

# A Reconfigurable Full-Digital Architecture for Angle of Arrival Estimation

Antonello Florio<sup>1</sup>, Associate Member, IEEE, Gianfranco Avitabile<sup>2</sup>, Senior Member, IEEE, Claudio Talarico<sup>3</sup>, Senior Member, IEEE, and Giuseppe Coviello<sup>4</sup>, Senior Member, IEEE

**Abstract**—In recent years, Real-Time Localization Systems (RTLs) are gaining significant attention from the scientific community. Usually, RTLs rely on positioning techniques based on the electromagnetic (e.m.) propagation properties of the received signals. Among the many possible approaches, RTLs based on the Angle-of-Arrival (AoA) estimation allows for accurate results with a comparatively lower number of receivers. In this work, we propose a novel, fully digital synchronous architecture for AoA estimation based on phase interferometry. After reviewing the state-of-the-art on full-hardware techniques for localization, the main blocks composing the proposed architecture, are discussed in detail, along with their dimensioning equations. The granularity with which the system estimates the phase shifts used to compute the AoA is reconfigurable according to the desired accuracy. The architecture is lightweight and computes the AoA in real-time. To validate the proposed approach, the RTLs architecture has been implemented on an Intel Cyclone IV E EP4CE115F29C7 FPGA board interfaced with a custom-designed receiver front-end, and experimental results on benchmark applications have been collected and analyzed.

**Index Terms**—Direction-of-arrival, angle-of-arrival, localization, field-programmable gate arrays, phase detection, phase interferometry, real-time localization systems.

## I. INTRODUCTION

THE Angle of Arrival (AoA), also known as Direction of Arrival (DoA), is a technique that allows determining the position of a device that is transmitting an e.m. signal in a given portion of the space [1], [2], [3]. The subset of positioning techniques involving the propagation properties of e.m. waves to determine a target location is known as radiolocation. AoA relies on the fact that the direction of a given wave impinging either on a set of receivers or a single receiver with multiple antennas determines a phase shift between the different receiving channels.

AoA is an example of geometric radiolocation. Along with AoA it is also worth mentioning the Time-of-Arrival (ToA) and the Time Difference of Arrival (TDoA) techniques [4].

Manuscript received 10 May 2023; revised 27 July 2023 and 3 October 2023; accepted 15 December 2023. This article was recommended by Associate Editor R. Yazicigil. (Corresponding author: Giuseppe Coviello.)

Antonello Florio, Gianfranco Avitabile, and Giuseppe Coviello are with the Department of Electrical and Information Engineering, Polytechnic University of Bari, 70125 Bari, Italy (e-mail: giuseppe.coviello@poliba.it).

Claudio Talarico is with the Department of Electrical and Computer Engineering, Gonzaga University, Spokane, WA 99258 USA (e-mail: talarico@gonzaga.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCSI.2023.3345161>.

Digital Object Identifier 10.1109/TCSI.2023.3345161

In ToA and TDoA the property exploited is the time it takes the e.m. signal to travel to the receiver or the time difference at different receivers. Once the trip time is known, it is relatively simple to compute the distance and apply multilateration techniques to recover the position in space. However, those techniques impose some constraints on the time synchronization of the devices. In particular, in ToA, transmitters and receivers require a strict synchronization process to agree on a common time reference and therefore define a precise start and stop time for the trip time definition. In TDoA, the constraint is slightly relaxed, requiring only synchronization to take place among the receivers. When comparing AoA and TDoA, it is worth mentioning that for an  $n$ -dimensional scenario, ToA/TDoA require  $n+1$  receivers, while AoA requires only  $n$  receivers [2]. However, in literature, some hybrid TDoA/AoA approaches such as the work of Aernouts et al. [5], have been developed for reducing the number of needed receivers.

Another class of localization approaches involves non-geometric techniques [1]. Received Signal Strength (RSS) belongs to this latter class. In RSS the position information derives from the fact the power associated with a wave traveling in space decreases with the square of the traveled distance. In this way, by using specific propagation models (also called path-loss models), it is possible to reconstruct the distance of the transmitter once the transmitted and the received power levels are known [6]. In practice, this is performed by using a set of receivers to implement multilateration algorithms that lead to the definition of the position according to a reference system. RSS belongs to the non-geometric technique since it is common to deploy it along with fingerprinting procedures. Usually, RSS-based localization techniques involve two phases: an offline phase and an online phase [7]. In the offline phase there is the definition of a so-called radio map in which every point of the space is associated with a given power level. In the online phase, the RSS analysis taking place consists of comparing the received power level with the expected power level using either a deterministic or probabilistic approach [8]. The problem is that a slight change in the environment can determine the need of repeating the offline phase, because the propagating scenario has changed, and this causes a huge overhead in terms of complexity. On the other hand, if the scenario is sufficiently static, the online phase can be performed very simply since it is sufficient to have an envelope detector and a power level comparator to extract the RSS information. Like the geometric techniques, RSS accuracy can

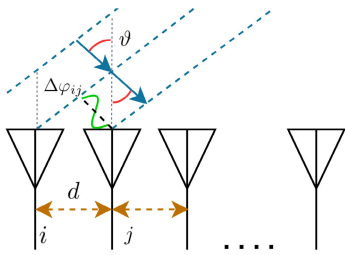


Fig. 1. A ULA receiving a plane wave from the direction  $\vartheta$  determining a phase shift  $\Delta\varphi_{ij}$  between antenna elements  $i$  and  $j$ .

also be improved through hybrid techniques relying on mixing geometrical and non-geometrical approaches [9].

This paper focuses on the estimation of AoA employing dedicated hardware. Over the years a lot of techniques have been developed for localization, but usually, they operate at the application layer of the networking stack [10]. We aim to bring the localization process to the physical (PHY) layer, thus making this information available to the upper layers of the networking stack. Location estimations performed at the PHY layer allow the saving of computational resources for performing other tasks. This feature is especially appealing for power-constrained devices [11]. Once the AoA is available, it is possible to directly exploit this information to obtain significant communication improvements by using the AoA information to perform adaptive and real-time beam steering and beamforming [12]. Furthermore, by employing dedicated hardware and thus removing delays relative to the context switch computation, it is possible to reduce the latency to obtain the estimation and achieve real-time operations [13].

The AoA is the angle subtended between the perpendicular to the plane containing the receiver and the direction of the incoming wavefront. To develop a formal definition of the AoA and the induced phase shift, we make some hypotheses. First, we assume to operate in far-field, thus we can consider the wave to be plane and the direction of the wavefront to be the same for each of the wavefront points that will intercept the receiving set of antennas. Furthermore, we will focus on the simple case of azimuthal AoA estimation. We assume to work with a Uniform Linear Array of  $M$  antennas ( $M$ -ULA) with spacing  $d$  (Figure 1). Under these hypotheses the AoA  $\vartheta$  can be related to the phase shift measured between the antenna elements  $i$  and  $j$  according to:

$$\Delta\varphi_{ij} = \frac{2\pi}{\lambda} d(i - j) \sin \vartheta \quad (1)$$

where  $\lambda$  is the wavelength. Let us now simplify (1) by calling  $\Psi_{ij}$  the equivalent array spacing:

$$\Psi_{ij} = \frac{d(i - j)}{\lambda} \quad (2)$$

At this point, we can redefine (1) as a function of  $\Psi_{ij}$

$$\Delta\varphi_{ij} = 2\pi\Psi_{ij} \sin \vartheta \quad (3)$$

By considering (3) is possible to transform the AoA estimation problem into a phase difference estimation problem, i.e. we perform AoA estimation through phase interferometry [14].

In particular, considering different antenna couples, it is possible to determine different estimations of the same angle  $\vartheta$ . The relationship between the measured phase shift  $\Delta\varphi_{ij}$  and the AoA  $\vartheta$  estimated through the antenna elements  $i$  and  $j$  (namely  $\vartheta_{ij}$ ) is

$$\vartheta_{ij} = \arcsin\left(\frac{\Delta\varphi_{ij}}{2\pi\Psi_{ij}}\right) \quad (4)$$

where  $\Psi_{ij}$  is known since the quantities it depends on are already known. Therefore, the only parameter required to compute  $\vartheta_{ij}$  is the phase difference  $\Delta\varphi_{ij}$ .

#### A. Contributions and Paper Organization

In the forthcoming sections, we will:

- Introduce a new technique for AoA estimation based on phase interferometry and implement it as a full-digital dedicated hardware. The proposed solution is modular and reconfigurable according to the level of precision desired for a specific application by changing word length and clock frequency.
- Discuss the implementation of the proposed digital hardware architecture on an FPGA. The architecture was implemented on a digital reconfigurable hardware platform based on the Cyclone IV E EP4CE115F29C7 Intel's FPGA.
- Analyze the performance achieved by the proposed architecture by presenting its main blocks and the equations determining the theoretical accuracy achievable. The architecture is implemented on a digital hardware platform based on the Cyclone IV E EP4CE115F29C7 Intel's FPGA.
- Validate the approach through a large set of experiments involving both custom-designed analog hardware boards and Hardware-in-the-Loop testing tools.
- Evaluate the impact of the reconfiguration parameters on the accuracy of the approach.
- Compare the achieved performance with classical benchmark algorithms available in literature.

