

Scalable Phase-Coherent Beam-Training for Dense Millimeter-wave Networks

Dolores Garcia^{*}, Student Member, IEEE, Jesus O. Lacruz^{*}, Pablo Jiménez Mateo^{*},
Joan Palacios^{*}, Rafael Ruiz^{*}, and Joerg Widmer^{*}, Fellow, IEEE

Abstract—mm-wave communications use analog beamforming techniques, which steer the signal energy in a desired direction, to overcome the high path-loss at such frequencies. To determine the direction in which to steer, mm-wave standards such as IEEE 802.11ad specify beam training mechanisms for both access points as well as client stations. However, the overhead of the beam training limits scalability as the density of network deployments increases and mobile devices that require constant *re-training* are supported. We design SPIDER, a low-overhead beam-training mechanism where only access points actively participate in the training and stations perform passive compressive estimation of the angle-of-arrival. To this end, stations carry out phase-coherent measurements by switching through multiple receive beam patterns on a time-scale of tens of nanoseconds when receiving a packet preamble. Since no suitable testbed platforms exist that support such fast antenna reconfiguration, we design a high-performance, full-bandwidth FPGA-based testbed platform for flexible mm-wave experimentation, that we make available as open source. The performance analysis with this testbed shows that our algorithm achieves highly accurate angle estimation used to drive the beam steering decisions and reduces overhead by an order of magnitude compared to IEEE 802.11ad beam training.

Index Terms—Millimeter Wave, Beamtraining, Compressive Sensing, FPGA, IEEE 802.11ad, Phase-Coherent, Phased Antenna Array

1 INTRODUCTION

MILLIMETER wave (mm-wave) network technology such as the IEEE 802.11ad [2] and the upcoming IEEE 802.11ay [3] standards for 60 GHz Wireless Local Area Networks (WLANs) can provide vastly higher data rates than networks operating below 6 GHz, due to the large bandwidth available at mm-wave frequencies. Thanks to the small wavelength, mm-wave devices typically integrate phased antenna arrays enabling analog beamforming. The beamforming gains of the antenna compensate for the higher path loss due to the smaller effective capture area of the antenna elements [4]. The narrow beam patterns that are possible with analog beamforming steer the signal in a desired direction. This not only ensures a good link range but also reduces interference and thus improves spatial reuse. It introduces new challenges to establish (and maintain) a high data rate communication link, since the main lobes of the phased arrays of the communicating devices need to always be aligned, pointing towards the strongest multipath components of the channel.

Beam training mechanisms are critical for efficient communication at mm-wave frequencies. With IEEE 802.11ad [2] beam training, one device transmits frames using each available antenna configuration, while a target device listens with a quasi-omnidirectional beam pattern. This avoids the complexity of training both ends simultaneously. The target

device then provides feedback with the index of the antenna configuration that offers the highest Signal-to-Noise Ratio (SNR) [5]. IEEE 802.11ad also provides the option of a subsequent beam refinement to quickly probe several (narrower) antenna sectors to further improve the beam alignment. Such brute force or simple hierarchical beam training works well for relatively static deployments of moderate density. However, the overhead may become prohibitive in more dense and dynamic scenarios.

Wireless access point and client densities are expected to increase significantly, partly to achieve higher data rates [6] and serve a higher number of connected devices per user, and partly due to new communication scenarios such as Internet-of-Things and Machine-Type Communication [7]. While current IoT networks are typically low rate, future use cases for example for Industry 4.0 networks and the Internet of Vehicles are expected to involve massive amounts of data with very strict latency requirements to transmit high definition video, sensor information from autonomous vehicles, and support image-guided systems [8, 9]. Mm-wave technology is highly appealing for such future IoT networks, since the latency is lower than that of sub-6GHz systems, the ultra high bandwidth enables the data rates required by such demanding applications, and directional communication enables high levels of spatial reuse for many concurrent devices [10, 11]. Its low interference footprint makes mm-wave technology extremely well suited for such dense deployments. It is also part of 5G New Radio Access Networks, where mobility plays a major role. In such use cases, beam training is triggered much more frequently than in simple static scenarios, incurring excessive training overhead and reducing the efficiency of the communication.

Mechanisms that better handle network dynamics may only probe sectors adjacent to the current sector [12, 13]

• D. Garcia, P. Jiménez Mateo and J. Palacios are with IMDEA Networks Institute and Universidad Carlos III de Madrid, Spain.

E-mail: {dolores.garcia,pablo.jimenez,joan.palacios}@imdea.org

• J.O. Lacruz, R. Ruiz and J. Widmer are with IMDEA Networks Institute, Madrid, Spain.

E-mail: {jesusomar.lacruz,rafael.ruiz,joerg.widmer}@imdea.org

* Co-Primary Authors

An earlier version of this work appeared in the ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'20) [1]

or other sectors predicted to work well [14,15], or update the beam direction using out-of-band information [16, 17, 18, 19]. More complex hierarchical approaches [20, 21, 22] further improve beam training performance and in the ideal case their overhead scales logarithmically with the number of antenna sectors. However, they usually require a large codebook and several feedback rounds which introduce additional delay. Phase-coherent [23, 24] or non-coherent [25, 26, 27] compressive beam training exploits the sparsity of the mm-wave channel [28, 29] and achieve similar overhead using only a single feedback round. Some of these schemes also cope well with imperfect beam shapes with side lobes. Of these, only the non-coherent works from [25, 26, 27] have been evaluated in real scenarios. Even better performance can be achieved by estimating the sparse multi-path channel with a constant overhead per multi-path component [30]. However, channel estimation from such few measurements requires solving a complex non-convex optimization problem using a genetic algorithm.

Besides handling network dynamics, another important aspect is the scalability of beam training with respect to the number of nodes in the network. A high node density significantly increases training overhead (especially in combination with high network dynamics). While an Access Point (AP) can beam train with multiple stations simultaneously (i.e., its beam training messages are received by all of its stations), the stations have to train one at a time, given that current APs only have a single RF chain and cannot concurrently receive from multiple stations. The beam training overhead thus scales with the sum of the number of APs and stations in the network. Even worse, whenever a station has to connect to a new AP due to mobility or blockage, it potentially needs to beam train with several or all of the APs in its vicinity, since only *after* beam training does the station know which AP provides the best link quality. If the number of candidate APs increases with the AP density, the beam training overhead scales with the product of the number of APs and stations in the network. All of the above mentioned beam training algorithms have in common that *both APs and stations actively participate in the beam training*. Given that networks usually have far more stations than APs, the station beam training overhead is critical for scalability.

