communicating with a geostationary satellite, a surface acoustic wave (SAW) filter centered on 70 MHz and 2.6 MHz wide (Sawtek 851547) with better than 40 dB rejection beyond 4 MHz from the carrier frequency is included at the output of the FPGA whose GPIO pin is toggling between binary states. The 25 dB insertion loss introduced by the SAW filter is either compensated for by tuning the internal preamplifiers of the AD9361 or LMS7002, or adding external amplifiers prior to the splitters to reach the 6 dBm needed to run the X310 analog to digital converters at full-scale range.

The implementation of the PRN generator and the BPSK or QPSK (in the latter case using the odd PRN samples for modulating I and even PRN samples for modulating Q) aims at platform independence by representing the hardware configuration in the Amaranth language as described at `https://github.com/oscimp/amaranth_twstft`. Although the PRN gen-

erators could be running on the B210 or X310 FPGAs, faster synthesis is achieved for a bare FPGA and all demonstrations are completed by running the gateware on a Pynq-Z2 board fitted with a Xilinx Zynq 7020 System on Chip.

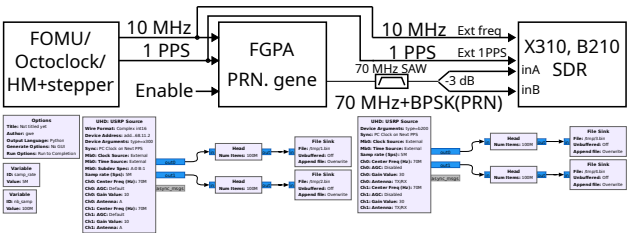

Figure 3. Experimental setup: a clock generator drives the frequency (10 MHz) and time (1-PPS) information of the FPGA generating the PRN sequence BPSK-modulating a 70 MHz carrier. The 70 MHz FPGA output is band-pass filtered and split to feed both inputs of SDR receivers also clocked by the same reference signals (top) unless stated otherwise in the text. The cross-correlation is computed during post-processing of the datastream generated by GNU Radio (bottom) from one or two SDR receivers.

The experimental setup principle is illustrated in Fig.3 and a picture shown in Fig.4, with an Ettus Research Octoclock generating the 10 MHz and 1-PPS references distributed to the FPGA and the SDR receivers, here an X310 and a B210 or two X310s. The objective of the analysis is

1. to assess how the 1-PPS timing input to the SDRs allows to synchronize the channels, here with sub-sampling period accuracy
2. how the external PRN signal allows for synchronizing multiple SDR receivers, with each possibly tuned to a difference carrier frequency and hence monitoring different frequency bands as would be needed for assessing the timing accuracy of Global Navigation Satellite Systems (GNSS) broadcasting in widely different bands well beyond the bandwidth of each individual receiver (e.g. 1176.45 MHz for GPS L5, 1227.6 MHz for GPS L2, and 1575.42 MHz for GPS L1 all broadcasting timing information for trilateration of the receiver).

2. Results

2.1. Absolute delay measurement

The first investigation aims at assessing whether the 1-PPS external trigger allows for synchronizing both channels of a given SDR receiver on the incoming trigger pulse and hence provide absolute time delay of the samples with respect to the input 1-PPS. As shown on Fig.5, within each

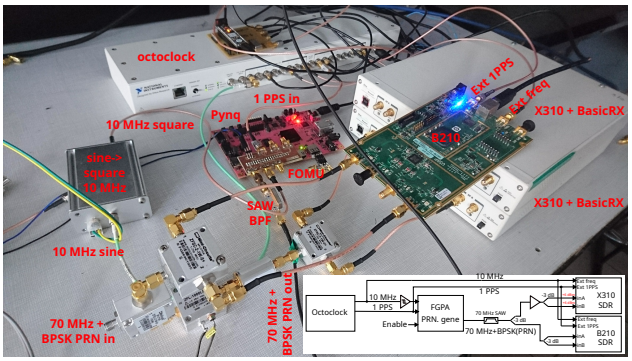

Figure 4. Implementation of the experimental setup, with the Octoclock output converted to a square wave compatible with driving the FPGA using a fast comparator.

acquisition the standard deviation on the correlation peak relative delay within each record fluctuates by a few picoseconds (to be quantified later in this document) but the absolute delay is randomly distributed within the sampling period of 200 ns since all sampling rates in this document are 5 Msamples/s.