The rest of the paper is organized as follows. Section II presents a short survey of the hardware-based approaches and algorithms available for geometric localization, with particular emphasis on AoA techniques. In Section III we describe, in detail, the proposed architecture, and derive the equations required for dimensioning the system. In Section IV we discuss the experimental setup, the hardware developed for the data acquisition, and the metrics employed for the performance analysis. In Section V we present and comment on the achieved results and the metrics used to evaluate performances. Finally, Section VI closes this work, proposing future improvements to investigate.

## II. RELATED WORKS

Over the years, several approaches have been proposed for performing localization through full-hardware architectures. As mentioned in the introduction, full-hardware approaches are the key to achieve real-time response. By surveying the

literature, two possible approaches emerge to solve as the best the problem.

The first approach focuses on implementing the positioning algorithms on digital architectures rather than general-purpose application layers. In this paper, we will focus on three of the most well-known positioning algorithms in the field of DoA estimation. The first one is the Multiple Signal Classification Algorithm (MUSIC), developed by Schmidt [15]. MUSIC deals with the computation of the covariance matrix of the signals received by an array of antennas. To allow the separation of the noise subspace from the signal subspace, the algorithm performs eigenvalue computation and reordering. These values are then employed for the computation of a pseudo-spectrum function, whose peaks correspond to the estimated DoAs. Each of the described steps requires a huge computational load [16]. This is because MUSIC works with complex-valued numbers and equations, and the Eigen-Value Decomposition (EVD) phase and the peak search phase are computationally intensive, even on dedicated hardware. Some of the proposed improvements aim to implement variations to the standard MUSIC algorithm to make it hardware efficient. For example, under given hypotheses, some implementations involve working with real values instead of complex numbers. This is the case of [17], where Minseok et al. propose to use a linear transformation to perform the EVD with real numbers only. The approach was tested on the Altera EP20K600 FPGA producing the final result in tens of microseconds. In [18], Wang et al. support the linear transformation with substeps decomposition of the EVD phase employing the CORDIC algorithm. In [19], Chen et al. present a hardware-friendly implementation of the MUSIC algorithm in which they decrease the EVD computational cost by exploiting the conjugate symmetry property of the covariance matrix. The same property is employed for decreasing the memory access time. The algorithm was simulated to work up to 1 GHz clock frequency, on a TMS320C40 40-nm CMOS technology. In [16], Butt et al. implemented an optimized version of the MUSIC algorithm with reduced processing time and resource occupation using a Xilinx FPGA. The improvement is obtained by normalizing the covariance matrix and reducing its size by considering only three adjacent antennas and then choosing to evaluate the best-received signal level. The estimation takes a few  $\mu$ s. However, the peak search range is limited to 60 degrees. In [20] Li et al. propose an EVD decomposition technique for Hermitian matrices with low resource occupancy and low latency. The MUSIC optimization technique proposed has been demonstrated for sparse arrays and ULAs, respectively.

Many other position algorithms were strictly developed for ULAs scenarios, where the so-called steering matrix is a Vandermonde matrix [10]. This is the case of the root-MUSIC (RMUSIC) algorithm [21], in which the peak search is substituted with a polynomial roots search. The RMUSIC algorithm has been empirically proven to perform significantly better than MUSIC on small samples [10]. Another algorithm that exploits the structure of the Vandermonde steering matrix is the Estimation of Signal Parameters via the Rotational-Invariance Technique (ESPRIT) algorithm [22]. However, both RMUSIC and ESPRIT have the same computational

complexity issue since both of them make use of EVD. Also in this case several implementations have been proposed to try to lighten the computational complexity. In [23], Boonyanant et al. propose the implementation of Recursive ESPRIT on a Xilinx Virtex-II FPGA. In this implementation, similarly to [17] they make use of a linear transformation to operate only on real numbers. This was one of the first efforts for implementing ESPRIT in digital hardware, and the implementation took about half of the available resources. More recently, in [24] Jung et al. propose an ESPRIT-based scalable system that supports 2 to 8 array elements. The processor is based on a multiple invariances algorithm, with better performances than the existing ESPRIT processor, and the performances are increased thanks to the least-square and Jacobi methods for EVD. The system was tested on a Xilinx Virtex-5 FPGA.

Finally, The second approach for deploying full-hardware positioning systems is to implement completely novel solutions. The majority of these systems are based on geometrical localization. For instance, Piccinni et al. in [4] propose a system for distance measurement and positioning through trilateration that relies on the TDoA analysis of an OFDM signal transmitting the Frank-Zadoff-Chu sequence. The core of the system is implemented through an Intel Stratix IV EP4SGX70HF35C3 FPGA. The system is a good example of RTLS. Another RTLS was developed by Bottigliero et al. in [25]. Here, the authors propose a TDoA-based localization system that makes use of Ultra-Wide Band (UWB) pulses. The coarse stage ToA estimation is performed on a Xilinx Zynq 7000 SoC FPGA that is in charge of correlating the received sequence with the stored one. The fine coarse ToA estimation is performed on an ARM processor. In [26], Zhang et al. propose a Fast-Fourier Transform-based (FFT-based) DoA estimation scheme. The FFT is run on the second-order difference co-array sum of the co-array (DCSC) signals. The simulation shows that the DOAs can be retrieved from the resulting spatial response with an improved resolution. Neunteufel et al. in [27] propose a system based on a novel wide-band chirp modulation scheme along with an architecture of anchor nodes suitably designed to estimate both AoA and ToA in the ISM band. The validation is done through reconfigurable hardware. In [28] Guo et al. propose a hardware-efficient DoA estimator dedicated to unequal-sized subarrays. The computing core consists of 3 CORDIC units and it is implemented on a 28 nm CMOS technology. The system is clocked at 2.17 GHz. In [29] BniLam et al. propose an AoA estimation system based on phase interferometry of LoRa signals. Once the phase differences are estimated by analog phase detectors, they are sent through LoRa packets to a LoRaWAN gateway connected to a processing system that extrapolates the AoA information.

### III. DESCRIPTION OF THE PROPOSED ARCHITECTURE

The proposed architecture takes as inputs, continuous-wave (CW) signals, which are suitably preconditioned to be further processed through the digital logic. The architecture is composed of three macroblocks: the Phase Estimation Block (PEB), the Frequency Estimation Block (FEB), and the AoA Computation Block (ACB). In this section, we detail the

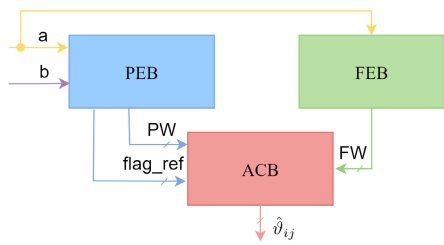


Fig. 2. Macroblocs and main signals interaction diagram, where  $a$  and  $b$  are the sample input channels.

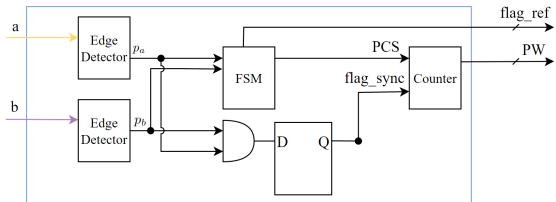


Fig. 3. Phase Estimation Block (PEB) diagram.

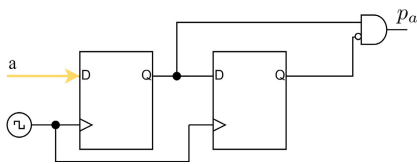


Fig. 4. Structure of the positive-edge detector circuit for signal  $a$ .

functionalities of each block. The macroblocks interaction diagram is shown in Figure 2. All blocks are driven by the same clock signal, so for the sake of compactness, the clock signal is omitted.

#### A. Phase Estimation Block (PEB)

The PEB is in charge of computing the phase difference between two channels, taking one of them as a reference. The computed phase shifts are in the range  $[0, 2\pi]$  and are represented by a binary word of length  $L$ , called the Phase Word (PW). The PW quantizes the time delay between the input signals and is equal to the number of system clock ticks corresponding to the measured delay. The PEB subblocks are depicted in Figure 3.

Let us consider two channels labeled as  $a, b$ . On both channels it is performed a positive edge detection (Figure 4), giving rise to the two pulses  $p_a, p_b$  with time duration  $T_{clk} = 1/f_{clk}$ , where  $f_{clk}$  is the system clock frequency. The frequency of  $p_a, p_b$  is nominally equal to the one of the input signals  $a$  and  $b$ , that is  $f_{IF}$ .

