Pathway to Coherent Phase Acquisition in Multi-channel USRP SDRs for Direction of Arrival Estimation

Alda Xhafa, Fran Fabra, José A. López-Salcedo, Gonzalo Seco-Granados
Dept. of Telecommunication and Systems Engineering
Universitat Autònoma de Barcelona (UAB)
Barcelona, Spain

Ivan Lapin
Radio Frequency Systems Division
ESTEC, European Space Agency
Noordwijk, The Netherlands

***Abstract*—This paper sheds lights onto the challenging problem of achieving coherent phase reception between the RF channels of existing multi-channel software-defined radios (SDR). This is a key problem that must be solved, for instance, in applications dealing with 5G positioning and integrated sensing and communications (ISAC) when direction of arrival estimation is needed. Despite being a fundamental problem, it is far from being straightforward to implement in practice with existing SDR devices. In this regard, the present paper provides a set of golden guidelines in order to succeed in this endeavor, and thus to leverage the full potential that multi-channel SDR devices have to offer. Apart from phase coherence, practical recommendations are provided as well to achieve the highest possible data throughput between the SDR and the controlling computer. Finally, a comparison among the most relevant features of different models of Universal Software Radio Peripheral (USRP) SDR devices is discussed, including an analysis of their performance in terms of phase differences between RF channels.**

***Index Terms*—USRP, SDR, Coherent Phase, Synchronization, Direction-of-Arrival, 5G.**

I. Introduction

Over the years, the advancement of technology and significant cost reductions have revolutionized the deployment of software-defined radio (SDR) devices, enabling them to become widely adopted and integrated into a myriad of applications. This has constituted a radical paradigm shift with respect to traditional radio systems, which required specialized and dedicated hardware components [1], and led to expensive and inflexible solutions. However, with the advent of SDR technology, the landscape changed dramatically. By utilizing software-based signal processing techniques, SDR devices gained the ability to adapt and reconfigure their functionality through software updates, eliminating the need for costly hardware modifications.

Gradual improvements in SDR technology led as well to the provision of SDR devices with multiple transmit/receive RF channels, thus paving the way for the implementation of multiple-input multiple-output (MIMO) communications. In the positioning arena, the availability of multiple RF receive channels allows for angular measurements, which can be used standalone or together with range and/or Doppler measurements in order to improve the user's position estimation accuracy.

Positioning, and actually, angular measurements, are expected to play a major role in next-generation cellular networks, which have experienced a revolutionary progress in the past decades, opening the doors to countless of new services and applications. At the present moment, for instance, downlink angle of departure (AOD) and uplink angle of arrival (AOA) are part of the positioning methods specified for 5G New Radio (NR). This enables new positioning paradigms such as single-BS positioning, where measurements from only one BS are needed to determine the user's position [2], [3].

As a result, determining the angle of departure/arrival of a radio frequency (RF) signals will soon become a vital component for numerous localization services, such as driving assistance and indoor positioning, along with other practical applications like channel estimation, beamforming, radar and sonar tracking. Unfortunately, angular estimation accuracy is extremely sensitive to the hardware impairments experienced by the multiple RF chains of the receiving SDR device that is in charge of processing the received signal [4], [5]. In particular, to the clock stability, synchronization and phase misalignment between channels. Manufacturing inaccuracies and temperature variations further aggravate these problems, which come on top of the problems already experienced due to the radio environment (i.e. multipath and interference), and the time-frequency misalignment between transmitter and receiver.

In recent times, many multi-channel SDR platforms have appeared in the market, thus bringing the SDR flexiblity to the spatial/angular domain. Some of these platforms are, for instance:

- KrakenSDR, previously known as Kerberos SDR, a platform composed of five RTL-SDR single-input SDR devices [6], which are coherently combined by means of an integrated calibration system [7];
- BladeRF 2.0 micro, an SDR board providing two transmit and two receive channels [8];
- LimeSDR, an open-source SDR board proving up to four receive channels [9];

This work was supported in part by the Spanish Agency of Research (AEI) under the Research and Development projects PID2020-118984GB-I00/AEI/10.13039/501100011033 and PDC2021-121362-I00/AEI/10.13039/501100011033; and mobility grant SEBAP (Societat Económica Barcelonesa d'Amics del País).

- Universal Software Radio Peripheral (USRP), a family of SDR platforms originally developed by Ettus Research and now part of National Instruments. Some of its multi-channel models are the E310, X300 or N310, providing from two up to four receive channels [10].

Among these SDR platforms, USRPs are widely spread and commonly adopted in many high-end developments due to the large FPGA that they embed, and the flexibility to meet RF requirements based on a common hardware platform. These USRP devices, with their flexibility and software support, offer an excellent environment for obtaining angular measurements using the signals simultaneously received a multiple RF channels. We will henceforth refer to this problem as angle or direction of arrival (DOA) estimation, and as it will be described later on, it poses many challenges from a practical implementation point of view. Given the aforementioned hardware-related impairments, maintaining phase coherence between RF channels of USRP-based systems is far from being trivial and not well-documented in the existing literature, even in the SDRs user manuals. While some contributions in the literature also face this problem [11], [12], [13], [14], they often provide very little practical details on how to overcome it, and thus newcomers to the field can barely ever succeed in solving such an underrated task.

