A 400 Msps SDR platform for prototyping accurate wideband ranging techniques

Cherif Diouf [1], Gerard J.M. Janssen [2], Tarik Kazaz [2], Han Dun [1], Farnaz Chamanzadeh [2], Christiaan C. J.M. Tiberius [1]

[1] *Geoscience and Remote Sensing, TU Delft, Delft, The Netherlands*
[2] *Circuits and Systems, TU Delft, Delft, The Netherlands*

***Abstract*—In order to validate and demonstrate newly developed ranging techniques, a flexible test platform for signal acquisition enabling offline signal processing is generally needed. Developing such a platform becomes challenging when working with wideband (> 100MHz) signals due to the critical timing, the very high sampling rates and the huge data throughput involved. In this paper, we introduce an Ettus X310 SDR platform using custom designed logic allowing for dual-channel 400 Msps data transmission and acquisition for centimeter level ranging applications. Furthermore, we present initial measurement results as a benchmark of the platform, which show that the time delay of a 10 m cable can be estimated with high accuracy, in the order of 50 ps.**

Keywords—Wideband communication and ranging, USRP, low duty cycle transmissions.

I. INTRODUCTION[1]

Software Defined Radio (SDR) systems are flexible programmable hardware devices capable of prototyping very diverse communication technologies. They are available from low complexity/low cost to very advanced and expensive devices. SDRs are hence used in various fields such as education, system development and for prototyping communication devices for applications such as radar, 5G or XG systems. A same SDR system can be programmed as, among others, a Wi-Fi Hub, a Bluetooth/Zigbee terminal [1], a GPS receiver or to implement more specific communication features. By using a software interface on a Host-PC, the SDR device is easily reconfigured to meet new development, prototyping or testing needs. Due to such flexibility, SDR based systems are very attractive solutions as platforms to evaluate novel signal processing and communication algorithms. The systems contain a DSP unit for signal processing operations and separate reconfigurable RF front ends and ADC/DAC units for transmission and reception. More advanced systems can also embed FPGA allowing the customer to develop on-device signal processing features.

One area of application of SDR systems is the validation and demonstration of novel ranging and positioning concepts. Several examples can be found in the literature. In [2], a navigation system using LTE signals is presented and validated using an SDR system. In [3], implementation of a real time SDR GPS receiver is presented. Multi-channel based ranging techniques for wide area networks are demonstrated in [4]. An SDR based TDoA indoor localization system using WiFi signals is presented in [5]. In [6], the performance of a spread spectrum technique for UHF RFID ranging is evaluated using an SDR testbed. The previous papers study ranging based on the narrowband signals where the signal bandwidth is not a challenge. The typical signal bandwidth is varying from hundreds of kHz (Sigfox, LoRa) to tens of MHz (Wi-Fi). As the ranging accuracy is inversely proportional to the signal bandwidth, the use of wideband signals allows for a better time resolution and enables more efficient mitigation of multipath effects. Achieving centimeter level accuracy therefore requires signals with a bandwidth in the range of 500 MHz or more. However considering such a large bandwidth using an SDR platform becomes constrained because of the high sampling rates, the accurate timing, the computational and data throughput requirements. Hence, for such cases, on device signal processing implementation is often needed.

However, custom hardware implementation of DSP algorithms, on a FPGA, is in general very time consuming or even inopportune, particularly in the early stages of new developments when the techniques are still immature. In such a case, it is more appropriate to acquire and store the relevant sampled data from prototyping experiments on a Host-PC and perform offline data processing and analysis to assess the techniques.

In this paper, we focus on the transmission and acquisition of data samples at very high sampling rates (i.e. i.e 200 Msps or more) for precise ranging, where real-time data throughput to the host-PC and timing accuracy become very constraining factors. We use multiple Ettus X310 USRPs, Universal Software Radio Peripherala, as SDR platforms. They can transmit and receive at a maximum overall bandwidth of 320 MHz with 2 channels. The remainder of this paper is organized as follows. In Section II, the Ettus X310 SDR device is presented and the challenges of acquiring samples at such a high sample rate are discussed. The specifications and the required functionalities of the SDR platform are put forward in Section III. In Section IV, the custom hardware developments are discussed in more detail and initial ranging results are presented in Section V. In Section VI, conclusions and an outlook on future works are given.