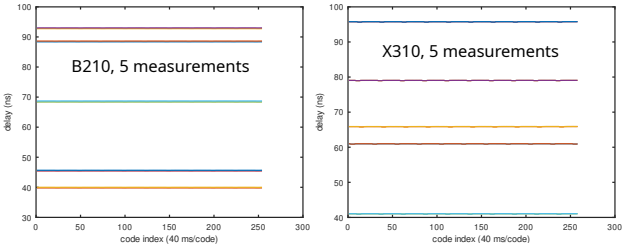

Figure 5. Single channel delay when repeating the 10-s long measurement multiple times. Each measurement is randomly distributed within the sampling period, although both channels are offset by the same random value: each trace is two curves overlapping on such a broad scale ranging from 30 to 100 ns in the Y-axis. On the left the measurement for a dual-channel B210 receiver, on the right for the X310 receiver.

Hence, the 1-PPS timing reference does **not** allow synchronizing measurements from one acquisition to the next with sub-sampling period: this result is expected if the 1-PPS is sampled prior to a Phase Locked Loop rising the clock frequency within the FPGA or if the clock is resampled by the ADC. Hence, from now on we shall focus on *relative* time delays between both dual-channel SDR inputs and assess whether distributing the PRN sequence on one channel allows for assessing the time offset of each SDR while the second channel would be used for recording the signal of interest. In this demonstration, both reference and measurement channels will be fed the same PRN for performance analysis.

2.2. Relative delay measurement

Despite absolute delays being randomly distributed within the sampling period, it is observed that both channels follow the same trend during each acquisition sequence, so we consider from now on the relative delay of one channel with respect to the other. Similar to the conditions of the previous run, a single X310 receiver with both inputs fitted with BasicRX baluns is fed the same PRN sequence BPSK-modulated over the 70 MHz IF carrier, sampled at 200 MS/s and downconverted to baseband at a datarate of 5 MS/s by the X310 gateware. The resulting IQ datastream is sent to a general purpose computer for post-processing: the cross-correlation fit is computed for 10 repetitions of each measurement. The standard deviation within each 10-second long measurement is observed to be in the **35 ps range when no amplifier** is inserted between the FPGA output, SAW filter and X310. Under such conditions, the 3.3 Vpp output of the FPGA GPIO pin feeding a $50\ \Omega$ load generates an output power of 14 dBm so that after the 25 dB losses induced by the SAW filter only -11 dBm reach the X310 input, well below the full scale range of 6 dBm of the X310 ADC, hence the poor standard deviation which is still 1/6000th of the sampling period.

By amplifying the FPGA output by 15 dB to compensate for the SAW insertion losses, the full scale range of the X310 is used and the standard deviation drops to **0.5 ps** within each 10-second long measurement. To demonstrate how reproducible this result is, the measurement is repeated by adding SMA extensions (Amphenol SMA5071A1-3GT50G-50) on one path and keeping the other one the same length. Fig.6 demonstrates how the standard deviation remains in the 0.5 ps range throughout the measurement (bottom) but the delay between both channels rises as each extension is added. Each delay difference is measured while adding the 9 SMA extensions as $0.5\ \text{ns}/9 = 0.056$ ns for each additional extension. This result is analyzed in terms of additional cable length as $0.056\ \text{ns} \times 20\ \text{cm/ns} = 1.12$ cm assuming a speed of light of $200\ \text{m}/\mu\text{s}$ in coaxial cables. The geometric length of each extension is measured as $11.5\ \text{cm}/9 = 1.28$ cm: the time of flight measurement matches the geometric measurement to better than 15% with large uncertainties attributed to the tabulated velocity of the electromagnetic wave in a coaxial cable and the electrical length estimate. The reproducibility of the measurement is furthermore emphasized by comparing results collected half a day apart between the addition of the 6th and 7th extension, or 3 days apart with the 9th extension kept in one of the arms. The 0.5 ps standard deviation on the relative delay demonstrates an improvement of $4 \cdot 10^5$ over the sampling period, with such a high ratio allowed by the excellent signal to noise ratio of the direct communication from FPGA to X310. These results are consistent with results published in the literature using similar processing techniques (Kawamura et al., 2017; Yasuda et al., 2019).