In this paper we design and implement a beam training mechanism that *eliminates station beam training overhead* entirely, so that the overhead scales only with the number of APs and the network remains efficient even for very high node densities and high mobility scenarios. Our mechanism for Scalable Phase-Coherent Beam-Training for Dense Millimeter-wave Networks (SPIDER) is able to work with unmodified standard-compliant IEEE 802.11ad APs that send out a beacon frames via each of their antenna sectors. A station implementing SPIDER starts receiving those frames with a quasi-omnidirectional antenna configuration like a normal station would. However, as soon as a station detects the start of a valid frame during the preamble reception, the station starts to rapidly switch between different receive antenna configurations on a time-scale of tens of nanoseconds. The duration of this training takes only a small fraction of the preamble without compromising the actual frame decoding. Using the measurements taken with different receive beam patterns, we can extract accurate angle informa-

tion of the received signal. Based on the estimated Angle of Arrival (AoA) to the AP, the station then determines which of its own beam patterns to use for the current channel and also gives feedback to the AP to indicate which AP beam pattern provided the highest SNR. Beam switching on a per symbol level is a prerequisite for 802.11ad beam refinement and for *in-packet* training mechanisms in 802.11ay systems [31] and current devices already support this [32]. Our mechanism requires only minor changes in the signal processing at the stations. In contrast to prior work based on Received Signal Strength (RSS) [1], we use phase coherent measurements taken for the different beam patterns over a single preamble, which enable much more accurate angle estimation. Our phase-coherent approach can also estimate the full multiple-path channel, whereas the approach in [1] only provides the single strongest path. Furthermore, we design a simple method to measure the normalized array response of the beam patterns, which is a prerequisite to implement SPIDER. With this design, legacy stations operating in the network are not affected. The approach is also compatible with compressive sector selection at the AP, where the AP trains only a (small) subset of sectors to further reduce beam training overhead. Finally, the design may be applicable in future 5G and 6G networks at sub-THz frequencies that use single-carrier modulation with sufficiently long correctable sequences for channel estimation [33].

For a practical implementation of SPIDER, we design a mm-wave experimentation platform which integrates a highly configurable FPGA baseband processor connected to a 60 GHz RF front-end with a 16+16 element phased antenna array. The testbed is able to transmit, receive and decode custom as well as standard-compliant IEEE 802.11ad/ay frames thanks to its multi giga-sampling rate capability. To support beam training research, the testbed includes real-time packet detection and fast antenna reconfiguration blocks that can reconfigure the antenna on a per symbol-level. The code and specifications of our testbed are made available as open source to the research community [34].

We test the SPIDER station with off-the-shelf APs to evaluate its performance in different static as well as mobile scenarios. We show that our mechanism can switch through 10 receive beam patterns during a single preamble, without compromising the detection accuracy or the frame decoding for typical SNR values. Only 3 beam pattern changes are required to accurately estimate the angles of the multi-path components of the channel. We also compare the performance of SPIDER to standard 802.11ad beam training and to our previous work [1], which performs this estimation based solely on RSS measurements. We further include experiments that demonstrate the multi-path estimation capabilities, beam training with simultaneous AP and station mobility, and more complex Line-of-Sight (LOS)/Non-Line-of-Sight (NLOS) trajectories. SPIDER achieves a median angle estimation error of 0.5° for static scenarios and 0.7° for mobile scenarios which leads to very accurate beam training decisions. Furthermore, we achieve a reduction in overhead by a factor of 8.8 compared to standard 802.11ad beam training with 7 stations (measured) and a factor of 25 for 64 stations (extrapolated).

2 BACKGROUND

We first revisit 802.11ad beam training and preamble decoding, which are prerequisites for our design.

2.1 IEEE 802.11ad beam training

To establish a directional link in mm-wave networks, IEEE 802.11ad devices implement a beam training mechanism as shown in Fig. 1a. First, the AP performs a so-called Sector Level Sweep (SLS), by sending beacon messages via each of its antenna sectors [5], while the station is listening using a quasi-omnidirectional beam pattern. Then, the station performs its own SLS, sending sector sweep messages to the AP via its different antenna sectors. These messages include the ID of the AP sector received with the highest SNR. Finally, the AP replies with a Sector Sweep Feedback that includes the ID of the strongest station sector, and the station concludes with a Sector Sweep ACK. At this point, both AP and station are aware of the best sector to use for communication. Optionally, an additional Beam Refinement Phase (BRP) can be carried out after the SLS, which allows to quickly probe several sectors over the course of a single packet and may be done with narrower antenna beams than during SLS to improve the link quality.

2.2 IEEE 802.11ad preamble processing

The IEEE 802.11ad standard specifies different physical layer modes: control, single carrier, OFDM and low-power single carrier. However, only the control and single carrier modes are supported by current devices. The frame structure shown in Fig. 2 is shared by all modes, with a preamble composed of a Short Training Field (STF) and Channel Estimation Field (CEF), a header containing information about the Modulation Coding Scheme (MCS), data length, and other information, the data itself, and an optional Beamforming Training (BFT) field.

IEEE 802.11ad beam training uses control frames sent with the lowest MCS 0 (BPSK modulation with 1/2-rate LDPC coding and a spreading factor of 32), which makes them very resilient to noise. The STF part of the preamble is used for frame detection, symbol synchronization, Carrier Frequency Offset (CFO) estimation and compensation, and coarse synchronization, whereas the CEF is used to estimate the Channel Impulse Response (CIR) for the equalization blocks and for fine frame synchronization. The STF of control frames consists of 48 repetitions of Golay sequences of length 128 $G_{b_{128}}$, followed by a single $-G_{b_{128}}$ and $-G_{a_{128}}$ to detect the end of the STF [2]. The STF duration is 3.63 μ s. Frame detection is usually performed through Normalized Auto Correlation (NAC), which avoids having to store the Golay sequences in memory and simplifies hardware implementation [1, 35, 36]:

$$\text{NAC} = \frac{\left| \sum_{n=0}^{127} [\mathbf{y}]_{i-n} [\mathbf{y}]_{i-n-127}^* \right|}{\left| \sum_{n=0}^{127} [\mathbf{y}]_{i-n} [\mathbf{y}]_{i-n}^* \right|}, \quad (1)$$

where \mathbf{y} is the vector of received IQ samples. A frame is detected when the NAC output exceeds a configurable threshold, which is adjusted to trade-off sensitivity and false detection rate. The CFO estimation is based on the

phase of the auto-correlation (the numerator in (1)). To reduce noise and refine the estimate, phase measurements are accumulated over multiple Golay sequences. In practice, accumulation over 7-8 Golay sequences (510 to 580 ns) is enough for a good estimate [37]. There are efficient hardware implementations for the CFO estimation and correction using the CORDIC algorithm [38]) and Direct Digital Frequency Synthesizers (DDFSs) [39], respectively. Symbol synchronization can be performed using multiple parallel interpolation filters over a discrete number of phase values [35], which reduce the processing time thanks to the parallelization. Finally, the $-G_{b_{128}}$ sequence at the end of the STF incurs a phase change in the correlation and serves for boundary detection (coarse synchronization). The CIR is computed in the CEF of the preamble. The CEF is composed of complementary Golay sequences $G_{a_{128}}$ and $G_{b_{128}}$, which have the property that their auto-correlations sum up to a 128 element vector with only one non-zero entry. This facilitates channel estimation by providing an unbiased, error-minimizing CIR estimator.