II. THE ETTUS X310 SDR AND HIGH SAMPLING RATE CHALLENGES

A. *The X310 USRP system*

The Ettus X310 Universal Software Radio Peripheral (USRP) is an advanced, high performance SDR platform. It

---

[1] This research is supported by the Netherlands Organization for Scientific Research (NWO) in the project SuperGPS under Grant 13970.

978-1-7281-2082-9/19/$31.00 ©2019 IEEE

has two individually configurable RF channels, each of which can be configured as either transmitter or receiver, which can operate at a maximum sampling rate of 200 Msps. A sample is represented as a 32-bit word: a 16-bit in-phase sample value and a 16-bit quadrature-phase sample value.

Each RF channel has an effective bandwidth of 160 MHz. By combining two frontends a total bandwidth of 320 MHz can be covered, resulting in a total sample rate of 400 Msps. The central frequency of each RF frontend can be tuned from 10 MHz to 6 GHz. Several X310 units can be associated within a synchronized network to form a flexible array. The Ettus X310 USRP is compatible with the GNU Radio / RFNoC software environment, which allows to use, develop and associate specific Host-PC based DSP blocks in order to build custom SDR systems. Moreover, the X310 USRP features an embedded Xilinx Kintex7 XC7K410T FPGA which operates around 200 MHz. This allows for implementation of hardware based DSP techniques and control functions. The main advantage is that an FPGA implementation allows for a much faster signal processing than the GNU Radio software based DSP blocks.

B. *High sample rate data acquisition challenges*

In an FPGA based near real-time online processing mode, the computational burden is mainly carried out by the FPGA. The signal processing algorithms are directly implemented on the FPGA. In an offline processing mode, data samples are acquired and transferred to the Host-PC for processing and thus the computational burden is on the Host-PC. While online processing is needed for real-time applications, there are several cases where offline processing of the recorded samples is more convenient. For example when prototyping or testing new ranging systems, the algorithms may be novel and their implementation in hardware usually difficult and time consuming. Therefore, hardware implementation of algorithms for ranging in the early stages of their design and evaluation is not favourable and needed.

On the other hand, when sampling at a high rate, the transfer of data to a Host-PC, for offline processing, can be difficult to sustain. For instance, consider transferring samples from an X310 working at maximum sample rate of 400 Msps. With 32 bits/sample this results in a data throughput of 1.6 GB/s or 12.8 Gb/s. This throughput is above the data-rate of a single high-end optical 10Gbe Ethernet interface which tops at 10 Gb/s. Even when the X310 uses a dual Ethernet interface to support a theoretical offloading speed of 20 Gb/s, such a rate is well above the writing speed of the fastest SSD drives. Currently, SSD drives can only sustain a writing speed of 3.5 Gb/s. Moreover, even if a 12.8 Gb/s bitrate would be sustainable, recording one minute of samples will generate a data file size of 96 GB, which will limit the actual duration of the experiment. On top of that, we should add that the Operating System (OS) introduces time latencies and delays that will reduce the throughput in a less predictable way. Hence for such high sample rate SDR systems, online signal processing is generally used. We will now introduce the X310 SDR based, high sample rate, signal transmission and acquisition platform for prototyping, evaluating and testing centimeter level ranging techniques through offline data processing.

III. SYSTEM SPECIFICATIONS

In this section, the system specifications is presented. The focus is on signal requirements, transfer and accurate timing.

A. *Signal requirements*

The key consideration here is that we do not need a continuous transmission of the ranging signal in order to compute range estimates. Even if the target is moving , like a car for instance, a periodic low duty cycle wideband signal can be used to perform ranging or positioning.

Fig. 1 shows such a low duty cycle ranging setup, where a first X310 acts as a transmitter and a second X310 acts as a receiver. Let τ be the transmission duration and T the transmission period, then the duty cycle d is $d = \tau / T$ results. A period of $T = 1\ ms$ has been specified, which allows the computation of 1000 range estimates per second. Even a road vehicle running at 30 miles per second will move by only 3 cm in 1 ms. For a duty-cycle of $d = 0.1$, the transmission duration is $\tau = 100\ \mu s$. With these values, around 20000 samples per channel are transmitted in a single burst (radio core is running at 200 MHz).