The main goal of this paper is therefore to provide a set of golden guidelines in order to succeed in achieving coherent phase reception across multiple RF channels of an SDR device. The focus will be placed on USRP devices, and in particular, on model X300 equipped with two TwinRx daughterboards [15], model N310 [16] and model NI-2955 [17]. These are the USRP boards most widely adopted in practice for medium- to high-end applications requiring from two up to four receive channels working at a high sampling rate, and therefore, the guidelines to be provided here are of utmost interest to a wide range of researchers and practitioners.

To address these challenges, this paper presents a comprehensive methodology aimed at achieving phase coherence in USRP-based systems enabling DOA estimation. An analytical model of time, frequency, and phase error is discussed following by a comparative analysis highlighting the benefits and limitations of three SDR models. The main focus lies in addressing the issue of phase synchronization, a critical factor that significantly impacts the accuracy of the DOA estimation setup. Indeed, in work [18], it was demonstrated a significant improvement in DoA estimation accuracy, with a maximum enhancement of 46.74% achievable when considering the RF channel errors. All these results can be useful for researchers planning to implement DOA estimation, or, more in general, any set up where channel alignment is important.

II. USRP Main Configuration Features

In this section, the main features of the three USRPs under analysis are discussed and evaluated based on their hardware capabilities in terms of performance, environmental application requirements, maximum sustained data rate throughput (that depends as well on the communication interface), etc. Moreover, an analysis of the available software environment to program the USRPs, as well as the configuration process to enable the local oscillator (LO) sharing capability are covered. Before going into details, a summary of the main features of each considered USRP is presented next in Table I.

TABLE I
USRP Main Features Comparison

USRP Model	N310	X300-2TwinRx	NI-2955
Selected connectivity [Total throughput] <Per channel >	2 x 10 Gb Ethernet [1600 MB/s] <100 MS/s >	2 x 10 Gb Ethernet [1600 MB/s] <100 MS/s >	MXIe [800 MB/s] <50 MS/s >
Sampling rate options	Even or odd division from three master clock options (in MS/s): 153.6, 125 and 122.88	Even or odd division from a master clock of 200 MS/s	Any value between 0 and 100 MS/s
Preferred environment	Linux OS (Ubuntu) UHD functions GNU radio	Linux OS (Ubuntu) UHD functions	Windows OS LabVIEW
Other relevant aspects	The 4 RF inputs can be configured as transmitter channels	Modular architecture	Accurate factory calibration

A. *USRP Connectivity for Maximum Throughput*

The throughput limit of the communication link between the USRP and the controlling personal computer (PC) becomes one of the main bottlenecks preventing us to reach the maximum data rate. In this paper, two main alternatives are considered to overcome this problem. The first one is to employ a four lines MXIe 1st generation cable connecting the USRP to a PCIe card at the PC. This approach can achieve a sustained transmission rate of 800 MB/s, which corresponds to an incoming sample rate of 50 MS/s per each of the four considered channel, when dealing with I/Q data stream (interleaved I/Q samples) of 2 bytes (16 bits) per I/Q samples. The second alternative is to employ two 10 Gigabit Ethernet cables connected through dual SFP+ ports for establishing the connection between the USRP and the PC. This approach allows a combined transfer rate of 1600 MB/s corresponding to an incoming sample rate of 100 MS/s per channel under the same conditions. This option has a rate, which is double respect to the former one.

Two of the selected USRP under analysis, the NI-2955 and the X300-2TwinRx, allow the use of both connections. However, at the time of the experimentation, the solution to utilize the 10 Gigabit Ethernet on NI-2955 required burning the NI-2955's FPGA image into an Ettus equivalent version. This means virtually converting the device into a X310-2TwinRx equivalent and consequently losing all the dedicated features and drivers provided by NI. The N310, instead, has only dual SFP+ ports for a 10 Gigabit connection each. Therefore, for the reasons mentioned, the recommended connectivity for maximum data throughout is a MXIe cable for the NI-2955 and two dual 10 Gigabit Ethernet cables for either the X300-2TwinRx or N310.

Apart from using high throughput connections, high CPU clock rates, memory bandwidth and I/O bandwidth, some additional parameter configurations are needed to ensure getting most of the USRP host performance:

- A Linux low-latency kernel is needed if real-time computations with very low tolerance to delays are required to be performed.
- The CPU power management needs to be configured by disabling Hyper-Threading and thus increasing the

single-core performance at the cost of having less core threads. Moreover, other power management functions such as sleep states and CPU frequency scaling should be disabled as well.