3 ALGORITHM DESIGN

The main motivation for our algorithm is to reduce the beam training overhead in highly dynamic and dense environments. To this end, we design a single phase beam training mechanism as depicted in Fig. 1b. Only the AP performs SLS, and *at the same time*, all stations are trained simultaneously by means of phase-coherent measurements during the preambles of the received AP frames, taking advantage of the sparse nature of the mm-wave channel [28, 29] for compressed estimation.

3.1 SPIDER overview

During the SLS, an AP sends N beacon frames, each via a different transmit sector, while the stations listen with a quasi-omnidirectional receive pattern. As soon as a station detects a frame, it starts switching through M ($M \ll N$) different receive beam patterns while the preamble is being received. The received I/Q samples are phase coherent for the measurements over a single preamble. Moreover, each of the M sequences of samples has different amplification and phase offset only due to the different receive beam patterns. With the complex gain information of the different receive beam patterns, we can then estimate the AoA of the signal arriving from the AP for each of the AP's beacon frames that is detected. We then estimate the full multi-path channel based on efficient dimensionality reduction methods and joint estimation over all received beacon frames.

Since the receiver switches through different beam patterns only *after* frame detection (i.e., the preamble has served its purpose), our method does not affect the detection probability. However, it is important that the beam reconfigurations end before the end of the preamble, to not compromise the decoding of the rest of the frame. To ensure that this is the case, below we analyze the preamble processing tasks and in Section 5 we also validate this experimentally. As explained in Section 2.2, the 48 Golay sequences $G_{b_{128}}$ forming the STF of control frames last 3.49 μ s and the frame detection takes around 1000 samples ($\sim 0.56 \mu$ s) for normal

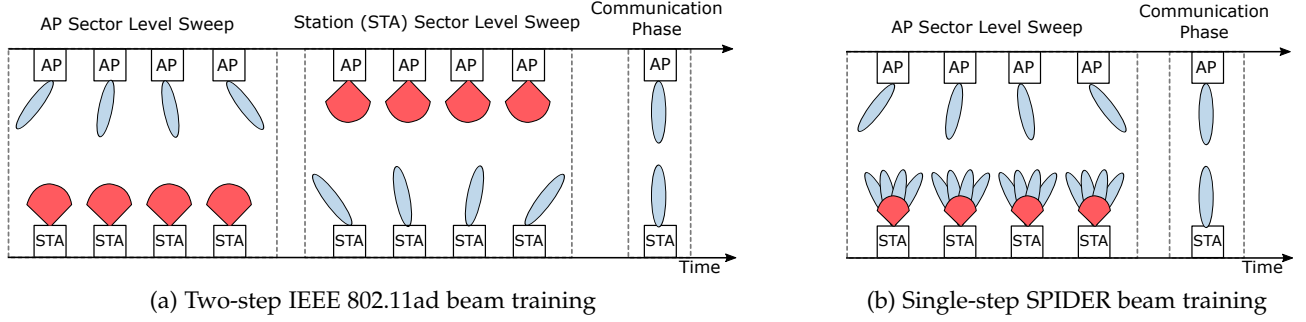


Figure 1: Comparison between IEEE 802.11ad beam training and the proposed SPIDER algorithm

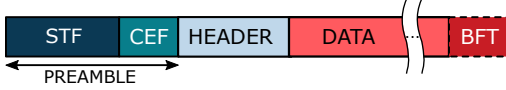


Figure 2: IEEE 802.11ad frame structure

SNR conditions (see Section 5). Besides, CFO estimation can be performed accurately within $0.58 \mu\text{s}$ and the symbol synchronization using parallel interpolation filters can be executed in $0.15 \mu\text{s}$. Finally, the boundary detection can be performed over the last 500 samples of the STF ($0.29 \mu\text{s}$) to not compromise the CEF and header decoding. This leaves $\sim 1.91 \mu\text{s}$ which can be used for additional tasks to be performed over the preamble. In this paper, we consider an upper bound of 10 beam pattern changes, each lasting 72.7 ns which matches the duration of a single Gb_{128} sequence). We thus require 727 ns or $\sim 20\%$ of the STF duration to perform the 10 reconfigurations which is more than enough for accurate AoA estimation, as shown in Section 5.¹

Fig. 3a provides an example of a received IEEE 802.11ad control frame, where 10 antenna reconfigurations have been made over the preamble of the frame. The figure also includes the other fields of the control frame. Fig. 3b is a zoom into the preamble from Fig. 3a, to show the amplitude pattern for the 10 different antenna configurations. As can be seen from the graph, switching from one beam pattern to the next is extremely fast and only affects very few IQ samples taken during the phase shifter reconfiguration.

3.2 Channel and transceiver model

To derive the algorithm to process the phase-coherent measurements, we use a wide band geometrical channel model with additive paths, each with a specific AoA, Angle of Departure (AoD), complex gain and time of arrival. We use the notation x , \mathbf{x} , \mathbf{X} for scalar, vector and matrix, respectively. $[x]_a$ denotes the a th element of a vector, \mathbf{x}^H and \mathbf{X}^H denote the Hermitian of a vector and a matrix, x^* , \mathbf{x}^* are the complex conjugate, and $\mathcal{R}e(x)$ is the real part of x . $|x|$ and $\|\mathbf{x}\|$ are the absolute value and Euclidean norm and $\mathbf{x}_1 * \mathbf{x}_2$ is the circular cross correlation with the fast Fourier implementation $\text{ifft}(\text{fft}(\mathbf{x}_1) \otimes \text{fft}(\mathbf{x}_2)^*)$, where fft and ifft are the fast Fourier transform and its inverse.

1. The number of reconfigurations is limited by the antenna switching speed and the duration of a Golay sequence. Phased arrays with switching speed of a few ns have been proposed [40], and shorter sequences such as Gb_{64} can be used for channel estimation, although they may lead to a noisier estimation.