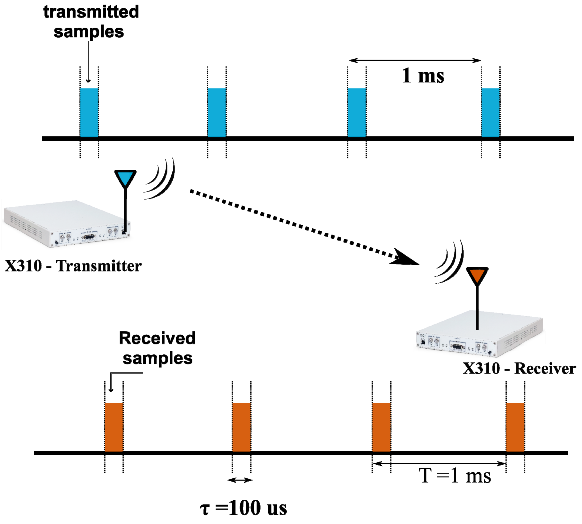


Fig. 1. Low duty cycle ranging signal transmission and reception, the *100 μs* is refered to as burst.

With two channels at full sampling speed, the throughput is now reduced to 1.28 Gb/s, a value well below the maximum optical Ethernet datalink capacity and the SSD's writing speed. Moreover, one minute of data recording will now result in a 9.6 GB file per second. When a lower ranging repetition rate is affordable the file size will be proportionally reduced. The ranging platform will be designed to support a RAW data format. For experiments, the signal samples are provided in this format to the X310 based transmitter, and the received samples are retrieved from the X310 based receiver. The sole requirement is that the signal should meet the X310 bandwidth limits. The transmitted signal can have a maximum bandwidth of 320 MHz when two channels of 160 MHz are combined.

B. *Device synchronization*

Synchronization is a critical issue in the presented ranging platform. As shown in Fig. 1, signals are transmitted and received periodically in bursts. Hence to correctly acquire these bursts, the transmitter and the receiver need to be synchronized. In addition, for positioning, multiple transmitters need to be very accurately synchronized. To ensure a robust synchronization of the transmitters, they are all connected to a 10 MHz reference clock and synchronized to a Pulse-Per-Second (PPS) signal using an external time reference distributor. In this way, several X310 devices will start streaming samples at the same instants. Such distributors are largely available on the market. One could also benefit from the Ettus software framework that allows to control several X310 devices in a MIMO/MISO configuration through the Ethernet network [7]. For the receiver the synchronization can be less strict in case Time-Difference-of-Arrival (TDoA) is applied, and can rely on a coarse detection technique as long as the burst is received. Otherwise it has to be synchronized to the reference time and frequency signals as well.

In the SuperGPS project, the developed platform is aimed at being part of a more comprehensive terrestrial positioning demonstrator targeting centimeter level accuracy. Instead of using a time reference distributor, a 10 ps atomic clock reference, distributed through an optical network, will be used. This time and frequency reference is then used to synchronize the timing as well as to syntonize the RF signal frequency of all the X310 transmitters.

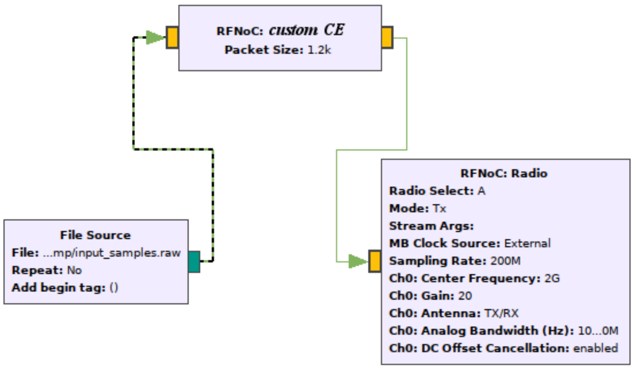
IV. SYSTEM DEVELOPMENT

In the GNU Radio – RFNoC environment, signal processing and control blocks or units can be developed by software using C++ or python. Those units can perform diverse DSP functions such as filtering, FFT, noise cancelation and correlation, which are running on the Host-PC. In addition it is also possible to develop hardware signal processing blocks, the Computational Engines (CEs), which are running, on device, on the Kintex7 FPGA. These CEs can implement custom DSP functions and provide samples to the radio frontend(s). The RFNoC framework [8] allows the implementation of up to 10 hardware CEs in the X310 FPGA. A complete radio system can be then set up by associating software or hardware DSP units in a coherent flowchart using the graphical user interface (GUI) of GNU Radio. This flowchart is then used to program the X310 SDR in order to perform the desired functionality.