- The CPU frequency should be set to the highest value within the range of its scaling limits and enable performance option of the CPU governor through the Linux utility *cpufrequtils*. Through these power schemes it is possible to determine the desired frequency for the CPU and achieving a maximum performance.
- The thread priority of the UHD software should be set to the highest scheduling priority.
- The Ethernet interface has its own sub-net, and the corresponding USRP device should be assigned an address in that sub-net. The maximum size of the socket buffers to avoid potential overflows and underruns at high sample rates should be increased. The size of the buffer depends on the SDR model's recommendations. The following commands can be used:

  sudo sysctl -w net.core.rmem_max=sizeSocketBuffer
  sudo sysctl -w net.core.wmem_max=sizeSocketBuffer
- For the case of Ethernet 10 Gb interface, prevent flow control errors at higher rates by increasing the Ring Buffers on the Network Interface Card (NIC) using the following command:

  sudo ethtool -G <interface> tx sizeRingBuffer rx sizeRingBuffer
- To achieve maximum performance, the Maximum Transmission Unit (MTU) should be set to 9000 for 10 GigE and 1500 for 1 GigE. It is possible to use smaller MTUs, but this can affect performance. Setting the MTU size is done using the following command:

  10 GigE: *sudo ifconfig <interface > mtu 9000*
  1 GigE: *sudo ifconfig < interface > mtu 1500*

Note that the above mentioned commands are compatible with Linux operating system. For other operating systems, these commands should be substituted with the corresponding version. The tuning process is needed only for the case of Ethernet interfaces while for the MXIe interface is sufficient fulfilling the installation steps.

B. USRP Clock Set-up

The USRP series generate a master clock on the motherboard, which is then used to drive the Analog to Digital Converter (ADCs), Digital to Analog Converter (DACs), and the radio blocks. The clock rate is referred to as the "master-clock-rate". There is always a single master-clock-rate per motherboard. This rate is also the base sample rate of the radio blocks. Knowing the master-clock-rate, the actual sampling rate available can be an integer divisor of the master-clock-rate. The USRPs under analysis sustain different master-clock-rates. In general the X300 series support a 200 MHz and a 184.32 MHz master-clock-rate, with 200 MHz being the default one. However, X300-2TwinRx only supports 200 MHz, since putting TwinRx locks the clock rate to 200 MHz. The N310 series support master-clock-rates: 122.88 MHz, 125 MHz, 153.6 MHz and NI-2955 supports a master-clock-rate of 200 MHz. Moreover, the X300-2TwinRx and NI-2955 are a 4-channel receiver while N310 is a 4-channel transmitter/receiver.

The transmission and reception of data depending on master-clock-rate, transmission rate, and reception rate is not always possible. In fact, when working at higher rates, a high number of dropped samples, overruns, transmit and receive sequence errors and underruns may occur. Another common error that can be faced is that "the receive packet handler failed to time-align packets". Even though several packets can be process by the handler, a timestamp match cannot be determined. Note that only strict integer decimation and interpolation are used within USRP hardware to achieve the desired sample rate where the ratio *master-clock-rate/desired-sample-rate* should be an even integer ratio. For the rest of sample rates where the ratio is odd, the performance of the USRP is not guaranteed, especially when phase coherent operation among channels is required. In this context, NI-2955 provides a re-sampling process that allows to use a wide range of sampling rates and, compared to the other USRPs under analysis, it gives a better flexibility for applications where the sampling rate of signal to be streamed is a key factor.

Additionally, the sampling rate is closely related with time alignment that depends also on phase alignment. The received signals at all front ends should be sampled at the same time instance. All USRP models include options for using an internal or external clock reference with the added ability to export the clock reference and time base. For the external part, they accept a 10 MHz reference from which the ADC/DAC clocks and local oscillator are derived. When the internal clock of the USRP itself is used, a frequency accuracy of 2.5ppm is obtained. A common reference clock that can be used also by the USRP is PPS, a standard pulse per second port or as a general-purpose digital trigger input line. Additionally, the USRP NI-2955 includes also a precision GPS-Disciplined OCXO (GPSDO), which provides improved frequency accuracy without using GPS and significantly improves frequency accuracy when disciplined to the GPS satellite network.

C. USRP Software Environment

USRP are controlled through the USRP Hardware Driver (UHD) support, which is an open-source software compatible with a large number of supported development frameworks, reference architectures, and open source projects. The UHD architecture provides all the functional blocks for digital down-conversion and up-conversion, fine-frequency tuning, and other Digital Signal Processing (DSP) functions allowing it to be interchangeable with other USRP devices. UHD is supported by multiple signal processing software, like GNU Radio, LabVIEW and Matlab. They support both operating systems: Linux and Windows. GNU Radio [19] is an open-source software package compatible with various SDR platforms. It provides solutions for developing wireless protocols at the PHYSICAL and DATA LINK layers of the protocol stack. The software is primarily written using Python, while performance signal processing blocks are implemented in C++. These blocks are basic operation units that process continuous data streams and can be connected together to form a flow-graph.