The signal $\mathbf{y}_{r,t}$ received with beam-patterns $\mathbf{c}_r \in \mathcal{C}^{K_T,1}$ for reception and $\mathbf{p}_t \in \mathcal{C}^{K_R,1}$ for transmission is given by

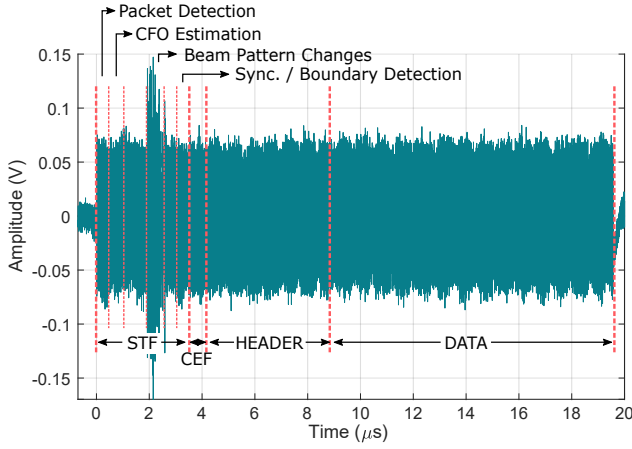
$$\mathbf{y}_{r,t} = \sum_{l=1}^L \alpha_l \mathbf{c}_r^H \mathbf{a}_{\text{Rx}}(\theta_l) \mathbf{a}_{\text{Tx}}^H(\phi_l) \mathbf{p}_t \mathbf{x} * \delta_{\tau_{l,t}} + \mathbf{n}, \quad (2)$$

where $K_{T/R}$ is the number of antenna elements at the transmitter (T) and receiver (R) arrays. The transmitter emits the signal \mathbf{x} that arrives at the transmit antennas with the phase shifts of the selected beam-pattern \mathbf{p}_t and goes into the channel at angle ϕ_l , i.e., it is multiplied by the steering vector $\mathbf{a}_{\text{Tx}}(\phi_l) \in \mathcal{C}^{K_T,1}$. These two effects combined are commonly called array factor $\mathbf{a}_{\text{Tx}}^H(\phi_l) \mathbf{p}_t$. The channel incurs a complex gain α_l . At the receiver, the signal reaches the antennas at angle θ_l and is multiplied by the steering vector $\mathbf{a}_{\text{Rx}}(\theta_l) \in \mathcal{C}^{K_R,1}$. It enters the device via the selected reception beam pattern with weights \mathbf{c}_r . The signal is also shifted in time according to the time of arrival $\tau_{l,t}$, which is measured from the moment the first reception beam pattern is selected. This shift depends on the packet detector delay and this delay may be different for different beam patterns. Therefore, $\tau_{l,t}$ is not the absolute time of arrival of the LOS path, but the time differences of arrival of the paths are constant and correspond to the different multi-path delays, i.e., $\tau_{l_1,t_1} - \tau_{l_2,t_1} = \tau_{l_1,t_2} - \tau_{l_2,t_2}$. The time shift is expressed by a convolution $*$ (in its continuous version) with the time response vector of delay $\tau_{l,t}$, labeled $\delta_{\tau_{l,t}}$. As paths interact, the received signal is the sum of the ones received through each of the paths. The receive antennas also capture additive white complex Gaussian noise \mathbf{W} with variance σ^2 via the receive beam-pattern \mathbf{c}_r , which gives $\mathbf{n} = \mathbf{c}_r^H \mathbf{W}$. Since the beam patterns are normalized to $\|\mathbf{c}_r\| = 1$, even though the dimension of \mathbf{n} differs from that of \mathbf{W} , its distribution remains unchanged. Note that the noise is independent and different for each r, t pair.

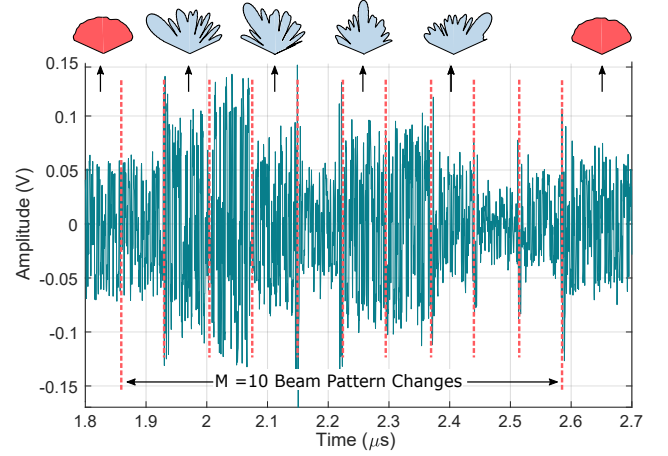
The objective of our algorithm is to estimate the angle of arrival θ_l of each path. The direction of the dominant path yields the estimated AP direction and is used to determine the best station beam pattern.

3.3 Fast channel estimation

Our channel estimation has to be performed using the samples from the STF of IEEE 802.11ad control frames, i.e., solely using Gb_{128} Golay sequences. Therefore, classic estimation techniques that take advantage of the correlation properties of complementary Golay sequences Ga_{128} and Gb_{128} cannot be used [41]. This makes channel estimation



(a) Signal processing tasks performed over the preamble of a control frame.



(b) Detailed beam pattern changes during the preamble of the control frame in (a).

Figure 3: Received IEEE 802.11ad control frame with multiple antenna reconfigurations performed during the preamble.

more challenging and we present a novel channel estimation algorithm that works well with Gb₁₂₈ Golay sequences.

We first discuss a simple greedy algorithm [42] to estimate the dominant path from the receiver measurements (2). This algorithm is iterative, and in the first iteration estimates the parameters of the dominant path $l = 1$ under the assumption that the received signal $\mathbf{y}_{r,t}$ is the result of only a single path. We then subtract this path from the original signal, to obtain the information of the remaining multi-path components and continue iterating to obtain all multi-path components, until the noise level is reached.