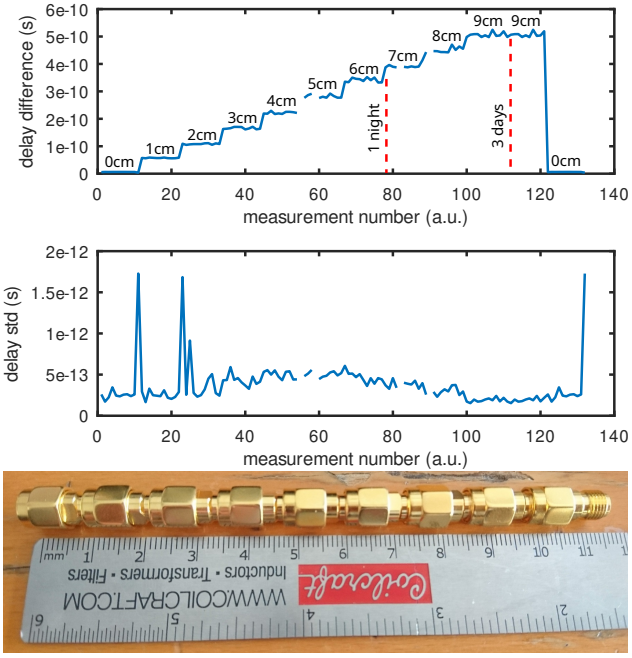

Figure 6. Time delay difference between the two channels of the X310 fitted with BasicRX frontends (top) as additional SMA-SMA extensions are added to one path with other remaining with a fixed length. Bottom: picture of the full set of SMA extensions next to a ruler graduated in cm (top) and inches (bottom).

Similarly for the B210, both inputs are fed the same PRN sequence and the same analysis is performed, this time with the full scale range reached by tuning the AD9361 frontend gains (Fig.7). Notice the Y-axis of Fig.7 which only spans 140 ps or 1/1000th of the span of the charts in Fig.4. In this case, the performance is degraded with a short term standard deviation remaining in the 42 ps range with peak to peak variations during the set of observations of 105 ps. The degradation by a factor of 10 of the relative delay standard deviation with respect to the X310 is attributed to the complex AD9361 frontend whose many processing steps might induce randomly fluctuating time delays well below the sampling period.

2.3. Impact of the SDR clock source

All measurements are performed with a common clock source driving the FPGA and the SDR receiver. When clocking the SDR with its internal frequency reference (Fig.8), each channel is observed to drift over time with respect to the FPGA signal: indeed, the frequency difference between the clocks driving the FPGA and the SDR receiver is integrated as a linear phase drift also observed

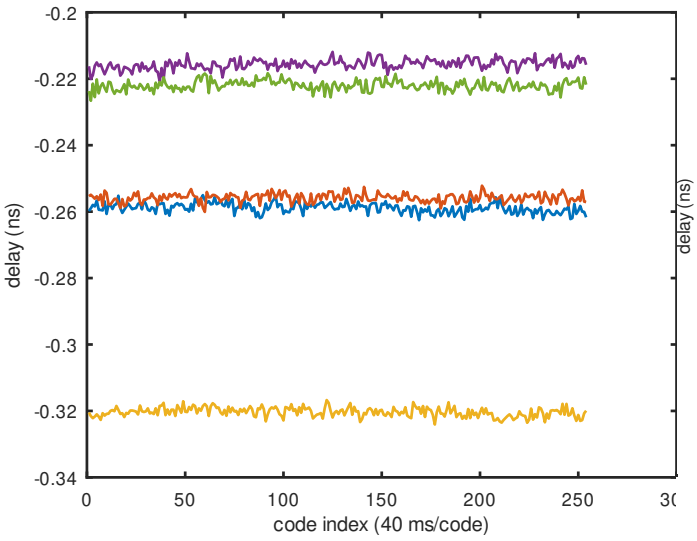

Figure 7. Time delay difference between the two channels of the B210 when the measurement is repeated 5 times.

as a linearly increasing time delay.