In order to meet the specifications of the ranging platform as defined in section 3, two custom hardware CEs were developed, one for the X310 based transmitter to periodically transmit the ranging signal and one for the X310 based receiver to periodically sample the received ranging signal and off-load the samples to the Host-PC. Next, we detail the implementations for the X310 based ranging platform.

A. *Custom hardware development for the transmitter*

For the X310 based transmitter, the hardware unit (CE) is the main block that generates a local 1 ms reference synchronized to the external PPS time signal. It also stores the data to be transmitted and drives their periodic over-the-air streaming. The CE consists of an internal buffer of 128 kB, implemented in Block-RAM (BRAM) and finite state machines enabling the synchronized transmission of the low duty cycle signal bursts (Fig. 1). Two transfer mechanisms have been implemented. The first transfer of samples is from the Host-PC to the FPGA BRAM buffer. At the end of the transfer process the ranging samples have been loaded in the FPGA and are ready to be streamed out. Then from that memory, through the 200 Msps DACs, these samples are periodically transmitted over-the-air by the RF front-end and the connected antenna. Hence at start-up, the transmitter first loads the signal samples from the Host-PC to the X310 Kintex7 FPGA. Then the samples are periodically streamed out, through the X310 RF frontend once every 1 ms. Once the streaming process has been started, the Host-PC can be disconnect from the X310 device. An example of a GNU Radio flowchart for programming the X310 for this task is presented in Fig. 2. It consists of a source file where the samples to be transmitted are located, the developed custom CE and the RFNoC Radio block which encapsulates both the DACs and the RF frontend.



RFNoC: *custom CE*
Packet Size: 1.2k

File Source
File: ...mp/input_samples.raw
Repeat: No
Add begin tag: ()

RFNoC: Radio
Radio Select: A
Mode: Tx
Stream Args:
MB Clock Source: External
Sampling Rate: 200M
Ch0: Center Frequency: 2G
Ch0: Gain: 20
Ch0: Antenna: TX/RX
Ch0: Analog Bandwidth (Hz): 10...0M
Ch0: DC Offset Cancellation: enabled

Fig. 2. Example of a GNU Radio flowchart allowing to program the X310 transmitter.

B. *Custom hardware development for the receiver*

For the X310 receiver, a custom hardware unit (CE) has also been developed. It records the incoming samples and drives their periodical offloading to the Host-PC. The block also uses a (BRAM) buffer of 128 kB and state machines. Contrary to the transmitter, here the CE unit is continuously waiting for new samples. Once a new burst of samples is available from the front-end through the 200 Msps ADCs, the data is stored in BRAM until it is full. The BRAM is by default tailored to record data for a duration of $\tau = 100\ \mu s$. Then a window of 900 µs is available to offload the received samples to the Host-PC. By using this low duty cycle approach (Fig. 1), high sample rate data transfer is supported. After offloading the data, the CE block is waiting for the arrival of the next signal burst to restart the save and offload cycle. For each recorded packet, the FPGA is adding a header, in front of the samples, to facilitate the separation between different received packets. At the end of the experiment, the data (headers and samples) are available in a RAW data file on the Host-PC and can be exploited for offline processing.

V. EXPERIMENTAL VALIDATION

In this section, some initial results obtained with the developed test platform are presented. In this setup, as shown in Fig. 3, we will estimate the time delay of the connecting RF cable between an X310 transmitter and an X310 receiver. Synchronization between the two X310 devices is done by using the same 10 MHz and 1 PPS clock distributor. When synchronization is not performed by an external clock source, a samples burst detector based on cross correlation techniques can be implemented. Such a scheme is less ideal because its accuracy depends on the SNR. Hence for the actual experiment we only consider the ideal synchronization scheme which relies on a clock distributor device.