For simplicity, we drop the index l in (2) and group the effects not related to the receiver in a variable $\beta_t = \alpha \mathbf{a}_{\text{Tx}}(\phi)^H \mathbf{p}_t \in \mathcal{C}$ and the ones corresponding to the receiving radiation properties in $b_r(\theta) = \mathbf{c}_r^H \mathbf{a}_{\text{Rx}}(\theta)$ to obtain

$$\mathbf{y}_{r,t} = \beta_t b_r(\theta) \mathbf{x} * \delta_{\tau_t} + \mathbf{n}. \quad (3)$$

Our problem consists of inferring the channel parameters β_t, θ, τ_t that best fit the measured signal $\mathbf{y}_{r,t}$. Since noise is Gaussian we can formulate a minimum mean square error problem (MMSE)

$$\min_{\beta_t, \theta, \tau_t} \sum_{r,t} \|\mathbf{y}_{r,t} - \beta_t b_r(\theta) \mathbf{x} * \delta_{\tau_t}\|^2. \quad (4)$$

We obtain a problem in three dimensions, t, r , and the length of \mathbf{x} , that is hard to solve since decomposition methods are typically designed for two dimensions. At a high level, to solve (4) we find an expression for β and then reduce the problem to that of comparing the entries of a matrix to obtain θ and τ_t . Since the signal \mathbf{x} contains only Gb₁₂₈ Golay sequences instead of complementary pairs, we cannot use conventional channel estimation. Instead, we obtain a solution by manipulating the expression in (4) to reduce its dimensions from equations (5) to (6) and perform a joint optimization over all receiver beam patterns in (7). Finally, by solving the minimization problem for β_t in (8), we transform (7) into a two-step element comparison problem, with (9) for the first step over the time of arrival and (10) for the second step over the angle.

Concretely, we develop the expression (4) and eliminate the constant terms to obtain the equivalent problem with reduced dimension

$$\min \sum_{r,t} |\beta_t|^2 |b_r(\theta)|^2 - 2\text{Re}(\beta_t b_r(\theta) (\mathbf{x} * \delta_{\tau_t}) \mathbf{y}_{r,t}^H). \quad (5)$$

This exchanges the vector norm by two scalar terms, of which the right one matches the angle radiation pattern with the measurements, while the left one acts as a regularization term. It converts the MMSE problem into an easier regularized matching problem. Since the correlation in the second term of (5) is acting on \mathbf{x} but not on $\mathbf{y}_{r,t}$, we simplify the problem to have a single equation in time.

We apply the circular cross correlation property $(\mathbf{x} * \delta_{\tau_t}) \mathbf{y}_{r,t}^H = \langle \mathbf{y}_{r,t}, \mathbf{x} * \delta_{\tau_t} \rangle = \langle \mathbf{y}_{r,t} * \mathbf{x}, \delta_{\tau_t} \rangle = (\mathbf{y}_{r,t} * \mathbf{x})^*(\tau_t)$. Note that the convolution result is often used as a CIR estimator. This result provides a formula depending on a CIR estimation $(\mathbf{y}_{r,t} * \mathbf{x}) \in \mathcal{C}^{P,1}$ as

$$\min \sum_{r,t} |\beta_t|^2 |b_r(\theta)|^2 - 2\text{Re}(\beta_t b_r(\theta) (\mathbf{y}_{r,t} * \mathbf{x})^*(\tau_t)), \quad (6)$$

where P is the number of points in the inverse Fourier transform. We provide an example of such CIRs in Fig. 4, measured in our testbed from Gb₁₂₈ sequences over a single preamble, for several different receive beam patterns and a fixed transmit sector.

We group the sum over the different receiver beam patterns r and obtain the following expression

$$\min \sum_t |\beta_t|^2 \|\mathbf{b}(\theta)\|^2 - 2\text{Re}(\beta_t \mathbf{z}_t^H(\tau_t) \mathbf{b}(\theta)), \quad (7)$$

where $\mathbf{z}_t(\tau_t) \in \mathcal{C}^{M,1}$, $\mathbf{b}(\theta) \in \mathcal{C}^{M,1}$ are the vertical concatenations over the index r of $(\mathbf{y}_{r,t} * \mathbf{x})(\tau_t)$, $b_{\text{Rx},r}(\theta)$. We can now solve the minimization problem for β_t , since it spans only over term t .

$$\beta_t = \frac{(\mathbf{z}_t^H(\tau_t) \mathbf{b}(\theta))^*}{\|\mathbf{b}(\theta)\|^2}. \quad (8)$$

Inserting the result in (7), the minimization problem becomes equivalent to the maximization of

$$\max \sum_t |\mathbf{z}_t^H(\tau_t) \bar{\mathbf{b}}(\theta)|^2, \quad \text{s.t. } \bar{\mathbf{b}}(\theta) = \frac{\mathbf{b}(\theta)}{\|\mathbf{b}(\theta)\|}. \quad (9)$$

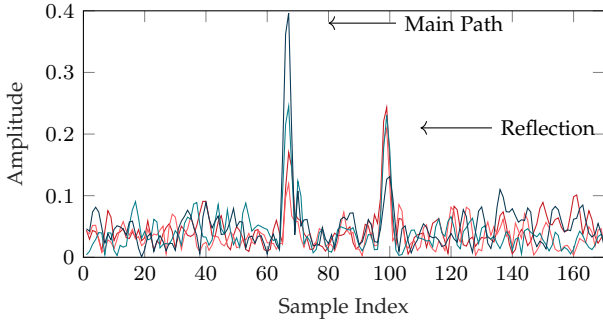


Figure 4: Estimated CIR $\mathbf{y}_{r,t} * \mathbf{x}$ at the receiver for different receive beam patterns (shown in different colors)

Given the normalized array response per beam-pattern $\bar{\mathbf{b}}(\theta)$ (measured as discussed in Section 5.1), only θ and τ_t are left to be computed. We estimate θ by correlating against the known complex gains of the receiver beam patterns on a grid of K values, where K is given by the resolution of our angle estimation as explained in section 5.1. Denote by $\mathbf{B} = [\bar{\mathbf{b}}(\theta_1), \dots, \bar{\mathbf{b}}(\theta_K)] \in \mathcal{C}^{M,K}$ the matrix containing the complex gain information of all receive beam patterns in the directions $\theta_1, \dots, \theta_K$ and let $\mathbf{Z} \in \mathcal{C}^{M,N}$ denote the matrix in which the row $\mathbf{z}_t = [\mathbf{z}_t(1), \dots, \mathbf{z}_t(N)] = [(\mathbf{y}_{1,t} * \mathbf{x})^T, \dots, (\mathbf{y}_{R,t} * \mathbf{x})^T]^T$ represents the estimated CIR received through the r^{th} beam pattern. Maximizing over τ_t in a joint estimation over all transmitter sectors t

$$\max_t \sum_k \max_{\tau_t} |[\mathbf{Z}^H \mathbf{B}]_{\tau_t, k}|^2, \quad (10)$$

we estimate θ as the θ_k for which k maximizes (10).

In order to reconstruct the dominant path, we need to obtain τ_t and β_t as follows. τ_t can be retrieved as the one maximizing $|[\mathbf{Z}^H \mathbf{B}]_{\tau_t, \hat{k}}|^2$ and β_t is given by (8) in terms of $\|\mathbf{b}(\hat{\theta})\|$ as $\beta_t = [\mathbf{Z}^H \mathbf{B}]_{\tau_t, \hat{k}}^* / \|\mathbf{b}(\hat{\theta})\|$.