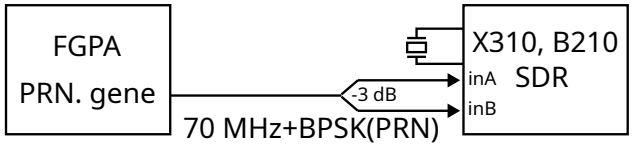

Figure 8. Experimental setup: rather than distributing the same clock reference to all peripherals, the 10 MHz reference clock drives the FPGA but the X310 SDR is running on its internal clock.

However, since we have concluded that only differential analysis between reference and measurement channels are relevant, we observe that by subtracting one delay drift with respect to the other, the common clock offset is rejected and performances similar to those observed when clocking the SDR with the external frequency signal are recovered. Indeed, the standard deviation within each acquisition is 0.7 ps and the maximum excursion is 1.7 ps (Fig.9).

2.4. Synchronizing multiple receivers

Having concluded that each individual SDR receiver exhibits reproducible delays between both channels in the few picosecond range, even when clocked with an internal source whose frequency is different than that of the PRN generating FPGA, we address the case of synchronizing multiple SDR frontends. In all cases the common external frequency source will be used to clock all peripherals.

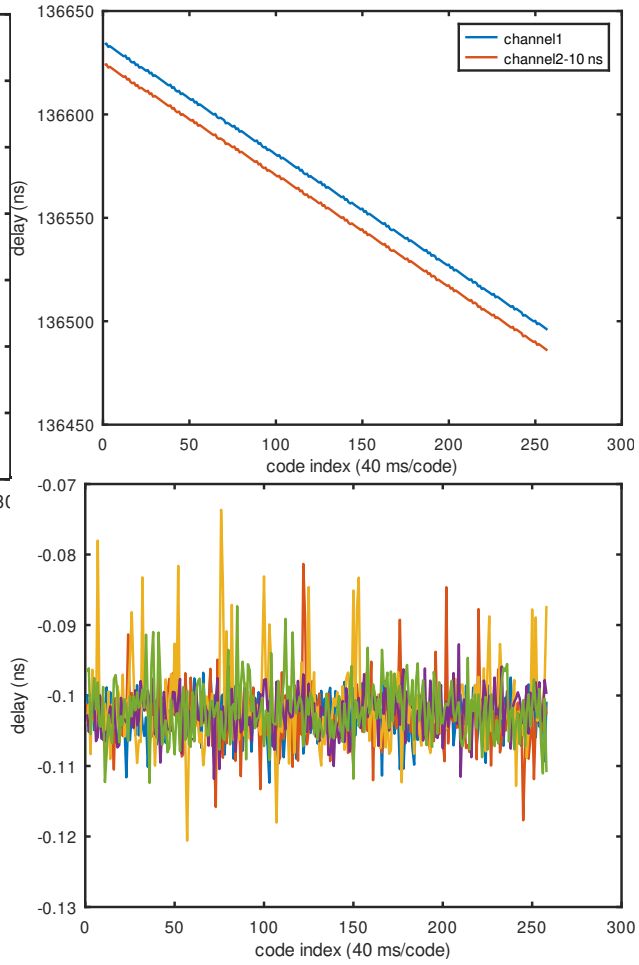

Figure 9. Top: each channel of the X310 is observed to drift over time since the SDR clock is not the same than that of the FPGA generating the PRN sequence. The red curve has been offset with respect to the blue curve by 10 ns for clarity. Bottom: subtracting both channel delays cancels the common mode clock offset.

Two dual-channel X310s fitted with BasicRX frontends are fed the same PRN and the time delay is computed by cross-correlating the known PRN sequence with the collected IQ stream. Fig.10 not only demonstrates how the result is reproducible from one X310 to another, but also that comparing one channel of one X310 with one channel of the other X310 allows for synchronizing multiple SDR receivers with sub-sampling period accuracy. On Fig.10, 1000 measurements lasting each 10 s have been overlaid, with the X-axis exhibiting the evolution within each acquisition of the delay and the line thickness the reproducibility from one acquisition to another. The thicker the line, the poorer the reproducibility: while all acquisitions from one X310, the other X310 or between both X310s exhibit 0.5 ps standard deviation within the 10-s sampling dura-

tion, the standard deviation from one run to another rises to 1.6 ps when comparing two different SDR channels. The excellent performance of the two X310 synchronization is attributed to the common synchronization mechanism of both platforms leading to reproducible behaviors.