The signals  $p_a, p_b$  are fed to a Moore-like Finite State Machine (FSM). The introduction of this subblock is for keeping track of the reference channel with respect to the time delay is computed. As an example, let us assume  $a$  is the default reference channel. If at any point in time  $b$  becomes the leading signal, without a reference tracking mechanism, the system would compute the phase shift considering the time difference between the pulses  $p_b$  and  $p_a$ , i.e. as if the leading channel would be  $a$ . However, with a reference tracking mechanism, the value can correctly referred to the channel

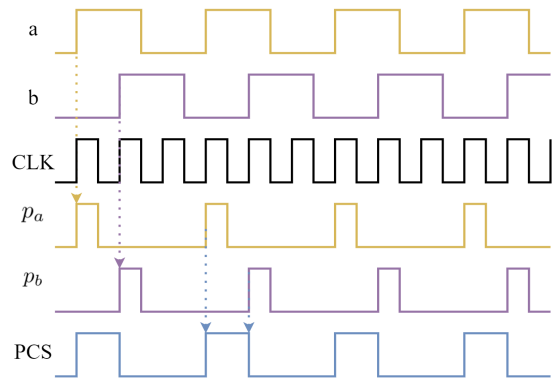


Fig. 5. Timing diagram illustrating the PCS signal generation starting from the sample preconditioned input signals (neglecting the latencies).

$a$  by computing the  $2\pi$ -complement of the estimated phase. The choice of a Moore FSM improves the synchronicity to the clock. The FSM has two outputs: the Phase Control Signal (PCS), whose duty cycle is proportional to the phase shift between the signals, and `flag_ref` which labels the reference channel (Figure 5). The FSM is described by a state space  $\mathcal{S}$  whose cardinality is  $|\mathcal{S}| = 5$ :

$$\mathcal{S} = \{zero, a\_lead, b\_lead, a\_zero, b\_zero\} \quad (5)$$

Each state has a specific role and performs a given set of tasks:

- *zero*: it is the initial state. In this state, the PCS is set to low, and the `flag_ref` is set to `''00''`
- *a\_lead*: this state is triggered when the first detected pulse rising edge is on  $a$ . In this state, PCS is set to high, and `flag_ref` is set to `1101''`
- *a\_zero*: this state is triggered after the state *a\_lead* is reached and a pulse rising edge on  $b$  is detected. In that case, PCS is set to low and `flag_ref` is kept to `''01''`.
- *b\_lead*: performs the same operations as *a\_lead*, but when the first detected edge is on  $b$ . Hence, the `flag_ref` is set to be `''10''` and the PCS is set to high.
- *b\_zero*: it is the dual of *a\_zero*, with `flag_ref` set to be `''10''` and PCS set to low.

When the states *a\_zero* and *b\_zero* are reached, in the rare case the inputs are synchronous (i.e. null phase difference), a transition towards *zero* is triggered, thus resetting the choice on the reference channel. This feature is implemented to also keep track of moving objects. The FSM state transition graph is depicted in Figure 6. Note that in the picture we used the notation  $(x, y) = (p_a, p_b) \in (0, 1)$  to illustrate the state transitions, where  $p_i, i = \{a, b\}$  represent the outputs of the edge detectors.

In parallel to the FSM there is another synchronization checking mechanism that takes  $p_a, p_b$  as inputs and it performs a registered AND operation between them, generating as output the signal `flag_sync`.

The PCS acts as the enable input of a *modulo*  $2^L$ -counter driven by the system clock. The counter acts as a coarse time-to-digit (T2D) converter associating a Phase Word (PW) to the phase delay. The counter output is stored and held until a change in the duty cycle of the PCS is detected. The count is updated only when the PCS goes to low (hence, the transition

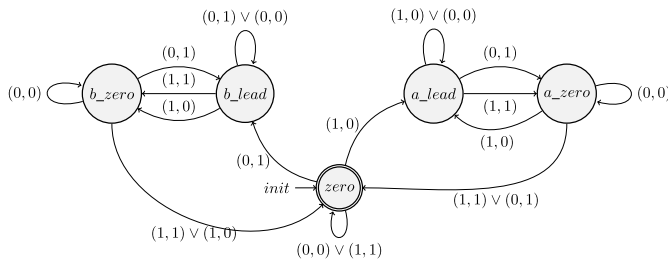


Fig. 6. States transition diagram for the Moore-like FSM of the PEB with inputs  $(p_a, p_b) \in \{0, 1\} \times \{0, 1\}$ .

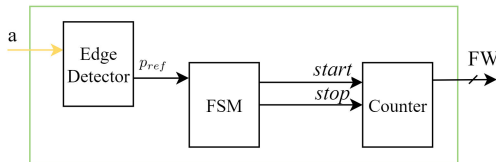


Fig. 7. Frequency Estimation Block (FEB) diagram.

is masked). The signal  $flag\_sync$  is in charge both of resetting the final counting value and disabling the counter with only one clock cycle of delay with respect to when synchronicity is detected.

The PEB latency depends on the phase shift estimated. The latency has a static component due to the edge detectors and the FSM, and a variable component due to the counter update mechanism. The static component takes 3 clock cycles (CC), while the variable component takes in the worst case a number of CC equal to the maximum value the PW can represent.

### B. Frequency Estimation Block (FEB)

The FEB is in charge of quantizing the duration of the period of the input signals. Under the hypothesis that each signal on each channel has the same frequency, the FEB needs to take as its only input the reference channel. The output of the FEB is a binary word of length  $C$ , called Frequency Word (FW), and representing the number of clock ticks of the period, that is:

$$FW = \left\lceil \frac{f_{clk}}{f_{IF}} \right\rceil \quad (6)$$

where  $f_{IF}$  is the input signal frequency,  $f_{clk}$  is the system clock frequency, and  $\lceil \cdot \rceil$  the ceiling operator. The structure of the FEB is very similar to the one of the PEB, with slight differences, and it is depicted in Figure 7.

First, edge detection is performed on the unique input  $a$ . The edge detector has the same structure as the one used in the PEB, depicted in Figure 4. Then, the pulse  $p_{ref}$  is fed to a Moore-like FSM with three states that generate the enable and disable signal to a *modulo*  $2^C$ -counter (Figure 8). The output signal from the FSM is set to high when is in the state *start* and is set to 0 in the other states. The FSM goes into the *start* state if the output of the edge detector is high, and it stays there until a new pulse is detected. The FEB latency is determined as the worst-case latency of the PEB.

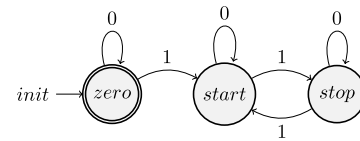


Fig. 8. State diagram for the Moore-like FSM of the FEB, describing the state interaction depending on the input pulse  $p_{ref} \in \{0, 1\}$ .

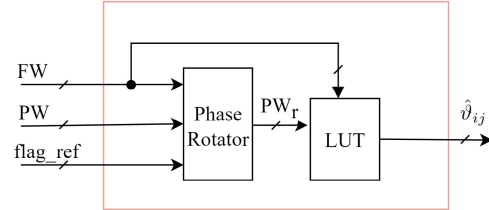


Fig. 9. AoA Computation Block (ACB) diagram.

### C. AoA Computation Block (ACB)

The ACB takes as inputs the PW,  $flag\_ref$ , and FW from the PEB and FEB and determines the AoA estimation  $\hat{\vartheta}_{ab}$ . Its block diagram is depicted in Figure 9.

The first block is the Phase Rotator. The Phase Rotator takes as inputs FW, PW and  $flag\_ref$  and returns the correct phase shift with respect to the defined reference channel. Its behavior is described in Algorithm 1. By looking at equation (4), we see that we have to compute the arcsine. To do this, there are several approaches, ranging from complex algorithms to Look-Up Tables (LUTs) [30]. In order to lighten the computational weight associated with this step, we chose to implement the arcsine operation through a LUT. Let us refer to the LUT address length as  $A$ , expressed in bits. In this way, the LUT can be populated with  $2^A$  different values. Note that the LUT depth choice imposes a trade-off between the precision of the AoA representation and the size of the LUT. To explain better the main steps involved, first, we recall that the relationship between time delay  $\Delta\tau_{ij}$  and phase shift  $\Delta\varphi_{ij}$  for a CW signal at frequency  $f_{IF}$  is:

$$\Delta\varphi_{ij} = 2\pi \Delta\tau_{ij} f_{IF} \quad (7)$$

---

#### Algorithm 1 Phase Rotation With Respect to the Reference Channel

---

**Input:**  $PW, FW, flag\_ref$

**begin**

**switch**  $flag\_ref$  **do**

**case**  $00$  **do**

$PW_r = 0$

**case**  $01$  **do**

$PW_r = PW$

**case**  $10$  **do**

$PW_r = FW - PW$

**otherwise do**

      do nothing

**return**  $PW_r$

---

Therefore,

$$\frac{\Delta\varphi_{ij}}{2\pi} = \frac{\tau_{ij}}{T_{IF}} \quad (8)$$

with  $T_{IF} = (f_{IF})^{-1}$ . By substituting in the (4), we find

$$\vartheta_{ij} = \arcsin\left(\frac{\tau_{ij}}{T_{IF}\Psi_{ij}}\right) \quad (9)$$

If we consider the available estimations, this leads to

$$\hat{\vartheta}_{ij} = \arcsin\left(\frac{PW}{FW \cdot \Psi_{ij}}\right) \quad (10)$$

Therefore we can adjust the argument of the (10) to be used as the address for indexing the LUT.

The ACB has a latency of 6 CC, of which, 5 CC are needed for the address generation and 1 CC for the LUT response.

#### D. System Dimensioning

After describing each block functionality, we can now delve into the block parameters dimensioning.