LabVIEW is also a flexible software platform that is compatible with USRP, especially with NI devices, where dedicated drivers and code examples and templates are available. In our analysis, NI-2955 is operated under this environment. In order to reach the maximum throughput, two important modifications are required to be applied to the default templates for data acquisition: (1) the block diagram must follow a producer/consumer architecture, where the block that gets data from the USRP and the block that writes this data into the output file are located in parallel working threads (the data transfer between them is managed by a queue); and (2) the data samples have to be kept in their original 16-bit format (short integer) instead of being converted into a 64-bit format (double), as it is done by default.

Every development tool selected can run independently from the other (e.g. MATLAB for signal modeling and GNU Radio for USRP handling) or they can interact with each other (e.g. LabVIEW controlling the USRP while making MATLAB calls for data processing). Matlab has also its own hardware support baggage, which is able to communicate with USRP. All the software drivers must be installed and configured in the host computer following the step by step installation guide. It is recommended using the latest stable version of UHD that is available.

III. USRP RF and Clock Calibration

DOA estimation is achieved by comparing the phase difference experienced by signals received at different RF ports using an antenna array. The accuracy of DOA estimation heavily relies on precise knowledge of the antenna response, making antenna calibration an essential step in the process. However, even with properly calibrated antennas, accurate DOA estimation cannot be guaranteed. To enhance the accuracy and reliability of DOA estimation systems, especially in scenarios where multiple antennas are utilized to determine signal arrival direction, RF calibration also plays a fundamental role. It involves compensating for hardware imperfections in the RF front-end components. This calibration process is necessary to ensure the best possible performance of the system. For this reason, in this paper we will focus specifically on achieving synchronization of channels in phase and time, as it significantly impacts DOA estimation accuracy. Different USRP models may require distinct processes, and we will explore these in subsequent subsections by providing a comprehensive guide on achieving phase alignment between channels.

A. N310 calibration

N310 has built-in calibration procedures that can be enabled from UHD. The applicable calibrations for N310 can be enabled/disabled at initialization time using the *tracking_cals* and *init_cals* device arguments, i.e. command keys recognized by the UHD drivers. The device can be set to the precise bit mask the chip uses to set those calibrations (e.g., init_cals=0x4DFF,tracking_cals=0xC3) or they can use the following descriptive keys provided by UHD (e.g.init_cals=DEFAULT, tracking_cals=TX_QEC|RX_QEC), a preset for enabling most calibration's configuration. The | symbol can be used to combine keys (equivalent to a bitwise OR). Calibrations can significantly delay the initialization of a session. However, by only picking relevant calibrations, sessions can be initialized faster. Because the two draughtboards of N310 are using different LO for references between the channels, the N310 itself has no way of aligning phase between channels, and phase will be random between runs. Therefore, to improve the phase alignment, it is recommended to apply an external LO source and splitting the signal source into the TX and RX, so that the phase ambiguity is reduced to ± 180 degrees. Note that the external LO frequency must be twice the center frequency, which is therefore limited to the range from 300 MHz to 4 GHz. The N310 itself has inputs for external local oscillators, where for every daughter-board, there is one input for TX and RX, respectively, resulting in 4 Local Oscillator (LO) inputs total per N310.

The phase alignment between channels through external LO, USRPs, is achieved by injecting a source signal from the external LO to the four TX/RX inputs of the N310 with a 4-way splitter. The data streaming to the N310 device is run. Stream commands with a time spec property are used, which instructs the streaming to begin at exactly the same time on all channels. Previously, a RF calibration [20] is performed by setting the necessary configurations define in the device arguments property *–args*, a parameter that the UHD utility application uses to take the device address together with calibration settings. The following *init_cals* setting parameters to be used for the initial calculation, are the ones that allow a minimal calibration:

- rx_lo_source=external,
- tx_lo_source=external,
- init_cals=BASIC|TX_ATTENUATION_DELAY|... RX_GAIN_DELAY|PATH_DELAY|... LOOPBACK_RX_LO_DELAY|RX_LO_DELAY|... RX_QEC_INIT|TX_QEC_INIT

Configuration parameters RX_QEC_INIT and TX_QEC_INIT require the external LO to be set to 5 GHz to perform the Quantum Error Correction (QEC) init calibration. Along with the above-mention parameters should be also set *force_reinit=1* and *tracking_cals=OFF*. The latter avoids the presence of frequencies that may distort the signal. The frequency value set is only necessary to perform calibration. After tuning the RF front-ends, the user can set the desired carrier frequency within the LO limited frequency range maintaining the same sampling rate and gain values with the calibration process and the same calibration settings as above without *force_reinit* argument. Each local oscillator may have a random phase offset due to the dividers in the VCO/PLL chains. This offset will remain constant after the device has been initialized, until the device is closed or retuned. Thus, phase coherency is not repeatable after re-tune or re-initialization of the device and the phases can suffer a 180° ambiguity. A phase re-calibration is required after these operations. Moreover, the quality of the onboard LO depends also on the external reference clock. By providing a custom LO signal, it is possible to increase accuracy assuming the externally generated LO signal is coming from a high-quality oscillator and will eventually improve the phase noise. In this work, no external clock is used.