3.4 Multi-path extension

We can extend this procedure to extract all the multi-path components of the channel. While not strictly necessary for beam training, it is of interest for path ranking, for example to deal with blockage of the dominant path.

Once the parameters of the dominant path are estimated, it is possible to reconstruct it as

$$\hat{\mathbf{y}}_{r,t} = [\mathbf{Z}^H \mathbf{B}]_{\tau_t, \hat{k}}^* [\bar{\mathbf{b}}(\hat{\theta})]_r \mathbf{x} * \delta_{\hat{\tau}_t}, \quad (11)$$

and subtract its contribution to the received signal $\mathbf{y}_{r,t}$

$$\mathbf{y}'_{r,t} = \mathbf{y}_{r,t} - \hat{\mathbf{y}}_{r,t}. \quad (12)$$

$\mathbf{y}'_{r,t}$ contains the information of the remaining multi-path components. We repeat the process given in (6) for the new path variables $\tau'_t, \theta', \beta'_t$ with the signal $\mathbf{y}'_{r,t}$ in the problem

$$\sum_{r,t} |\beta'_t|^2 |b_r(\theta')|^2 - 2\text{Re}(\beta'_t b_r(\theta') (\mathbf{y}'_{r,t} * \mathbf{x})^* (\tau'_t)), \quad (13)$$

until we extract all visible paths from the measurements.

4 TESTBED DESIGN

One of the important contributions of this paper is to provide a flexible testbed for full-bandwidth real-time mm-wave experimentation (*mm-FLEX*). This not only allows to experimentally validate our SPIDER mechanism, but more importantly fills an important gap in available testbed platforms for the research community. Although Commercial-Off-The-Shelf (COTS) mm-wave hardware includes functionality to quickly change the antenna configuration, the fact that this functionality cannot be accessed and the beam training implementation itself cannot be modified severely limits the usefulness of such devices for beam training research. To the best of the authors knowledge, prior to the proposed mm-FLEX design [1], there were no experimentation platforms that offered the ability of fast antenna reconfiguration. The whole design and all required software is made available as open source [34].

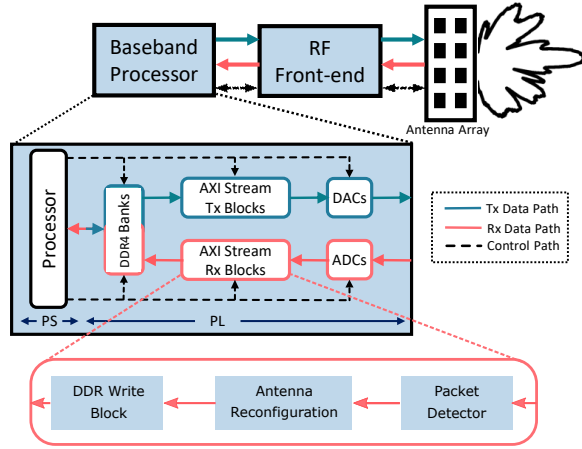
We now present the testbed architecture and features, and show how it can be used to implement mm-wave protocols and specifically beam training algorithms.

4.1 Hardware components

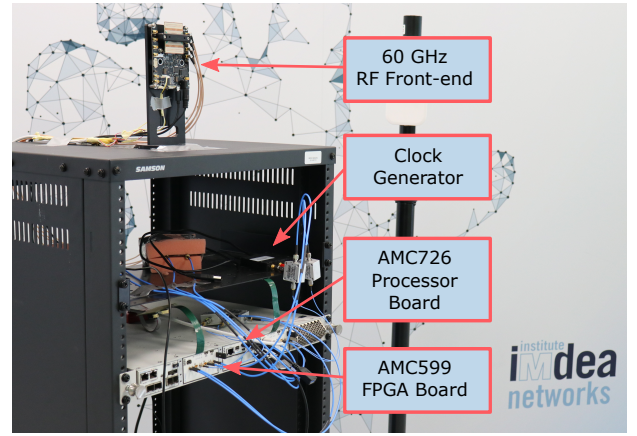
The architecture follows the premise of using a limited number of readily available hardware components, to make it easy to build the platform. The proposed system is composed of two main components, i) a powerful baseband processor that can generate and capture baseband IQ samples at multiple Giga-sampling rates from a ii) 60 GHz RF front-end integrating phased array antennas with analog beam-forming capabilities, which performs direct up/down conversion from/to the baseband samples, as shown in Fig. 5b. Fig. 5a provides a high-level block diagram of the experimentation platform. While we describe the experimentation platform with 60 GHz RF front-end for IEEE 802.11ad/ay research, the baseband system supports different front-ends, such as 28 GHz transceivers for 5G NR systems [43].

The *baseband processor* is built around a Vadatech chassis which is able to host multiple slot cards. Specifically, we use an i) AMC599 board, integrating a Xilinx Kintex Ultrascale FPGA, high-speed AD/DA converters and DDR4 memory banks, among other features [44] and ii) an AMC726 board with an Intel Core i7 processor [45] which is connected to the FPGA card through the chassis' back-plane. The AMC599 board provides the Programmable Logic (PL) to implement the hardware blocks for processing the IQ samples that are sent to and captured from the converters and stored in the on-board DDR memory. The AMC726 board provides the Processing System (PS), serving as a management and configuration processor for the FPGA, AD/DA converters and RF front-end. The PL and PS subsystems communicate through a PCIe interface. The system can be extended to a MIMO configuration by connecting additional AMC599 boards managed from the same PS, e.g., to support the IEEE 802.11ay standard [3].

For the *RF front-end* we use the EVK06002 development kit from Sivers IMA [46] with 60 GHz up/down converters. It supports the frequencies and bandwidth of IEEE 802.11ad/ay channels and includes a 16+16 (Tx/Rx) element phased array arranged in two linear array structures. The kit provides 2D analog beam forming capabilities through



(a) High-level architecture of the proposed testbed, showing transmit and receive data paths and the control path



(b) Photo of the experimentation platform indicating the main hardware components

Figure 5: Top-level hardware architecture of the proposed testbed

phase shifters with 6 bit resolution for each antenna element. The kit can be controlled via three different interfaces: USB, SPI and GPIO pulses. We use the USB interface (connected to the AMC726 processor) to configure the kit as transmitter or receiver (at runtime), transfer the codebook, set the antenna sector, and configure the GPIO control interface. Given that those tasks are not time-critical, the latency introduced by the USB interface does not affect the performance of our system. In addition, the kit incorporates fast beam switching capabilities through simple pulses via a GPIO interface, that allow to switch beam patterns every 10 ns. The kit has a maximum RF settling time of 35 ns when changing beam patterns. The GPIO functionality is as follows: i) one GPIO input is used to increment the pointer for the active sector in the stored codebook for each received pulse; ii) a second input is used to return to a predefined sector, which can be freely configured from the USB interface. The front-end supports codebooks containing 64 different antenna sectors. The standard codebook consists of a quasi-omnidirectional beam pattern and 63 directional ones with main lobes at angles ranging from -45° to 45° . The upper part of Fig. 3b shows examples of the predefined beam patterns. The codebook can be freely configured, allowing the user to design custom beam patterns as needed.