1) *Clock Frequency*: For a CW signal with a frequency  $f_{IF}$ , it is possible to link the phase shift to a time delay through the relationship:

$$\Delta\varphi = 2\pi f_{IF}\tau \quad (11)$$

For a system with clock frequency  $f_{clk}$ , to resolve a time delay  $\tau$ , the corresponding clock period  $T_{clk}$  should be less or equal to the delay to resolve, i.e.

$$T_{clk} \leq \tau(\Delta\varphi, f_{IF}) \quad (12)$$

where  $\tau(\Delta\varphi, f_{IF})$  denotes the time delay corresponding to the phase shift  $\Delta\varphi$  associated with the signal of frequency  $f_{IF}$ :

$$\tau(\Delta\varphi, f_{IF}) = \frac{\Delta\varphi}{2\pi f_{IF}} \quad (13)$$

Therefore, substituting the (13) in the (12), it is possible to define the minimum resolvable phase shift as:

$$\Delta\varphi_{min} = 2\pi \frac{f_{IF}}{f_{clk}} \quad (14)$$

Finally, by substituting the (14) in the (3) it is possible to find the minimum resolvable AoA as follows:

$$\vartheta_{min,ij} = \arcsin\left(\frac{1}{\Psi_{ij}} \frac{f_{IF}}{f_{clk}}\right) \quad (15)$$

2) *PW Length*: The counter is enabled when the digital signal PCS is high. PCS has the same frequency as the input signals but with a variable duty cycle. If the PW is  $L$ -bits long, the maximum time PCS may be high is  $2^L - 1$  clock ticks, therefore:

$$T_{IF} \leq (2^L - 1) \cdot T_{clk} \quad (16)$$

with  $T_{IF}$  being the input signal period. Therefore, the minimum number of bits required to represent PW is:

$$L_{min} = \left\lceil \log_2\left(\frac{f_{clk}}{f_{IF}} + 1\right) \right\rceil \quad (17)$$

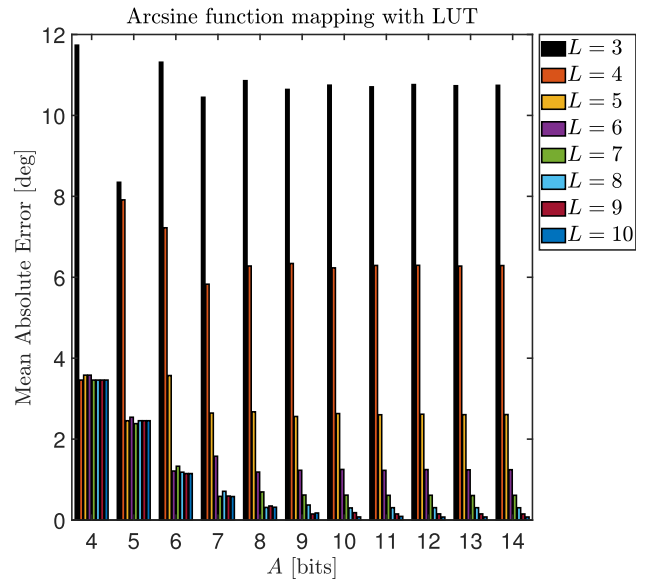


Fig. 10. Mean absolute error varying the LUT address length  $A$  and the PW length  $L$ .

3) *FW Length and Maximum Input Frequency*: Following the same considerations used for the PW length dimensioning, we can derive a similar expression for the FW length. The *start* signal produced by the FSM has the same frequency as the input signal, that is  $f_{IF}$ . Hence, similarly to (17):

$$C_{min} = \left\lceil \log_2\left(\frac{f_{clk}}{f_{IF}} + 1\right) \right\rceil \quad (18)$$

If we consider  $C = L$ , that is PW and FW have the same length, it is possible to define the maximum allowable input frequency as follows:

$$f_{IF,max} = \frac{f_{clk}}{(2^L - 1)} \quad (19)$$

4) *ACB LUT Dimensioning*: The purpose of the LUT inside the ACB is to perform the computation of the inverse sine (arcsin) function in (10). We recall that the arcsin function is defined as the function such that

$$x \in [-1, +1] \mapsto \arcsin(x) \in [-\pi/2, +\pi/2] \quad (20)$$

It becomes apparent that the values to store in the LUT always fall in the range  $[-\pi/2, +\pi/2]$ , so the absolute value of the integer part can be only 0, 1. To keep complexity low, we can represent the values stored in the LUT using  $B$  bits with fixed point notation: one bit for the sign, one bit for the integer part of the value, and  $B - 2$  bits for the fractional part.

The final error accumulated on the AoA estimation depends on the word length  $L$  (i.e. the ratio between  $f_{IF}$  and  $f_{clk}$ ), and the address length  $A$  of the LUT. By considering a ULA spacing of  $\lambda/2$  and two adjacent antenna elements (i.e.,  $|\Psi_{ij}| = 0.5$ ), we can study the impact of these parameters on the final estimation. The statistic of the mean absolute error is provided in Figure 10.

The error exhibits a negative power dependency with the address length  $A$ . For  $A \geq 7$  the effect of the LUT depth on the error becomes negligible and we can assume that the error depends only on  $L$ .



Fig. 11. Layout of the proposed architecture mapped on the Intel Cyclone IV E EP4CE115F29C7 FPGA (4 channels,  $L = C = 8$  bits,  $A = 7$  bits).

TABLE I

INTEL CYCLONE IV E EP4CE115F29C7 POST-FITTING RESOURCES OCCUPANCY SUMMARY

Parameter	Value
Combinational Functions	1082
Logic Elements	1104 (0.98%)
Total registers	441
Total pins	27 (5%)

5) *Prototype*: The proposed architecture has been prototyped on the Intel Cyclone IV-E EP4CE115F29C7 FPGA available on the Terasic DE2-115 development board using the Intel Quartus Prime development environment. The architecture has 4 input channels and delivers three estimations for each AoA. The PW length and the FW length have been set to  $L = C = 8$  bits. The LUT address length is set to  $A = 7$ . The physical layout of the FPGA is shown in Figure 11. Table I summarizes the FPGA fitting results in terms of resource allocation.

The compilation reports outline a maximum clock frequency of 210 MHz and an estimated total power consumption of 136.73 mW.

#### IV. EXPERIMENTAL SETUP

To validate the proposed architecture, we implemented an experimental testbed that involved both hardware and software components. To compare our approach with classical array signal processing algorithms from literature, it was necessary to store the waveforms of the received signals and process them through a MATLAB/Simulink model that is in charge of digitalizing and conditioning the experimental analog signals acquired by the receiving antennas. The digital data obtained via MATLAB/Simulink is then fed to our proposed AoA estimation hardware through a custom digital interface implemented inside the Intel Cyclone IV E EP4CE115F29C7 FPGA via the Simulink's FPGA-in-the-Loop® (FiL) feature [31]. The FiL feature is only employed as an interface between the stored waveforms and the FPGA. The core AoA estimation architecture proposed is implemented upfront using VHDL-2008 and Quartus Prime.

The precision of the estimations achieved through the proposed architecture has been validated by analyzing the results

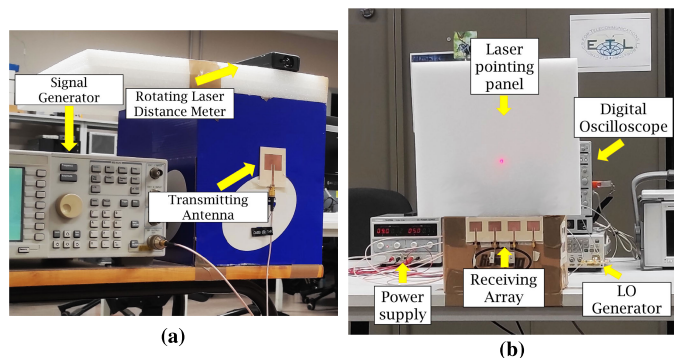


Fig. 12. (a) Transmitter setup: transmitting antenna, signal generator, and laser distance meter to evaluate both  $l$  and  $x$ . (b) Receiving array setup. Behind the pointing panel it is possible to glimpse the digital oscilloscope and the LO signal generator.

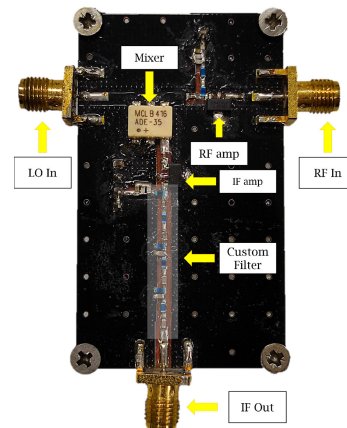


Fig. 13. The custom-designed receiving front-end.

obtained on three classical algorithms: MUSIC, RMUSIC, and ESPRIT. For each of them, we considered as reference their MATLAB implementation.

To demonstrate the different performances achievable by simply scaling the clock frequency we repeated the experiments at three different frequencies  $f_{clk} = \{50, 100, 200\}$  MHz. Unfortunately, due to the limitations of the FiL feature, it was not possible to employ the same testbed for frequencies over 200 MHz [31].