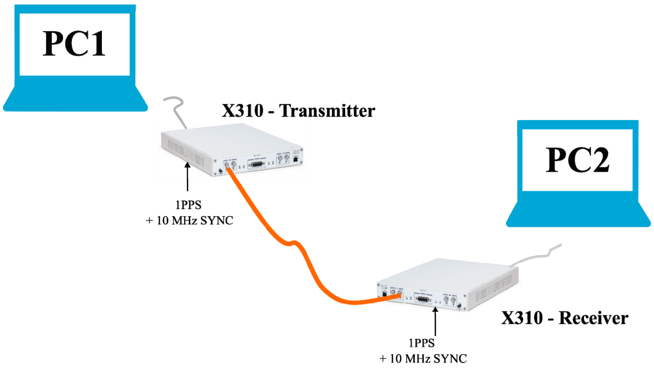


Fig. 3. Experimental setup : transmitter and receiver connected by cable.

The main objective is to estimate the propagation time delay based on the average Time-of-Arrival $\overline{\tau}$ of transmitted samples/packets, between the RF front-ends. Both X310 devices are linked to a Host-PC. A PN sequence is generated by Host-PC1, loaded to the X310 transmitter and periodically transmitted at 3.5 GHz (Fig. 1). The receiver periodically acquires samples of the incoming signals which are then recorded at the Host-PC2 hard drive. The transmitted and received digital signals are sampled at 200 Msps, resulting in a time resolution of 5 ns. In an offline processing mode, we then perform oversampling by a factor of 1000 to obtain a resolution of 5 ps.

Let *s[k]* be the reference signal obtained from the PN sequence and *r[k]* the received samples, then the propagation time $\hat{\tau}$ can be estimated using cross-correlation between *s[k]* and *r[k]:*

$$\hat{\tau} = T_s \left(\arg\max_{\tau}\left\{\left|\sum_{k=0}^{N-1} \underline{r}[k]\underline{s}^*[k-\tau]\right|\right\} - \mu\right) . \qquad \text{(eq. 1)}$$

Where N is the correlation size, T_s is the sampling period and μ is the hardware delay introduced by both the X310 transmitter and receiver devices. This delay should be constant and characterized beforehand (calibration). Along with the ToA estimation, we will also consider its standard deviation and also other parameters such as the throughput and the samples file-size at PC2.

A. RF cables propagation time measurements, one channel

For benchmarking we use the system to estimate the propagation time $\hat{\tau}_{5m}$ and $\hat{\tau}_{10m}$ of two RF cables, with a length of 5 m and 10 m length, respectively. The propagation times estimated are then compared to the measured propagation time, τ_{5m} and τ_{10m}, which was obtained by using a 20 GHz bandwidth Tektronix CSA803A oscilloscope with an SD-26 plugging and an S-52 pulse generator with < 25 ps rise time. The setup is presented in Fig. 3. The transmitter and the receiver, are linked through the RF cables (5 m then 10 m). From Host-PC1, a 2^{16} complex PN signal is fed to the transmitter which periodically streams the samples at 1 ms interval. The receiver periodically saves the samples to the Host-PC2 hard drive. Cross-correlation is then used to estimate the propagation times. One should note that we do not directly perform cross-correlation between the baseband generated PN sequence and the received samples. In an initial step, a reference signal, which can be considered as the overall system response to the PN sequence is extracted from the setup. In that calibration, a 1 m cable, with a known propagation time τ_{1m}, is used to link the transmitter and the receiver.

TABLE I. MEASURED PROPAGATION TIMES FOR DIFFERENT CABLES

τ_{1m} (ns)	τ_{5m} (ns)	τ_{10m} (ns)
4.801	20.238	40.356

The extracted calibration/reference signal incorporates the time delay due to the 1 m cable but also linear impairments of the RF transmitter and receiver frontends. The received samples from the 5 m and the 10 m experiments are correlated with this reference signal to obtain the relative propagation time named $\hat{\tau}_{5m\_r}$ and $\hat{\tau}_{10m\_r}$, respectively. The estimates are thus $\hat{\tau}_{5m} = \hat{\tau}_{5m\_r} + \tau_{1m}$ and $\hat{\tau}_{10m} = \hat{\tau}_{10m\_r} + \tau_{1m}$. With a sampling frequency of 200 MHz, the resolution is 5 ns. Up-sampling of a 1000 factor is performed offline, allowing for a time resolution of 5 ps which corresponds to a 1.5 mm distance resolution in over-the-air propagation. In Table. 1 we present the measured propagation times with the CSA803A oscilloscope.