Finally, an external dual channel clock board is used to generate *independent* sampling clock signals for the AD/DA converters (i.e., we do not require that the communicating transceivers are clock synchronized). The system specifications far exceed the bandwidth and processing requirements for an IEEE 802.11ad compliant transceiver and offer enough flexibility to be used for more powerful future mm-wave communication standards.

4.2 Basic functionality

To ensure modularity, simplify extension of the design, facilitate block interconnection and avoid inter-block synchronization issues, we adopt AXI-streams as the interface for both data paths (Fig. 5a) to move the IQ samples through the signal processing blocks. To be able to process IQ samples at giga-sampling rates with an FPGA device, it is necessary

to adopt a Super Sample Rate (SSR) design. This allow to process multiple IQ samples in parallel to stay within the clock frequency range of the PL. Here, we adopt an SSR of 16 parallel samples for full-bandwidth operation. This parameter can be configured depending on the specific design requirements, in favor of area reduction or timing constraint relaxation. While the testbed can be used with a wide range of sampling frequencies to support different standard requirements, in the remainder of this paper the sampling frequency for the DACs/ADCs is set to 3.52 Giga-Samples per Second (GSPS), to support 1.76 Gsymbols/s at 2 samples/symbol, meeting the symbol rate requirements of the IEEE 802.11ad/ay standards. This requires a clock frequency of 220 MHz for both data paths, considering an SSR of 16.

For communication between the PL and the PS, we design a PCI-to-AXI IP block that enables fast large data transfers from/to the on-board DDR memory, as required due to the Giga sampling rates. Access to the DDR memory from the PS and PL is done by means of custom-designed DDR write and DDR read hardware blocks. These efficient write/read operations to/from the on-board memory match the SSR adopted for the AXI-stream data-width in the design. Specifically, the DDR write core is user configurable from the PS to operate in two different modes: i) *simple writing*, where the user directly triggers a configurable number of IQ samples to be stored in memory in a single capture and ii) *burst saving*, where the user specifies an external trigger (e.g., the signal coming from a packet detector) to capture a configurable number of IQ samples. Using this basic functionality, the system can be operated in a hardware-in-the-loop mode, streaming IQ samples directly from DDR memory to the DACs, and writing the IQ samples from the ADCs directly to DDR memory.

4.3 Real-time enhancement

While the system can be extended to a full real-time transceiver, to keep it flexible we implement only the necessary functionality in hardware, whereas packet encoding and decoding is done in software. To support real-time beam

training algorithms, we design packet detector and antenna reconfiguration blocks that match the SSR factor and clock frequency of the system. They include AXI-stream wrappers for easy integration in the datapath (Fig. 5a).

- The **packet detector** block detects the periodic Golay sequences used in the STF. It implements the NAC algorithm given in Eq. (1). In our implementation, we rearrange the terms to avoid division and simplify the hardware implementation [37]. A valid frame is detected when the NAC exceeds a programmable threshold (configurable by the user at run-time from the PS). To improve the robustness and reduce the amount of false packet detection, we add a stage that counts the number of cycles during which the signal exceeds the threshold. This introduces a small additional latency in the detection. The output of the packet detector serves as trigger for other IP blocks. We use it in conjunction with the DDR write block to only write IQ samples that belong to valid frames to the DDR memory.

- The **antenna reconfiguration** block enables fast real-time beam steering of the antenna array from the PL by means of GPIO pulses. These are sent to the antenna controller to perform rapid phase-shifters reconfiguration. The block implements a state machine which continuously waits for a trigger signal to generate the corresponding pulses to the RF front-end controller that change the beam-pattern. This trigger can be sent from the PS or from other hardware blocks in the PL (e.g., the packet detector output). Both the number of beam-pattern changes per input trigger as well as the separation between the pulses can be configured. For our phase-coherent beam training algorithm, we configure the block to perform 10 beam pattern changes with a spacing of 72.7ns (matching the duration of a Gb₁₂₈ Golay sequence). The block can also be used for standard IEEE 802.11ad beam training or *in-packet* training as used in IEEE 802.11ay [3, 31].

4.4 AD/DA converter to FPGA latency

The AMC599 board uses a JESD204B interface [47] for data transfers between the AD/DA converters and the FPGA. The interface serializes input at the transmitter side to reduce the number of pins required for the link. This introduces a deterministic latency, which critically impacts our design, since we need to ensure that the beam pattern changes occur at the right point in time during the preamble.

Through measurements we determine that the latency introduced by the buffering and serialization process is 600 ns, and the overall latency is 700 ns due to the pipeline registers used in the different processing blocks on the receiver datapath to meet timing constraints in the design. All the signal processing operations needed for our algorithm have to terminate before the end of the STF, which imposes tight timing constraints on the design. The antenna reconfiguration block sends the GPIO pulses as soon as the packet detector output goes high. To minimize the impact of the JESD204B interface latency, we estimate the CFO and SNR during this delay. The CFO estimation requires 580 ns while the interface latency is 700 ns. In parallel, we estimate the SNR over the same Golay sequences used for CFO estimation, by subtracting the I/Q samples of one Golay sequence from the next, to obtain an estimate of the noise variance. This is repeated over multiple sequences

for a more accurate estimate. Both CFO and SNR are thus obtained *before* the changes in the signal due to the beam pattern reconfigurations affect the received IQ samples.

4.5 IEEE 802.11ad software implementation

We develop the necessary functions to generate IEEE 802.11ad compliant frames for the different MCS (Single Carrier and Control PHY) as well as the formatting functions to load them into the testbed. Similarly, on the receiver side we include the necessary software functions to separate the received IQ samples into frames (based on the packet detector flag from the hardware implementation), and do preamble, header, and payload decoding. This allows to extract the MAC address of the AP, which is used in the evaluation section to identify the source of the beacon frames in multi-APs scenarios. Of course the system can also transmit and receive frames in a custom frame format defined by the user. In this way, the user is able to generate and decode frames with our experimentation platform in a *hardware-in-the-loop* manner. The functionality is implemented offline in software, and is based on the WLAN toolbox from MATLAB [48] as well as custom functions.