#### A. Experiment Data Acquisition

The experiment was conducted in an indoor environment (i.e., the laboratory) without any attempt to reduce or prevent the multipath phenomenon. Regarding the practical execution of the experiment, the transmitter was moved by an operator towards the different positions on the perpendicular to the array broadside axis, while the receiver was set to be fixed. The transmission frequency was chosen to be  $f_{RF} = 3.36$  GHz, to be resilient to other telecommunication standard interference during the experiment. The distance was chosen to satisfy the condition of plane wave. Given the considered frequency, the minimum distance is  $l_{min} \approx 0.89$  m. Hence, the distance  $l = 2.20$  m  $> l_{min}$  was chosen. The transmitter radiated a CW signal at frequency  $f_{RF}$  with power  $p_{tx} = +10$  dBm through a custom patch antenna element with  $-13$  dB return loss. The

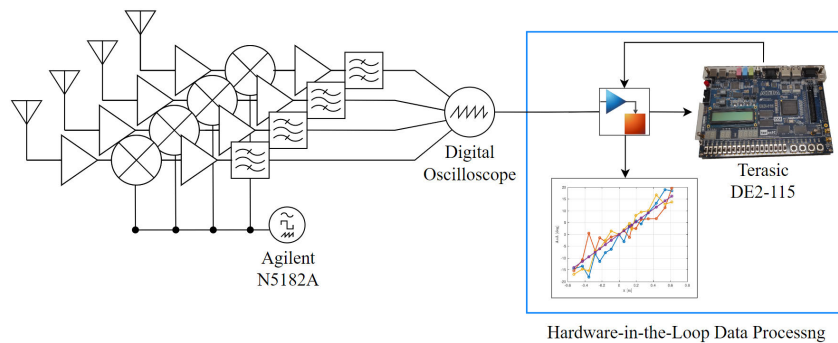


Fig. 14. Block diagram depicting the receiver side of the experimental setup, along with the interfacing with Simulink.

HP E4433B Series Signal Generator [32] was employed as the transmitter. Figure 12 shows some details on the experimental setup.

The receiving end of the system is in charge of acquiring, amplifying, downconverting, filtering, and quantizing the incoming signal, and relies on a custom-designed 4-element Uniform-Linear-Array of antennas (4)-ULA with spacing  $d = \lambda/2$ . In particular, the array is composed of antennas nominally identical to the transmitting one and shares a common ground plane. Each antenna is connected to a custom-designed RF front-end (Figure 13), composed of an RF path amplifier, a passive mixer for downconversion, an IF path amplifier and a custom band-pass filter that has been implemented as a 5<sup>th</sup> order Chebyshev filter realized through lumped components because of the low analog IF frequency, 150 MHz. The IF signal is then sampled and quantized through a digital oscilloscope with 12-bit ADC, 1 GHz bandwidth, and sampling frequency  $f_s = 5$  GS/s. The local oscillator signal is a CW with  $f_{LO} = 3.21$  GHz generated through an Agilent N5182A Vector Signal Generator [33]. Thus the analog IF is 150 MHz. No hypothesis has been made on the synchronization between transmitter and receiver, i.e. each side of the communication pairs rely on its own frequency sources which are independent from one another. The Ground-Truth (GT) AoA to which compare the estimations is computed as

$$\vartheta(x) = \arctan\left(\frac{x}{l}\right) \cdot \frac{180}{\pi} \text{ [deg]} \quad (21)$$

To study the repeatability of the estimations, for each experiment, we acquired 20 traces. Each of the acquisitions spanned the downconverted signal dynamic for a time window of 2  $\mu$ s. To ensure statistical independence, each trace acquisition has been taken after an idle interval of time of 500 ms.

The calibration point is set in the geometrical center of the array at broadside distance  $l$  in the absolute reference system, in order to seek for the calibration condition:

$$\vartheta_{12} = \vartheta_{23} = \vartheta_{34} = 0 \quad (22)$$

### B. Data Processing and Digital Hardware Interfacing

The entire receiver side of the architecture is depicted in Figure 14, where it is possible to see how the analog and the digital portions are interfaced. The results of the oscilloscope acquisition are first undersampled to the clock frequency

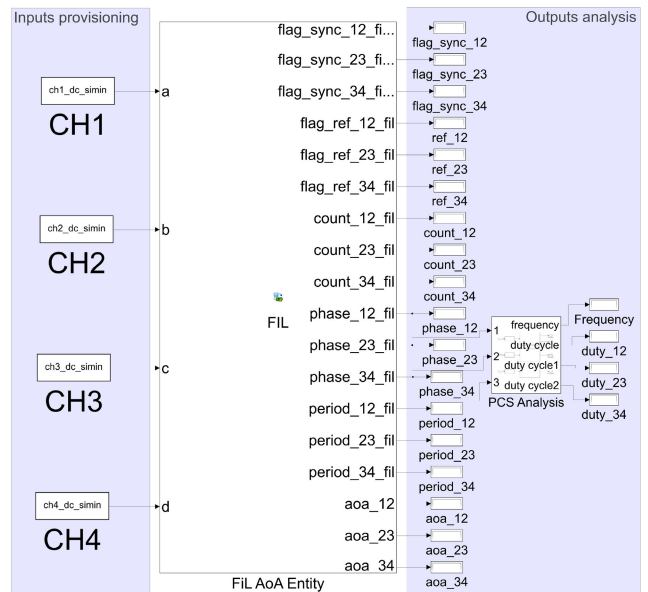


Fig. 15. FiL block diagram for testing the digital architecture.

employed for performing the test, and then digitally downconverted, to the frequency  $f_{IF} = 5$  MHz. Then, the waves are fed to a block whose operation models the behavior performed by a squaring circuit and is summarized in Algorithm 2. The output of this block is then processed through the FiL feature of MATLAB/Simulink (Figure 15) via Ethernet interfacing [31]. This feature enables us to feed the Hardware-Under-Test (HUT) with data from MATLAB/Simulink and capture the results of the computations. For each of the clock frequencies  $f_{clk} = \{50, 100, 200\}$  MHz used to validate the architecture, we generated a different FiL programming binary file for the FPGA.

We will denote with  $\hat{\vartheta}_{ij}(x)$  the estimation of the AoA corresponding to the position  $x$  relative to the calibration point through the antenna pair composed by the elements  $i, j$ . In particular, we will consider the antenna pairs  $\{12\}, \{23\}, \{34\}$ .

In Figure 16b-e, we summarized some of the test signals the FiL block is outputting for diagnostic purposes. Moreover, Figure 16a shows the input signal to the squarer block, and Figure 16f shows the AoA output plot. Given the availability of three different estimations one for each antenna pair, we will consider their average. The reason is that each receiving



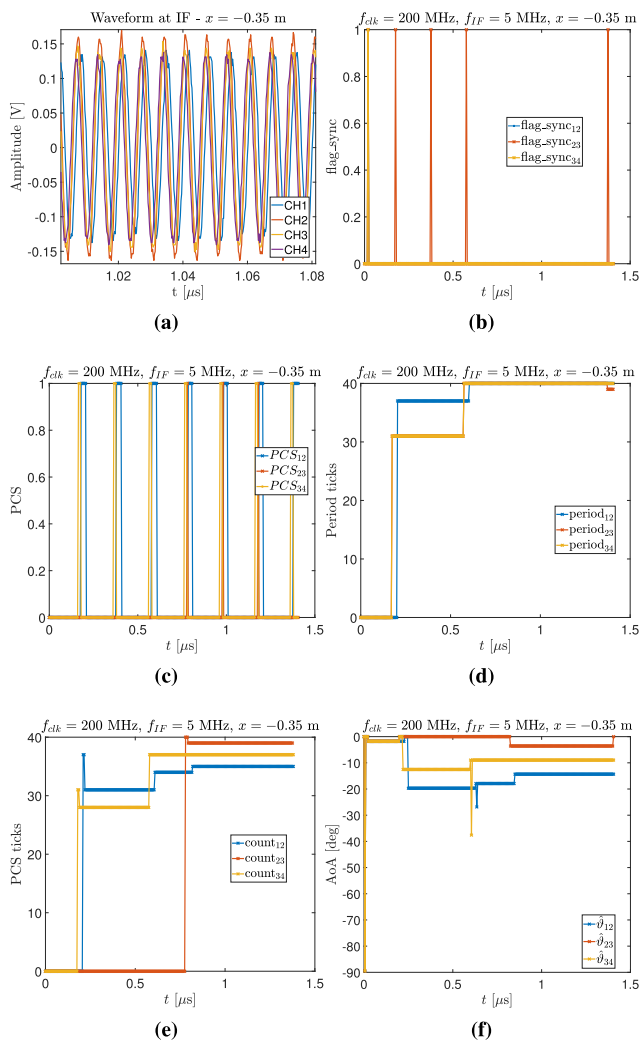


Fig. 16. Example of the output signals from the digital architecture: (a) Zoom on the output signal from the digital oscilloscope (b) Synchronization flag (c) PCS (d) estimated period ticks count (e) estimated PCS ticks count (f) AoA output from the LUT.

antenna has a 25 deg -3dB Half-Power Beam Width (HPBW) that may lead to some positions to have the transmitter being in a null or a sidelobe or with a lower SNR than the other pairs. Hence, by averaging, we take into account even this possibility.