B. X300-2TwinRx calibration

The TwinRX daughter-board for the USRP X Series SDR platform is a two-channel superheterodyne receiver that has the ability to share the LO between channels across multiple daughter-boards that enables the phase-aligned operation required to implement scalable multi-channel phased-arrays. The antenna ports are MMCX connectors with 50Ω input impedance. By default Antenna 1 (RX1) is routed to Channel 1 and Antenna 2 (RX2) to Channel 2. This routing can be changed swapping the antennas or sharing a single antenna between both channels. Each channel of TwinRx daughter-board has two local oscillators, LO1 and LO2. The local oscillators of a channel can be sourced from the channel's internal synthesizers, the companion channel's synthesizers, or external inputs. The LO of each channel can be sourced as *Internal, Companion* or *External*. Therefore, to activate the sharing ability between channels, the configuration of calibration of each of the channels consists on *Internal, Companion, External, External* LO source respectively. The channel that has the property *Internal* will be the one that will serve as reference and that will enable the capability to export its clock to the rest of the channels. The other channel in the same daughter-board will have the property *Companion*. The rest of the channels (in the other daughter-board) will be set to the property *External*, meaning that they will receive the reference from the other daughter-board. Apart from LO configurations, in order to synchronize the channels it is necessary to set the frequency using time commands. By using time commands, a timestamp will be set to make sure that the tune request is executed at the same time for all channels. This will ensure that the LOs of the X300-2TwinRx are returned synchronously on the same clock cycle and therefore assuring phase coherent operation.

C. NI-2955 calibration

There are several possible LO configurations:

- *Independent*, where each channel will use its own LO and the LO input ports will be disabled.
- *Import*, where channels will use the LO provided at the corresponding import ports.
- *Shared and Export*, where channels will use the same LO from first channel, whose LO will be exported to their LO output ports.
- *Re-import*, where the LO from first channel is exported and both channels will use the LO provided at the corresponding import ports.
- *Shared*, where both channels of the same daughter-board will use the same LO from first channel of the channel couple.

In order to achieve coherent phases among the four channels, two options are recommended:

- *LO-sharing*, where one pair of channels is configured with *Shared and Export* and the other pair is set to *Import*. Then, the LO signal from the first pair is injected to the other pair with external cables through the corresponding LO ports.
- *LO re-import*, where one pair of channels is configured with *Re-import* and the other is set to *Import*. The LO signal from the first pair is then amplified and shared among both pairs with external elements (cables, amplifiers and power splitters). This mode offers maximum coherence between channels since it shares the same locally generated and re-imported LO signal.

For simplicity in the setup, the first option has been selected in our analysis.

IV. USRP Experimental Calibration and Performance Evaluation

A. Experimental setup

This section describes the experimental validation of the phase calibration that has been conducted on the testbed illustrated in Figure 1, which is built using the USRP SDR platforms addressed herein. Calibration is done using two USRPs, one acting as a transmitter and the other one as a receiver.

On the transmitter side, the USRP N200 [21] is used for periodic transmission of Gaussian pulses through its transmit RF port. The N200 has been deployed on a PC with an Intel i7-6700 CPU @ 3.4GHz and 16GB of RAM memory and an Ubuntu 22.04 LTS OS installed. The USRP is connected with the PC through a GB Ethernet cable. The transmit RF port of the USRP N200 is then linked to a four-port splitter in order to stream the Gaussian noise signal across all four RF channels in the reception section. The Gaussian noise signal has been chosen for the channel alignment and synchronization of the RF system due to its statistical properties and mathematical convenience. These statistical nature and wide-band characteristics make it easy to model and predict its behavior, which simplifies the synchronization process. Following the proposed configuration in III, all phases should be aligned. However, it is essential to acknowledge that it is possible to use any signal limited to the capabilities of the chosen SDR model to generate and stream said signal.

On the receiver side, the USRPs N310, X300-2TwinRx and NI-2955 are used for reception once at a time. The USRPs N310 and X300-2TwinRx have been deployed on a PC with an Intel i5-9500 CPU @ 3GHz and 32GB of RAM memory and an Ubuntu 22.04 LTS OS installed. Both USRPs are connected with PC through two 10 Gigabit Ethernet cables. The NI-2955 has been deployed on a PC with an Intel i7-10700 CPU @ 2.9GHz and 32GB of RAM memory and a Windows 10 OS installed that will allow the use of LabVIEW for the signal reception and enabling the LO sharing capability explained in III. The NI-2955 is connected with the PC through a four lines MXIe 1st generation cable. In all instances, the PCs utilized are equipped with Solid Stated Drives (SSD) NVMe-based of Category 3, which potentially achieve sustained writing rates above the data throughput of the two communication channels considered.