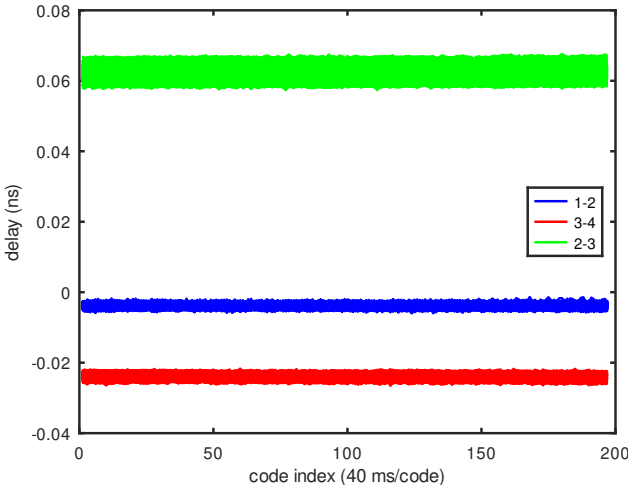

Figure 10. Synchronizing both channels of one X310 (blue) referred to as channels 1 and 2, or another X310 (red) referred to as channels 3 and 4, and comparison of the synchronization of channels 1 and 3 (green) on different SDR receivers. Although the standard deviation is degraded, it remains well below the sampling period.

This situation is degraded when attempting to repeat the experiment with one X310 and one B210 since the latter exhibits degraded standard deviation with respect to the former, but most significantly both platforms rely on difference sampling techniques (direct ADC v.s AD9361) and different communication interfaces (Ethernet v.s USB) which might induce different trigger mechanisms (Fig.11). Indeed while the X310 and B210 each exhibit delay fluctuations between their respective channels consistent with previous analysis – 0.5 ps and 1.5 ps mean standard deviation within each trace, and 0.6 ps and 38 ps standard deviation from one measurement to another – the time delay between one channel of the X310 with respect to one channel of the B210 is randomly distributed within the sampling period.

2.5. LMS7002 frontend on the XTRX

We attribute the excellent performance of the X310 to the direct sampling by the ADC with no preliminary processing that might introduce random delays, and the degraded performance of the B210 with its AD9361 frontend to uncontrolled delay fluctuations. Another widely available radiofrequency frontend is the Lime Microsystems fitted on multiple popular SDR receivers including the Fairwaves