As already discussed, we compared our results with the MATLAB implementations of the MUSIC, RMUSIC and ESPRIT algorithms [34]. The sampling frequency was set equal to  $f_s = 5$  GS/s. For the autocorrelation matrix computation, we used the MATLAB built-in functions `musicdoa`, `rootmusicdoa` and `espritdoa`. Finally, for a more fair comparison of the results, the angle was quantized to be represented with the same precision as the ones in the digital hardware LUT.

### C. Comparison Metrics

To evaluate the performances in terms of precision, we employed two error metrics, i.e., the absolute error (also referred to as Absolute Estimation Error, AEE) and the Root Mean Square Error (RMSE) [35]. The precision of each

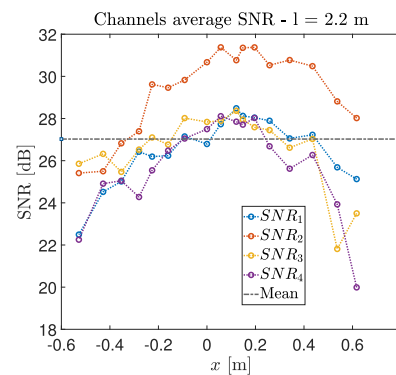


Fig. 17. Average SNR for each position and channel.

experiment is computed with respect to the three considered algorithms.

1) *Absolute Estimation Error*: The absolute error is computed for each position by considering the average estimation over the available acquisitions performed on the channels. Recalling there are  $P = 20$  acquisitions available:

$$err_{mean}(x) = \frac{1}{P} \sum_{p=1}^P |\hat{\vartheta}_p(x) - \vartheta(x)| \quad (23)$$

with

$$\hat{\vartheta}_p(x) = \frac{1}{|\mathbf{C}|} \sum_{i \in \mathbf{C}} \hat{\vartheta}_i^p(x) \quad (24)$$

where  $\mathbf{C}$  denotes the set of antenna pairs,  $\mathbf{C} = \{12\}, \{23\}, \{34\}$ , and  $\hat{\vartheta}_i^p(x)$  denotes the estimation made through antenna pair  $i \in \mathbf{C}$  when employing the sample  $p$  in position  $x$ .

The AEEs for the algorithms employed is computed as

$$err_{\alpha}(x) = \frac{1}{P} \sum_{p=1}^P |\hat{\vartheta}^{\alpha,p}(x) - \vartheta(x)| \quad (25)$$

where  $\hat{\vartheta}_p^{\alpha}(x)$  denotes the estimation made by employing the sample  $p$  in position  $x$  using the algorithm  $\alpha \in \mathcal{A}$ , with  $\mathcal{A}$  being the set of algorithms.

---

#### Algorithm 2 Squaring Block Model Requiring High and Low Voltage Thresholds $V_H, V_L$ and Discrete Time Input Signal $s$

---

**Input:**  $V_H, V_L, s(k-1), s(k)$   
 $\Delta = s(k) - s(k-1);$   
**if**  $s(k) < V_L \wedge \Delta < 0$  **then**  
  |  $y(k) = 0$   
**else if**  $s(k) \geq V_H \wedge \Delta \geq 0$  **then**  
  |  $y(k) = 1$   
**else if**  $s(k) < V_H \vee s(k) \geq V_L$  **then**  
  | **if**  $\Delta \geq 0$  **then**  
  | |  $y(k) = 0$   
  | **else**  
  | |  $y(k) = 1$   
**return**  $y(k)$

---

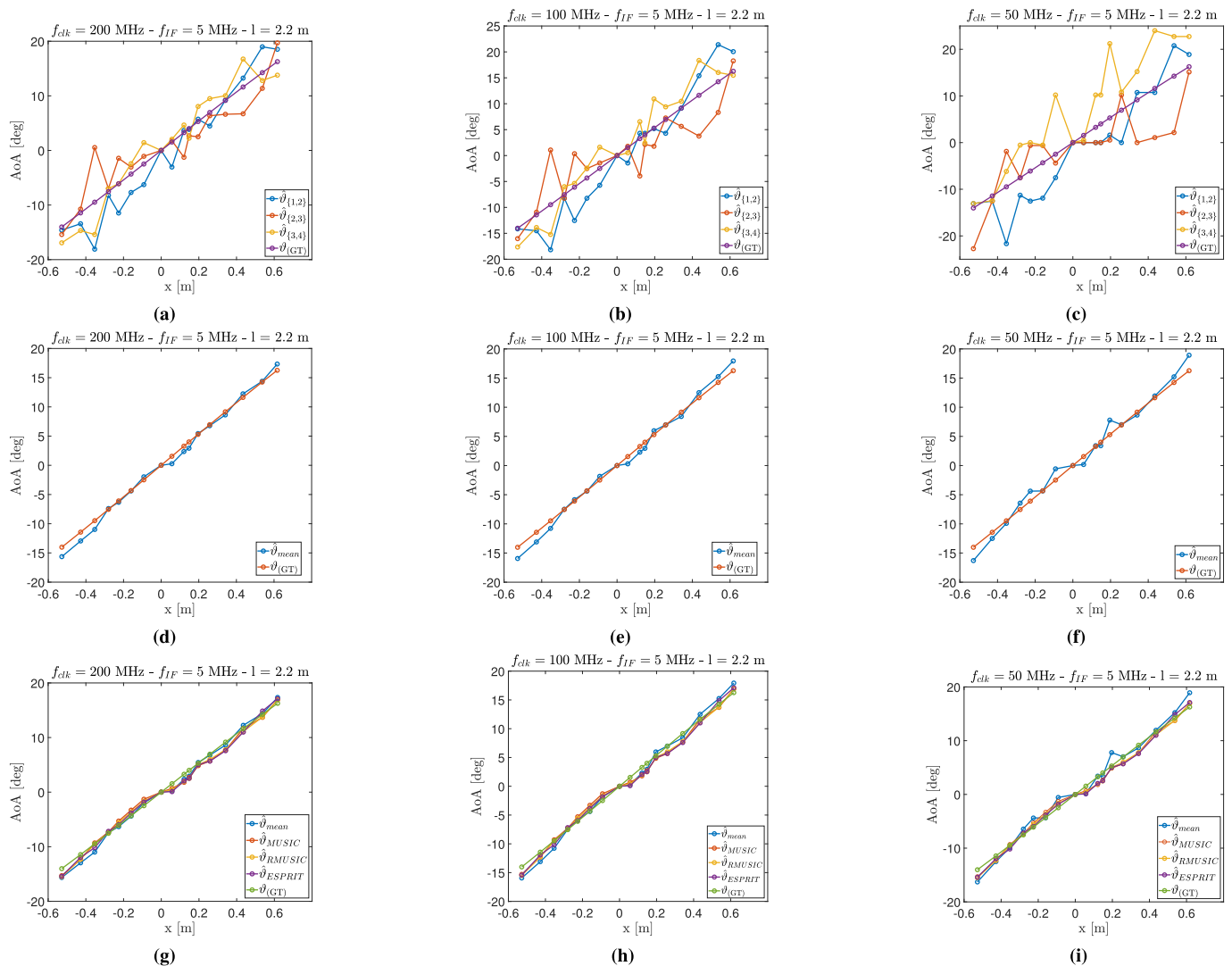


Fig. 18. Experiment results: antenna pairs estimations with (a)  $f_{clk} = 200$  MHz (b)  $f_{clk} = 100$  MHz and (c)  $f_{clk} = 50$  MHz. Figures (d)-(e)-(f) show the average estimations for the available antenna pairs at  $f_{clk} = 200$  MHz,  $f_{clk} = 100$  MHz and  $f_{clk} = 50$  MHz respectively. Figures (g)-(h)-(i) show the comparison of the mean value with the considered reference algorithms at the usual frequencies.

TABLE II

STATISTICS ON THE AEE AND ON THE RMSE FOR THE ANALYZED APPROACHES AND CLOCK FREQUENCIES. ALL THE VALUES ARE EXPRESSED IN DEGREES. (AND  $\pm\%$  ERROR VS REFERENCE)

	$f_{clk} = 200$ MHz [reference]	$f_{clk} = 100$ MHz	$f_{clk} = 50$ MHz	MUSIC	RMUSIC	ESPRIT
<b>AEE</b>						
Mean	0.67	0.83 (+24%)	1.04 (+55%)	0.79 (+18%)	0.76 (+13%)	0.80 (+19%)
Std	0.57	0.62 (+ 8%)	0.90 (+58%)	0.49 (-14%)	0.48 (-16%)	0.50 (-12%)
Max	1.61	1.92 (+19%)	2.65 (+64%)	1.47 (- 8%)	1.50 (- 7%)	1.58 (- 2%)
<b>RMSE</b>						
Mean	1.02	1.38 (+35%)	1.85 (+81%)	1.28 (+25%)	0.89 (-12%)	0.93 (- 9%)
Std	0.10	0.28 (+180%)	0.49 (+390%)	0.07 (-30%)	0.01 (-90%)	0.01 (-90%)
Max	1.22	2.23 (+83%)	3.70 (+203%)	1.23 (-0.8%)	0.92 (-24%)	0.96 (-21%)

2) *Root Mean Square Error*: We decided to adopt the RMSE as a cumulative error evaluation metric since it emphasizes the errors occurring in a much stronger way than the mean AEE. Two different kinds of RMSE were evaluated.