Four experiments (one for each SDR model: NI-2955 and X300-2TwinRx; and two different experiments for the case of N310: one using the internal LO and the other using

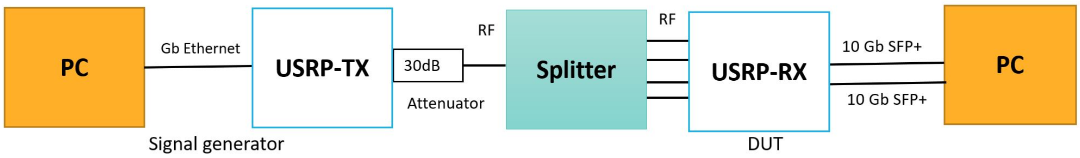

Fig. 1. Calibration testbed, where USRP N200 serves as TX and USRPs: N310, X300-2TwinRx or NI-2955 serve as RX. Note that, for the case of NI-2955, MXIe cable are used instead of 10 Gigabit Ethernet cables to connect with the PC.

an external LO) are conducted to perform phase calibration and evaluate its performance. For X300-2TwinRx and N310, two snapshots of 10 sec are taken in individual acquisitions within one power-cycle. For the case of NI-2955, in order to achieve such interval recordings, the testing procedure works acquiring small data segments of $\sim$20 ms every 200 ms, which represents a reduction of $\sim$90% of the total amount of data collected. Increasing these numbers results in failure of our acquisition procedure. The main reason for such behavior is the limitation of Windows OS to perform high-demanding real-time operations. All experiments are repeated with the re-initialization of the device. For the case of N310, when using the external LO, a Valon 5009a Dual Frequency Synthesizer [22] module is used as an external LO that is able to work in a frequency range from 23 MHz to 6000 MHz.

Finally, the Gaussian noise signal is transmitted at a carrier frequency of 2.45 GHz, transmission gain of 20 dB and sampling rate of 2 MHz. On the receiver side, the parameters used are given in Table II. The gain values are set to avoid distortion of the signal and therefore this parameter is different for each of the USRPs under analysis.

TABLE II
EXPERIMENTAL PARAMETER CONFIGURATION.

USRP	X300-2TwinRX	N310	NI-2955
Carrier frequency (GHz)	2.45	2.45	2.45
Sample rate (MHz)	50	61.44	50
Gain (dB)	60	40	40
Master clock rate (MHz)	200	122.88	200
I&Q samples (bits)	16	16	16
Nr. of power-cycles	2	2	2
Duration per power-cycle (sec)	20	20	20

B. *Performance results*

The experiments have been performed in conditions of CPU high performance achieved following the recommendations in II-A. Results obtained from the measurements verify the performance of the LO-sharing synchronization for each of the USRP models. The phase difference between channels corresponds to the angle of the maximum peak of the cross correlation between channels 1 to 3 with the first channel (0) taken as reference.

Figure 2 presents the set of results to achieve synchronization between the SDR channels through the configuration of the LO to enable the sharing ability when using the USRP X300-2TwinRx. Even after power-cycling the system, the phase difference between channels is maintained constant. Apart from being phase aligned, all channels are time aligned as well. The distribution of the calibrated phase deviation of the channels shows that the maximum phase deviation is below ± 0.08 degrees. There is a phase stability referred to the averaged phase of all channels over the observed period of time. This constitutes the reliability of the system synchronization over time. However, to guarantee higher accuracy, it is recommended performing phase calibration after a period of time.

Results depicted in Figure 3 are obtained when the USRP N310 uses its internal LO for the synchronization among channels. Because the first two channels are using the same LO as reference and the second two channels the same LO for reference but different from the previous ones, the phase deviation from the averaged phase differences is very small compared to the phase offsets between channels 2 and 3 having channel 0 as reference. This happens because the channels not sharing the same LO adds different phase noise to the data by adding additional phase error in the computation. The phase deviation reaches ± 0.68 degrees, whilst the phase deviation stays below ± 0.04 degrees when the same LO is shared. On the other hand, the phase difference among channels not sharing the same LO is not maintained constant during different acquisition even after power-cycling the system. A better performance is achieved using an external LO. Results presented in Figure 4 show a decrease in the phase error at maximum with ± 0.57 degrees. The phase difference between channels is maintained constant over time during the same power-cycle. However, a phase ambiguity of 180 degree is applied. Therefore, it is recommended performing phase calibration after every power-cycle.

Finally, Figure 5 shows that the phase deviation referred to the averaged phase of all channels in NI-2955 is below 0.12 degrees and it is maintained stable over time even after power-cycling the system. The results also show that the channels are aligned in time as well. However, it must be recalled that the dataset has not been acquired in a continuous and sustained manner (the gaps are not visible in the figure), as it was done for the other USRPs.