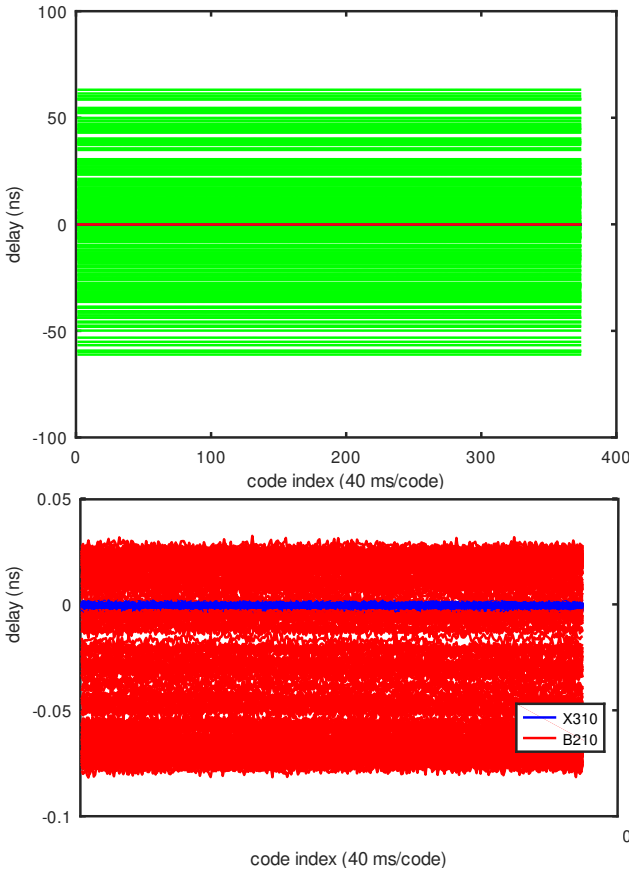

Figure 11. Top: overlap of 1000-measurements each lasting 10 s illustrating the synchronization of both channels of a X310 (blue) or a B210 (red), and comparison of the synchronization of one channel of the X310 and one channel of the B210 (green). While both channels of each SDR receiver exhibit consistent delays, the comparison of the B210 to the X310 is randomly distributed within the sampling period. Bottom: zoom on the region close to 0-delay emphasizing the consistency of the delay between the two channels of the B210 (red) and the two channels of the X310 (blue) despite the lack of synchronization between the two SDR receivers.

XTRX. Being a dual channel input frontend fed with external 10 MHz clock and 1-PPS, the same analysis is performed using the GNU Radio flowchart shown on Fig. 12. Similar to past observations, the absolute delay is randomly distributed within the sampling period (data not shown) and only the differential analysis will be developed.

The PRN generated by the FPGA is fed on the one hand to two channels of the X310 receiver as reference, and to the two channels of the XTRX receiver fitted on its PCI carrier board. Similar to the synchronization of the X310 with the B210, the delay between XTRX and X310 channels is observed to be randomly distributed within a sampling period. However, while the X310 exhibits the expected standard

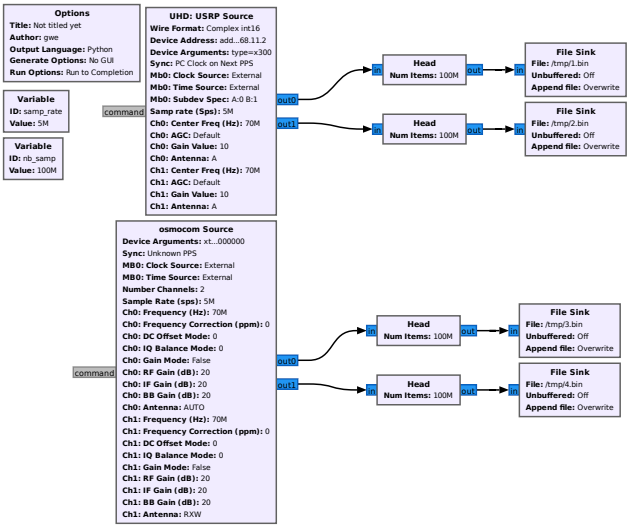

Figure 12. GNU Radio Companion flowchart collecting samples from an Ettus Research X310 SDR receiver and a Fairwaves XTRX, saving IQ samples to files for post-processing.

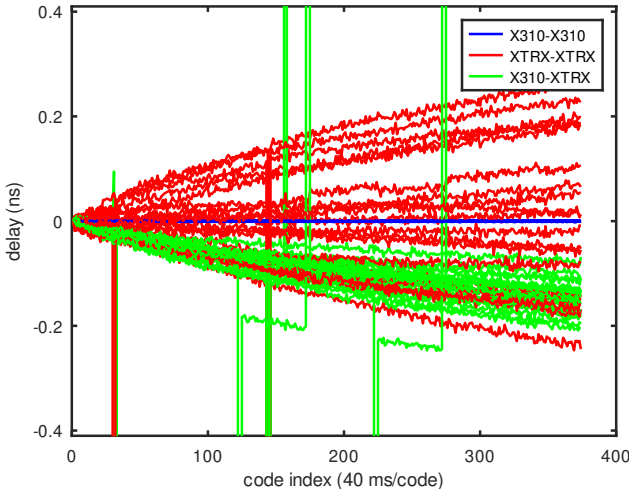

Figure 13. Evolution within each 15-s long acquisition of the delay between two XTRX channels (red), two X310 channels (blue) and one XTRX v.s one X310 channel (green) after removing the random initial offset within the sampling period.

deviation of a fraction of a picosecond within each trace and from one measurement to another, the XTRX exhibits 375 ps standard deviation over the 15 s acquisition duration. The analysis of the short-term evolution of the delay of one XTRX channel with respect to the other (Fig. 13) demonstrates that this high standard deviation value is not due to a random fluctuation of the phase (delay) but a slow drift during the acquisition. This drift is not reproducible from one measurement to another and might even change sign.