TABLE II. PROPAGATION TIME ESTIMATION ON THE 5 M CABLE

Run		1	2	3	4
ToA	$\overline{\tau}_{5m}$ **(ns)**	20.200	20.200	20.135	20.200
	STD (ps)	0	0	0	0
$\overline{\tau}_{5m} - \tau_{5m}$ **(ps)**		38	38	103	38

In Table II, the estimated propagation time, average value, $\overline{\tau}_{5m}$, is shown for 4 runs. A run consists in turning off both transmitter and receiver and turning them on again in order to launch a new experiment. From the presented table, the estimates are very close to the measured values, there can be still a 38 ps bias which would correspond to a 1.14 cm distance (over-the-air) while an error of 103 ps (3 cm, over-the-air) has also been observed. For each run the average ToA and the standard deviation are computed using 500 packets of 1 ms (so, 0.5 second of signals is used to this end).

In Table III, the estimated propagation time, average value, $\overline{\tau}_{10m}$, is also shown for 4 runs with the 10 m cable The estimation is also close to the measured values, the maximum error is 56 ps error which would correspond to a 1.68 cm distance (over-the-air). Here we also can observe slightly different mean values for different runs.

TABLE III. PROPAGATION TIME ESTIMATION ON THE 10 M CABLE

Run		1	2	3	4
ToA	$\overline{\tau}_{10m}$ **(ns)**	40.325	40.350	40.300	40.300
	STD (ps)	0	0	0	0
$\overline{\tau}_{10m}-\tau_{10m}$ **(ps)**		31	6	56	56

For both cables the empirical standard deviation within one run and with respect to several 1 ms packets is zero. This is due to the very low noise level in the setup. The SNR is high and the received 1 ms packets, within a given run, are essentially alike. In this one channel configuration, a throughput of 99 MB/s has been observed, 10 s of experiment will thus generate a file size of around 1 GB.

B. RF cable TDoA using two channels

In the previous experiments, only one channel of each X310 device was used. We now move to a two channel configuration where both X310 channels, channel 0 and channel 1, respectively, are streaming samples at the transmitter. Also both RF frontends at the receiver are sampling the incoming signals. The central frequency is the same: 3.5 GHz and the PN sequence is common for both channels.

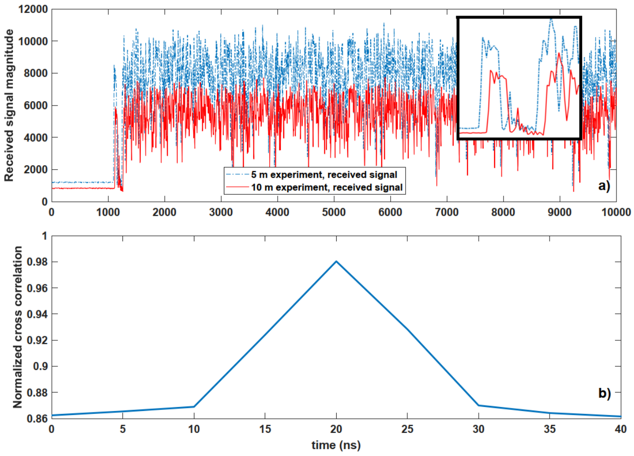


Fig. 4. a) Received samples magnitude from the two channels extracted from 1 ms packet, b) Normalized magnitude of the cross-correlation between the two received signals, using 20000 samples of one burst.

The main difference here is that we are using the 5 m cable to connect channel 0 of the transmitter and the receiver. While the 10 m cable is used to connect the channel 1 of the transmitter and the receiver. Our objective is to estimate the propagation time difference between the two cables. To that purpose we perform cross-correlation between the two received signals of the channel 0 and channel 1, which is shown in Fig. 4a.