V. CONCLUSIONS

In this work, we showed the required steps to achieve coherent phases between RF channels for three SDRs: X300-TwinRx, N310 and NI-2955. We investigated both the phase deviation for multiple power-up cycles, as well as the phase stability over time. Results have shown that X300-2TwinRx and NI-2955 maintain constant phase differences between channels for multiple power-cycles with a maximum phase

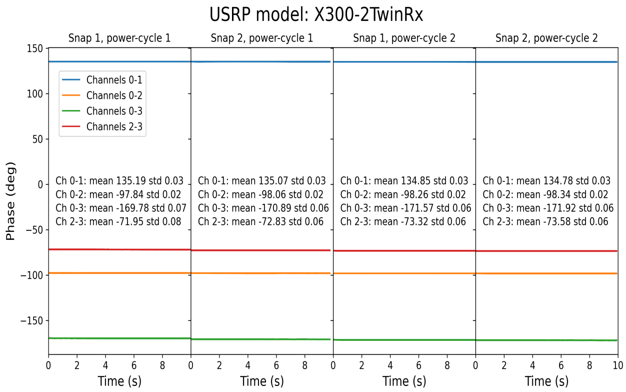

Fig. 2. Phase coherence with X300-2TwinRx.

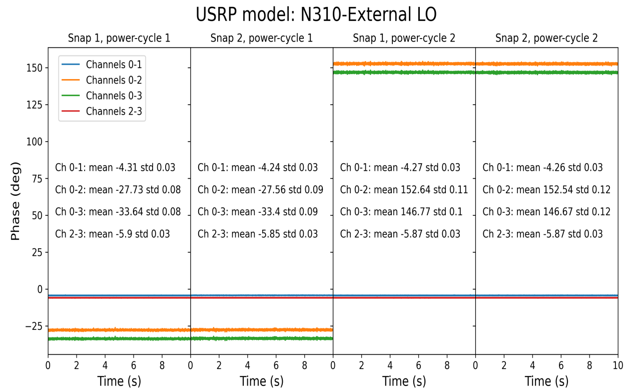

Fig. 4. Phase coherence with N310 (external LO).

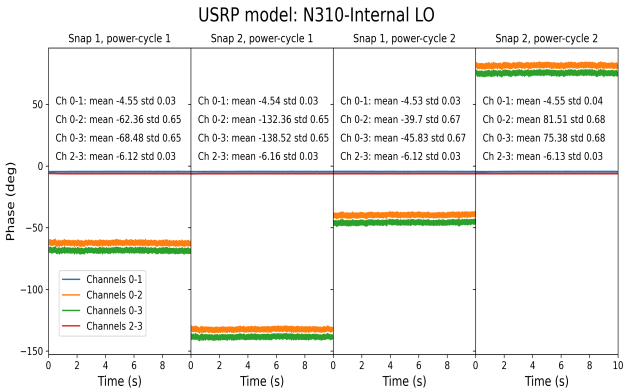

Fig. 3. Phase coherence with N310 (internal LO).

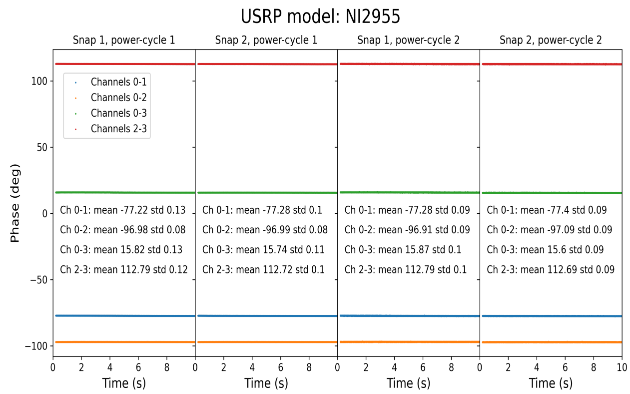

Fig. 5. Phase coherence with NI-2955.

deviation of 0.08 degrees for the case of X300-2TwinRx and 0.12 degrees for the case of NI-2955 and a phase stability over time (over 40 sec) with a maximum deviation of 2 degrees and 0.02 degrees respectively. On the other hand, results show that it is possible to achieve phase coherence between channels using an external LO as a reference when using N310. A significant improvement in the phase deviation between channels that are not sharing the same LO with a maximum of ± 0.57 degrees is observed. However, a phase ambiguity of 180 degrees is applied after re-tune or re-initialization of the device. A phase re-calibration is required. All in all, the obtained results prove the ability of all the USRPs under analysis to fulfill phase coherence. This is a requirement for high-accuracy directional of arrival computations, an important factor in achieving accurate positioning information, mainly using 5G signals. Depending on the application and type of the signal being used, the user can decide which of the USRP under analysis to use. For an easy calibration configuration without any additional hardware and high incoming sample rate, X300-2TwinRx can be used. On the other hand, NI-2955 provides a re-sampling process that allows to use a wide range of sampling rates compared to the other USRPs, but it does not provide a sustainable acquisition at higher rates limited to the Windows environment for real-time operations. Finally, despite the cost of an additional HW for phase calibration, N310 has three master clock rates that allow a higher incoming sample rate, which makes it suitable when dealing with 5G signals where sampling rate and bandwidth are important.