3. Conclusion

We have investigated the timing capability with sub-sampling period resolution of various consumer-grade software defined radio receivers including the Ettus Research X310 and B210 and the Fairwares XTRX. We conclude that sub-picosecond synchronization can be achieved with the direct sampling X310 fitted with the BasicRX frontend under the assumption that one channel is dedicated to synchronization and the other to measuring a signal of interest. The performance is degraded with the B210 to a few tens of picoseconds, probably limited by random fluctuations within the complex AD9361 frontend from one measurement to another. Finally, the XTRX fitted with the LMS7002 exhibits a slow drift of a few hundreds of picoseconds over the few second measurement duration, leading to a strong performance degradation with respect to the other two platforms.

Most importantly, we have shared the measurement procedure which can be adapted to a wide range of hardware for generating the pseudo-random number sequence generator for the reproduction and extension of these measurements to more dual-channel receiver platforms. The github archive of the necessary tool is available at `https://github.com/oscimp/amaranth_twstft` whose output is designed more generally for time and frequency dissemination using software defined radio hardware.

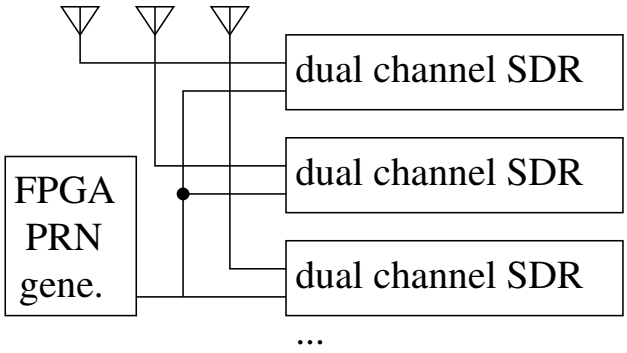

Figure 14. Envisioned architecture for analyzing with sub-sampling period the signals collected in multiple frequency bands beyond the bandwidth of each SDR receiver.

The final envisioned architecture for analyzing the fine time delay between signals widely spread over the radiofrequency spectrum, e.g. broadcast by Global Navigation Satellite System constellations, is illustrated in Fig. 14, in

which the radiofrequency signal of interest is shifted using an external mixer from the radiofrequency band to the intermediate frequency carrying the pseudo-random sequence as binary or quad-phase shift keying modulation, and the radiofrequency frontend samples both channels under the exact same conditions. Then the reference channel used for estimating the time delay between receivers is processed on the one hand, the payload of the measurement channel is extracted, and the former information is used for timestamping with sub-sampling period the events of the latter channel.

References

Friedt, J-M, Droit, C, Martin, G, and Ballandras, S. A wireless interrogation system exploiting narrowband acoustic resonator for remote physical quantity measurement. *Review of scientific instruments*, 81(1), 2010.

Friedt, Jean-Michel, Eustache, Clément, Carry, Émile, and Rubiola, Enrico. Software-defined radio decoding of dcf77: Time and frequency dissemination with a sound card. *Radio Science*, 53(1):48–61, 2018.

Kawamura, Seiji, Ohta, Hiroki, Hanado, Hiroshi, Yamamoto, MK, Shiga, Nobuyasu, Kido, Kouta, Yasuda, Satoshi, Goto, T, Ichikawa, Ryuuichi, Amagai, Jun, et al. Water vapor estimation using digital terrestrial broadcasting waves. *Radio Science*, 52(3):367–377, 2017.

Yasuda, S, Ichikawa, R, Hanado, Y, Kawamura, S, Hanado, H, Iwai, H, Namba, K, Okamoto, Y, Fukunaga, K, Iguchi, T, et al. Horizontal atmospheric delay measurement using wireless two-way interferometry (wiwi). *Radio Science*, 54(7):572–579, 2019.