The first one regards the estimation made through each antenna pair by considering the incrementing number of available experimental sample repetitions for each position. We define the cumulative average operator  $\mathbb{A}(\gamma | q)$  for

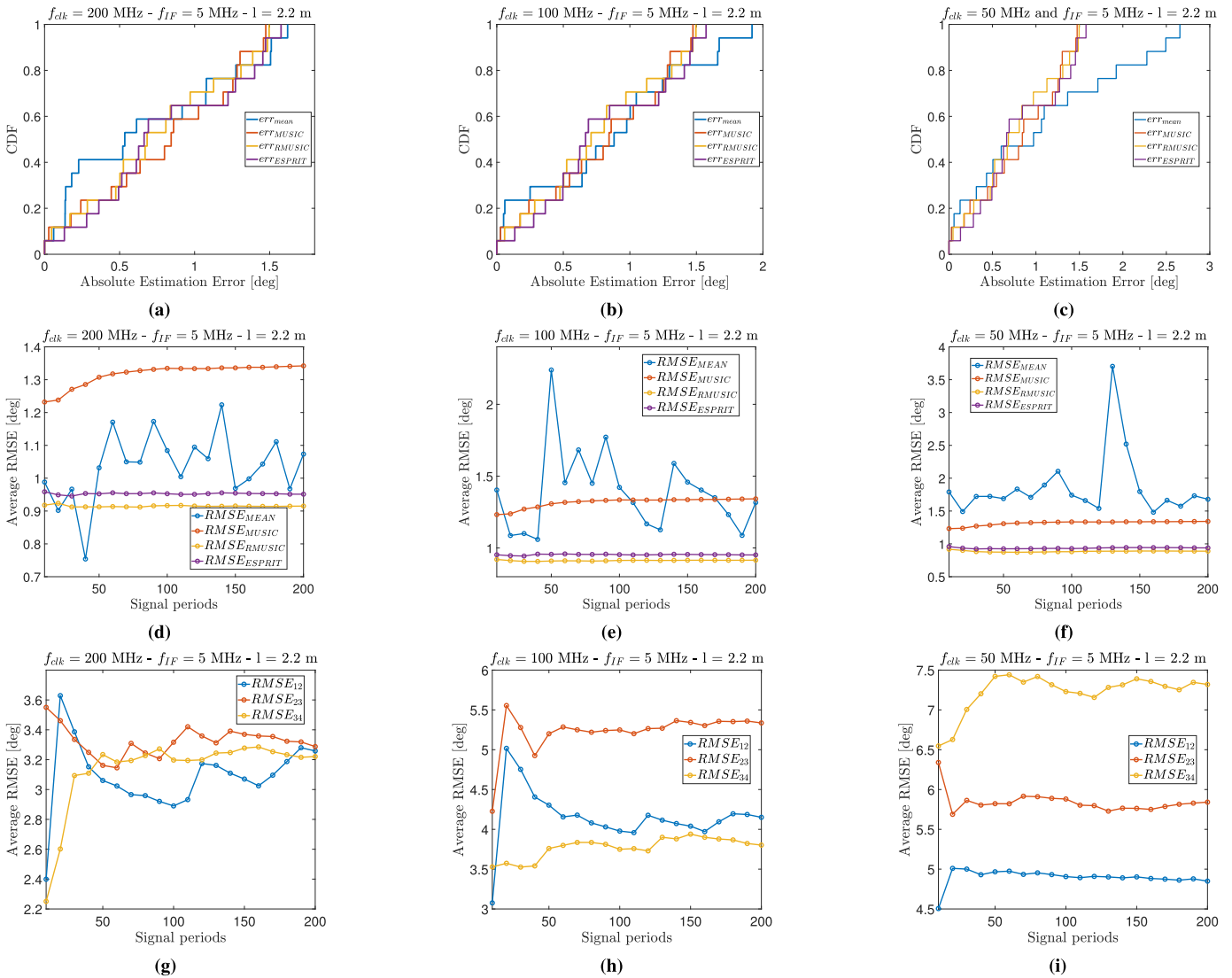


Fig. 19. Comparison of the empirical CDF for the AEE of the proposed approach and the three reference algorithms when considering (a)  $f_{clk} = 200$  MHz (b)  $f_{clk} = 100$  MHz and (c)  $f_{clk} = 50$  MHz. Figures (d)-(e)-(f) show the RMSE comparison of the proposed approach with the three algorithms for  $f_{clk} = 200$  MHz,  $f_{clk} = 100$  MHz and  $f_{clk} = 50$  MHz respectively. Figures (g)-(h)-(i) show the comparison of the RMSE for each of the three antenna pairs for  $f_{clk} = 200$  MHz,  $f_{clk} = 100$  MHz and  $f_{clk} = 50$  MHz respectively.

estimating  $\gamma$  as:

$$\mathbb{A}(\gamma | q) = \frac{1}{q} \sum_{c=1}^q \gamma(c) \quad (26)$$

with  $\gamma$  being the parameter of interest varying on the repeated measure set.

Therefore:

$$RMSE_{ij}(q) = \sqrt{\frac{1}{|\mathbf{X}|} \sum_{x \in \mathbf{X}} \left[ \mathbb{A}(\hat{\vartheta}_{ij}^q(x) | q) - \vartheta(x) \right]^2} \quad (27)$$

The second RMSE considered is the one regarding the cumulative average of the estimations made through the average AoA estimated by each antenna pair. The RMSE considering  $q$  samples is defined as:

$$RMSE_{mean}(q) = \sqrt{\frac{1}{|\mathbf{X}|} \sum_{x \in \mathbf{X}} \left[ \frac{1}{|\mathbf{C}|} \sum_{i \in \mathbf{C}} \mathbb{A}(\hat{\vartheta}_i(x) | q) - \vartheta(x) \right]^2} \quad (28)$$

with  $\mathbf{X}$  denoting the set of positions relative to the array center position. These two RMSE were compared to the ones achieved by using the three reference algorithms, taking into account the cumulative average made on the available experiment repetitions. Each experimental sample repetition contains 10 signal periods of the IF.

## V. RESULTS

The first analysis performed is the SNR evaluation of each experimental sample. Figure 17 depicts, for each position, the average SNR of all acquired samples.

As can be easily observed, the SNR for channel 2 is sensibly higher than the others. This is due to the fact nominally identical front-end prototypes can lead to different signal power levels due to realization mismatches. However, all the channels experience the same behavior, with the SNR lowering in the positions far from the array center. This is because the transmitting terminal approaches the walls, so there is an

increase of the reflected paths that destructively compose at the receiver side, and hence the SNR lowering.

Figure 18 summarizes, for each clock frequency, the results of our hardware architecture and then compares them with the results obtained for the MATLAB reference algorithms. The single antenna pair estimations tend to be less accurate in estimating the GT, especially at  $f_{clk} = 50$  MHz. As expected, when considering lower clock frequencies, the average curve tends to be less precise. However, as shown in Figure 18g, increasing clock frequency allows the proposed architecture to outperform the reference algorithms. To quantify the achieved precision, we first analyze the AAE. The results are depicted in Figure 19a-b-c as empirical Cumulative Distribution Functions (CDFs) of the AEE, for each of the three clock frequencies and for each approach under analysis. When considering lower clock frequencies the AAE error increases, but when considering the maximum available clock, the result is 13 to 19% better than the one achieved with the reference algorithms.

In addition, the proposed architecture has the major advantage of requiring a significantly lower computational complexity. The time needed for the execution of the reference algorithms is sensibly lower because of the simpler operations to perform. Moreover, as expected, for all the considered algorithms the errors are higher in those positions where the SNR was said to be lower due to higher multipath components.

Therefore, we analyze the results considering the cumulative RMSE of the mean estimation. Figure 19d-e-f summarize the results. For high clock values the RMSE again decreases, and is either very close or in the case of MUSIC about 10 to 40% lower than the references. Furthermore, the independence between the number of samples considered for the estimation and the RMSE suggests that the proposed AoA approach is actually eligible for real-time and memory-less implementations. Finally, if we look at the RMSE of each individual channel with respect to the number of samples, we note that the fluctuations observed have the same magnitude as the ones in the global RMSE.

Table II summarizes the results of the comparison between the AEE and the RMSE for the three clock frequencies realizations of the proposed approach and the three other classical algorithms. In detail, for the RMSE, the table summarizes the average values, the standard deviations and the maximum values over all the available periods.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a new architecture for AoA estimation based on phase interferometry. The architecture developed is fully digital and synchronous. The proposed solution is modular, and varying the clock frequency allows us to achieve different levels of granularity for the AoA estimation. The experimental analysis proved the capability of the approach to operate in real-time. The obtained performances confirm the possibility of using the proposed technique for implementing real-world applications where usually the classical algorithms for AoA estimation are employed.

Future work will involve the implementation of a logic conversion interface layer between the receiving front-end and the

digital architecture. In addition, we will also investigate other ways of exploiting the AoAs' multiple estimations available. For instance, it is possible to perform a weighted average of the different estimation contributions, where the weights are determined based on the radiation pattern of the antenna elements and the received signal strength level. Moreover, the proposed architecture can be integrated into an ASIC to deploy a low-cost RTLS solution, given the limited prototype area occupation.