References

[1] J. Mitola, "Software radios: Survey, critical evaluation and future directions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 8, no. 4, pp. 25–36, 1993.

[2] B. Sun, B. Tan, W. Wang, and E. S. Lohan, "A comparative study of 3D UE positioning in 5G new radio with a single station," *Sensors*, vol. 21, no. 4, p. 1178, 2021.

[3] E. Y. Menta, N. Malm, R. Jäntti, K. Ruttik, M. Costa, and K. Leppänen, "On the performance of AoA–based localization in 5G ultra–dense networks," *IEEE Access*, vol. 7, pp. 33 870–33 880, 2019.

[4] M.-C. Hua, C.-H. Hsu, W.-J. Liao, C.-C. Yao, T.-H. Yeh, and H.-C. Liu, "Direction-of-arrival estimator using array switching on software defined radio platform," in *2011 IEEE International Symposium on Antennas and Propagation (APSURSI)*. IEEE, 2011, pp. 2821–2824.

[5] M.-C. Hua, C.-H. Hsu, and H.-C. Liu, "Implementation of direction-of-arrival estimator on software defined radio platform," in *Proc. International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*. IEEE, 2012, pp. 1–4.

[6] RTL-SDR, "About RLT-SDR," https://www.rtl-sdr.com/about-rtl-sdr/, 2023, [Accessed on 19-Jul-2023].

[7] Krakenrf, "Documentation, Supporting Files and Wiki for KrakenSDR," https://github.com/krakenrf/krakensdr_docs/wiki, 2023, [Accessed on 19-Jul-2023].

[8] Nuand, "BladeRF 2.0 Micro," https://www.nuand.com/bladerf-2-0-micro/, 2023, [Accessed on 19-Jul-2023].

[9] LimeMicrosystems, "Lime SDR," https://limemicro.com/products/boards/limesdr/, 2023, [Accessed on 19-Jul-2023].

[10] E. Research, "USRP Software Defined Radio (SDR) online catalog," https://www.ettus.com/products, 2023, [Accessed on 19-Jul-2023].

[11] J. Hershberger, T. Pratt, and R. Kossler, "Automated calibration of a dual-polarized SDR radar," in *Proc. IEEE Conference on Antenna Measurements & Applications (CAMA)*. IEEE, 2016, pp. 1–4.

[12] Y. Yang and R. S. Blum, "Phase synchronization for coherent MIMO radar: Algorithms and their analysis," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5538–5557, 2011.

[13] M. Krueckemeier, F. Schwartau, C. Monka-Ewe, and J. S. Technische, "Synchronization of multiple USRP SDRs for coherent receiver applica-

tions," in *Proc. International Conference on Software Defined Systems (SDS)*. IEEE, 2019, pp. 11–16.
[14] M. Eberhardt, P. Eschlwech, and E. Biebl, "Investigations on antenna array calibration algorithms for direction-of-arrival estimation," *Advances in Radio Science*, vol. 14, pp. 181–190, 2016.
[15] E. Research, "Ettus Research Knowledge Base: TwinRX Properties," https://files.ettus.com/manual/page_dboards.html#dboards_twinrx, 2023, [Accessed on 19-Jul-2023].
[16] E. research, "Ettus Research Knowledge Base: N310 Properties," https://files.ettus.com/manual/page_usrp_n3xx.html#n3xx_mg, 2023, [Accessed on 19-Jul-2023].
[17] N. Instruments, "Product documentation NI-2955," https://www.ni.com/docs/de-DE/bundle/usrp-2955-specs/page/specs.html, 2023, [Accessed on 19-Jul-2023].
[18] A. Xhafa, F. Fabra, J. A. López-Salcedo, and G. Seco-Granados, "Preliminary Results of 5G Direction of Arrival Estimation using In-Situ Calibration with USRP," in *Proc. International Conference on Localization and GNSS (ICL-GNSS)*, 2023, pp. 1–8.
[19] GNUradio, "GNU Radio usage manual," https://wiki.gnuradio.org/index.php/Usage_Manual, 2023, [Accessed on 19-Jul-2023].
[20] R. Pöhlmann, S. Zhang, E. Staudinger, S. Caizzone, A. Dammann, and P. A. Hoeher, "Bayesian in-situ calibration of multiport antennas for DoA estimation: Theory and measurements," *IEEE Access*, vol. 10, pp. 37 967–37 983, 2022.
[21] E. Research, "USRP N210 software defined radio (SDR)." https://www.ettus.com/product/details/UN210-KIT, 2023, [Accessed on 19-Jul-2023].
[22] V. Technologies, "Valon 5009 Frequency Synthesizer," https://www.valonrf.com/frequency-synthesizer-6ghz.html, 2023, [Accessed on 19-Jul-2023].