A difference of amplitudes can be observed, the 10 m cable presenting high loss values. A difference of delays can also be observed the 5 m cable having lower propagation time. The magnitude of the cross-correlation function is plotted in Fig. 4b. The cross-correlation is performed without using up-sampling. We can observe that the correlation peak is located at around $\Delta\hat{\tau}=\hat{\tau}_{10m}-\hat{\tau}_{5m}=20\,ns$ which is close to the time difference $\Delta\tau=\tau_{10m}-\tau_{5m}=20.118\,ns$.

We now use oversampling by a 1000 factor to obtain more accurate TDoA estimates average. The results of 4 runs are shown. In that case we can obtain an error lowered to 12 ps which should correspond to an over-the-air distance of 3.6 mm.

TABLE IV. PROPAGATION TIME DIFFRENCE OF THE TWO CABLES

Run		1	2	3	4
TDoA	$\overline{\tau}_{10m}-\overline{\tau}_{5m}$ **(ns)**	20.115	20.130	20.130	20.130
	STD (ps)	0	0	0	0
$\Delta\overline{\tau}-\Delta\overline{\tau}_{m}$ **(ps)**		3	12	12	12

VI. CONCLUSIONS

This paper presents an SDR test platform for ranging and positioning that allows transmitting and acquiring a dual channel signal at 400 Msps for further offline processing. The platform is developed using Ettus X310 USRPs operating with a dual 160 MHz bandwidth and a sample rate of 200 Msps. Due to the high data throughput generated under such conditions, the use of low duty cycle signals is proposed. After introducing the technical specifications of the X310, we put forward the limits implied by acquiring signals at high sampling rates. Ethernet link throughput, SSD disks writing speed, data file size and OS time latencies impede data acquisition sustainability. We have introduced platform requirements to overcome these limitations, the developments carried out and how to use the platform in a GNU Radio environment. Furthermore, a strict timing synchronization between different USRPs was implemented based on a 10 MHz and 1 PPS reference clock. Experimental validation was carried out by using the system to measure RF cable delay times through ToA estimates. It was shown that the time delay of a 10 m cable could be estimated with high accuracy, in the order of 50 ps. Future works includes using the platform for ranging, positioning and channel sounding experiments.

REFERENCES

[1] Kazaz, T., Jiao, X., Kulin, M. and Moerman, I., "WiSCoP-Wireless Sensor Communication Prototyping Platform," EWSN'17 - International Conference on Embedded Wireless Systems and Networks, Uppsala, Sweden - February 20-22, 2017.

[2] K. Shamaei, J. Khalife and Z. M. Kassas, "Exploiting LTE Signals for Navigation: Theory to Implementation," in IEEE Transactions on Wireless Communications, vol. 17, no. 4, April 2018, pp. 2173-2189.

[3] E. Schmidt, D. Akopian and D. J. Pack, "Development of a Real-Time Software-Defined GPS Receiver in a LabVIEW-Based Instrumentation Environment," in IEEE Transactions on Instrumentation and Measurement, vol. 67, no. 9, Sept. 2018, pp. 2082-2096.

[4] F. Wolf, J. Doré, X. Popon, S. d. Rivaz, F. Dehmas and J. Cances, "Coherent Multi-Channel Ranging for Narrowband LPWAN: Simulation and Experimentation Results," 2018 15th Workshop on Positioning, Navigation and Communications (WPNC), Bremen, 2018, pp. 1-6.

[5] T. Xie, C. Zhang and Z. Wang, "Wi-Fi TDoA indoor localization system based on SDR platform," 2017 IEEE International Symposium on Consumer Electronics (ISCE), Kuala Lumpur, 2017, pp. 82-83.

[6] F. Galler, S. Grebien, T. Faseth, K. Witrisal, G. Magerl and H. Arthaber, "Extension of an SDR UHF RFID Testbed for MIMO and Monostatic Time of Flight Based Ranging," in IEEE Journal of Radio Frequency Identification, vol. 1, no. 1, March 2017, pp. 32-38.

[7] S. Malkowsky et al., "The World's First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation," in IEEE Access, vol. 5, 2017, pp. 9073-9088.

[8] M. Braun and J. Pendlum, "A flexible data processing framework for heterogeneous processing environments: RF Network-on-Chip™," *2017 International Conference on FPGA Reconfiguration for General-Purpose Computing (FPGA4GPC)*, Hamburg, 2017, pp. 1-6.