## ACKNOWLEDGMENT

Antonello Florio would like to thank the Intel FPGA Academic Program and Terasic for the donation of boards and software licenses for developing this research project. MATLAB, Simulink, and FPGA-in-the-Loop are registered trademarks of The MathWorks Inc.

## REFERENCES

- [1] S. Zekavat et al., "An overview on position location: Past, present, future," *Int. J. Wireless Inf. Netw.*, vol. 28, no. 1, pp. 45–76, Mar. 2021.
- [2] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.
- [3] M. Yassin and E. Rachid, "A survey of positioning techniques and location based services in wireless networks," in *Proc. IEEE Int. Conf. Signal Process., Informat., Commun. Energy Syst. (SPICES)*, Feb. 2015, pp. 1–5.
- [4] G. Piccinni, G. Avitabile, G. Coviello, and C. Talarico, "Real-time distance evaluation system for wireless localization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 10, pp. 3320–3330, Oct. 2020.
- [5] M. Aernouts, N. BniLam, R. Berkvens, and M. Weyn, "TDAoA: A combination of TDoA and AoA localization with LoRaWAN," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100236.
- [6] H. Kwame and S. Ekin, "RSSI-based localization using LoRaWAN technology," *IEEE Access*, vol. 7, pp. 99856–99866, 2019.
- [7] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A survey of machine learning for indoor positioning," *IEEE Access*, vol. 8, pp. 214945–214965, 2020.
- [8] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiqzaman, and D. O. Wu, "Indoor intelligent fingerprint-based localization: Principles, approaches and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2634–2657, 4th Quart., 2020.
- [9] D. Sun et al., "A BLE indoor positioning algorithm based on weighted fingerprint feature matching using AOA and RSSI," in *Proc. 13th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2021, pp. 1–6.
- [10] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, Jul. 1996.
- [11] P. S. Farahsari, A. Farahzadi, J. Rezazadeh, and A. Bagheri, "A survey on indoor positioning systems for IoT-based applications," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7680–7699, May 2022.
- [12] S. Kutty and D. Sen, "Beamforming for millimeter wave communications: An inclusive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 949–973, 2nd Quart., 2016.
- [13] G. Avitabile, A. Florio, and G. Coviello, "Angle of arrival estimation through a full-hardware approach for adaptive beamforming," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 12, pp. 3033–3037, Dec. 2020.
- [14] I. T. Union. (2011). *Spectrum Monitoring Handbook*. [Online]. Available: <http://handle.itu.int/11.1002/pub/80399e8b-en>
- [15] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.
- [16] U. M. Butt, S. A. Khan, A. Ullah, A. Khaliq, P. Reviriego, and A. Zahir, "Towards low latency and resource-efficient FPGA implementations of the MUSIC algorithm for direction of arrival estimation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 8, pp. 3351–3362, Aug. 2021.
- [17] M. Kim, K. Ichige, and H. Arai, "Implementation of FPGA based fast DOA estimator using unitary MUSIC algorithm," in *Proc. IEEE 58th Veh. Technol. Conf. VTC*, vol. 1, Oct. 2003, pp. 213–217.

- [18] F. Wang, H.-T. Gao, L. Zhou, and Y.-X. Sun, "Hardware implementation of MUSIC algorithm for airborne digital direction finding system," in *Proc. 7th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2011, pp. 1–4.
- [19] H. Chen, K. Chen, K. Cheng, Q. Chen, Y. Fu, and L. Li, "An efficient hardware accelerator for the MUSIC algorithm," *Electronics*, vol. 8, no. 5, p. 511, May 2019.
- [20] Z. Li, W. Wang, R. Jiang, S. Ren, X. Wang, and C. Xue, "Hardware acceleration of MUSIC algorithm for sparse arrays and uniform linear arrays," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 7, pp. 2941–2954, Jul. 2022.
- [21] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 1983, pp. 336–339.
- [22] T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *Opt. Eng.*, vol. 29, no. 4, p. 296, 1990.
- [23] P. Boonyanant and S. Tan-a-ram, "FPGA implementation of a subspace tracker based on a recursive unitary ESPRIT algorithm," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2004, pp. 547–550.
- [24] Y. Jung, H. Jeon, S. Lee, and Y. Jung, "Scalable ESPRIT processor for direction-of-arrival estimation of frequency modulated continuous wave radar," *Electronics*, vol. 10, no. 6, p. 695, Mar. 2021.
- [25] S. Bottigliero, D. Milanese, M. Saccani, and R. Maggiora, "A low-cost indoor real-time locating system based on TDOA estimation of UWB pulse sequences," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
- [26] Z. Zhang, C. Zhou, Y. Gu, and Z. Shi, "FFT-based DOA estimation for coprime MIMO radar: A hardware-friendly approach," in *Proc. IEEE 23rd Int. Conf. Digit. Signal Process. (DSP)*, Nov. 2018, pp. 1–5.
- [27] D. Neunteufel, S. Grebien, and H. Arthaber, "Indoor positioning of low-cost narrowband IoT nodes: Evaluation of a TDoA approach in a retail environment," *Sensors*, vol. 22, no. 7, p. 2663, Mar. 2022.
- [28] Z. Guo, W. Wang, X. Wang, and X. Zeng, "Hardware-efficient beamspace direction-of-arrival estimator for unequal-sized subarrays," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1044–1048, Mar. 2022.
- [29] N. BniLam, S. Nasser, and M. Weyn, "Angle of arrival estimation system for LoRa technology based on phase detectors," in *Proc. 16th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2022, pp. 1–5.
- [30] X. Liu, Y. Xie, H. Chen, and B. Li, "Implementation on FPGA for CORDIC-based computation of arcsine and arccosine," in *Proc. IET Int. Radar Conf.*, Oct. 2015, pp. 1–4.
- [31] *FPGA-in-the-Loop—MATLAB & Simulink*. Accessed: Nov, 7, 2022. [Online]. Available: <https://www.mathworks.com/help/hdlverifier/fpga-in-the-loop.html>
- [32] *E4433B ESG-D Series Digital RF Signal Generator, 4 GHz, Keysight*. Accessed: Nov. 7, 2022. [Online]. Available: <https://www.keysight.com/us/en/product/E4433B/esgd-series-digital-RF-signal-generator-4-ghz.html>
- [33] *5182A MXG Vector Signal Generator, 100 Khz to 6 GHz, Keysight*. Accessed: Nov. 7, 2022. [Online]. Available: <https://www.keysight.com/us/en/product/N5182A/mxg-vector-signal-generator-100khz-6ghz.html>
- [34] *Direction of Arrival Estimation—MATLAB & Simulink—MathWorks*. Accessed: Nov. 7, 2022. [Online]. Available: <https://www.mathworks.com/help/phased/direction-of-arrival-doa-estimation-1.html>
- [35] S. M. Ross, "Parameter estimation," in *Introduction to Probability and Statistics for Engineers and Scientists*, S. M. Ross, Ed., 5th ed. Boston, MA, USA: Academic Press, 2014, ch. 7, pp. 235–296.



**Antonello Florio** (Associate Member, IEEE) was born in Bari, Italy, in 1995. He received the B.Sc. degree (Hons.) in computer and automation engineering and the M.Sc. degree (Hons.) in telecommunications engineering from the Polytechnic University of Bari, the M.Sc. degree in computer science from the University of Nice-Sophia Antipolis, France, and the Ph.D. degree in electrical and information engineering from the Polytechnic University of Bari. In 2023, he was a Visiting Expert with the Radio Frequency Division, European Space Research and Technology Centre (ESTEC), European Space Agency (ESA), The Netherlands. His research interests include phased arrays, theories and techniques for localization, and green wireless sensor networks.



circuits for telecommunication applications.

**Gianfranco Avitabile** (Senior Member, IEEE) was born in Livorno, Italy, in 1958. He received the B.S. and M.S. degrees in electronics from Florence University, Italy, in 1982. He joined the Department of Electronics, University of Florence, as an Assistant Professor. In 1998, he joined the Department of Electronics, Polytechnic University of Bari, as an Associate Professor, where he currently works. His research interests mainly include the high frequency circuits and systems for both civil and military applications and high-performance analog integrated



communications. His research interests include digital and mixed analog/digital integrated circuits and systems, computer-aided design methodologies, and the design and analysis of systems-on-chip.

**Claudio Talarico** (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Genoa, Italy, and the Ph.D. degree in electrical engineering from the University of Hawaii. He is currently a Professor in electrical and computer engineering with Gonzaga University. Before joining Gonzaga University, he was with Eastern Washington University, The University of Arizona, and with industry, where he held both engineering and management positions with Siemens Semiconductors, IKOS Systems, and Marconi Com-



**Giuseppe Coviello** (Senior Member, IEEE) was born in Bari, Italy, in 1981. He received the Laurea degree in electronic engineering and the Ph.D. degree in electric and information engineering from the Polytechnic University of Bari, Italy, in 2006 and 2015, respectively. He is currently an Assistant Professor with the Department of Electrical and Information Engineering, Polytechnic University of Bari. His research interests include RF and mixed-signal circuits and embedded systems for healthcare.