5 EVALUATION

In this section, we first present the procedure to measure the normalized array response of the receive beam patterns which is needed for SPIDER. We then experimentally validate that our design does not compromise frame decoding. Finally, we present a series of experiments under different conditions to validate the performance of our phase-coherent algorithm in realistic scenarios.

5.1 Measuring the normalized antenna array response

To measure the normalized array response of the beam patterns, we mount the phased array on a motorized pan/tilt platform that mechanically steers over the whole azimuth. We time the movement of the rotation table to wait for a full SLS before moving to the next angle. We record $n = 10$ measurements of the signal for each receive beam pattern r and transmit beam pattern t , respectively, obtaining

$$\mathbf{Y}_{t,n} = \beta_{t,n} \mathbf{b}(\theta) \mathbf{x} * \delta_{\tau_{t,n}} + \mathbf{N}, \quad (14)$$

where θ is the AoA determined by the pan/tilt platform, and $\mathbf{Y}_{t,n}, \mathbf{N}$ is the vertical stack of the different $\mathbf{y}_{r,t,n}$ and \mathbf{n} , respectively, over r . The rotation table advances in steps of 0.5° , giving approximately 0.5° maximum resolution in our angle estimation. With the obtained measurements, we use the following expression to obtain the normalized array response of the beam patterns

$$\mathbb{E}(\mathbf{Y}_{t,n} \mathbf{Y}_{t,n}^H). \quad (15)$$

Since the shape of (14) is such that $\mathbf{Y}_{t,n}$ is expressed as a rank one matrix plus additive white Gaussian noise, the maximum eigenvector of (15) converges to the normalized version of the rank deficient matrix's left vector $\mathbf{b}(\theta)$, the normalized array response of the receive beam patterns.

While the 60 GHz front-end allows to freely configure the codebook, in this paper we use the standard set of pre-defined beam patterns, consisting of a quasi-omnidirectional beam pattern and 63 directional ones with main lobes at angles ranging from -45° to 45° .

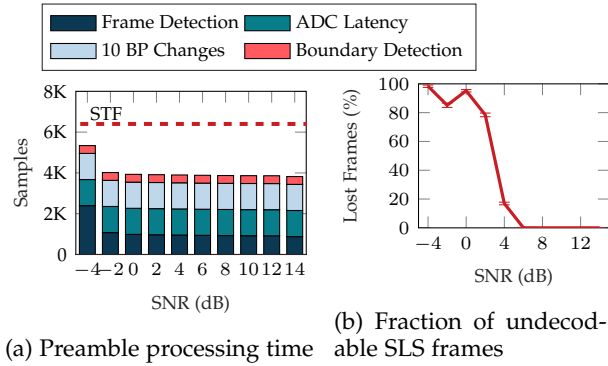


Figure 6: Frame detection for different SNR values

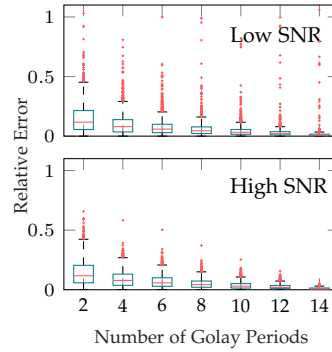


Figure 7: CFO analysis

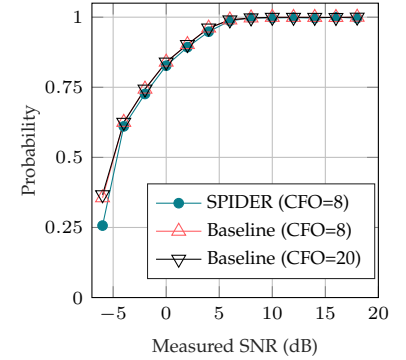


Figure 8: Decoding probability

5.2 Preliminary measurements

We now analyze how SPIDER affects the normal operation of an IEEE 802.11ad receiver. It is critical that the sequence of beam pattern changes of SPIDER terminates before the end of the STF in the preamble, to not compromise packet decoding. Therefore, frame detection has to occur sufficiently early during the preamble. Detection time primarily depends on the SNR. In Fig. 6a we stack the delay for frame detection, the ADC-to-FPGA latency, the time required for 10 receive beam pattern changes, as well as the time for boundary detection. We also include a line at 6400 samples corresponding to the duration of the STF. We observe that for SNR values from 14 to -4 dB the frame detection delay indeed increases, but for all SNR values packet detection occurs early enough for SPIDER to work. Moreover, as shown in Fig. 6b, for SNR values below 2 dB, most of the frames cannot be correctly detected. SLS frames are most robust than data frames, so that data communication at these SNR levels would be impossible. Thus, SPIDER does not affect frame decoding.

In Section 4.4 we decided to estimate the CFO in the time between our receiver sent the GPIO pulses to the antenna controller and their effect appear in the received signal. We present in Fig. 7 an analysis on how the number of Golay repetitions used to compute the CFO, affects the accuracy of the estimation. To this end, we use the designed testbed to capture SLS from a COTS IEEE 802.11ad router in different positions and distances. We classify the measurements in low SNR (< 10 dB) and high SNR (> 10 dB). As can be seen from Fig. 7, the median relative error of the CFO estimation considering 8 Golay sequences is 0.04 for both SNR scenarios. Furthermore, the difference between selecting 8 or 10 Golay sequences for the estimation is less than 0.01 for both cases, which allow to confirm that no significant deviation on the CFO estimation is introduced using 8 Golay sequences compared with 10 sequences or more. The relative error for this experiment was computed using as reference the CFO estimation performed using 20 Golay Sequences.

As we stated in Section 3.1, it is very important that the multiple antenna reconfigurations do not compromise the frame decoding. To this end, it is important that the beam pattern changes occurs during the STF without compromising the other signal processing tasks, as explained in Section 3.1. We confirm this fact by means of comparing the de-

coding probability between the decoder implementing the multiple antenna reconfigurations over the preamble and the one without implementing SPIDER (baseline). For this experiment, the number of antenna reconfigurations was set to $M = 10$ and the number of Golay sequences used to compute the CFO was set to 8. The baseline was implemented using 8 and 20 Golay sequences for the CFO estimation to account for this effect in the decoder behaviour. In all cases we capture the SLS frames from a COTS IEEE 802.11ad AP and decode them offline using a software-based decoder. We consider a decoding failure if any parity check equation of the payload “fails”. The results of both experiments are presented in Fig. 8, where we can see that regarding the CFO effect for the baseline approach, the results perfectly match for all SNR values except for $\text{SNR} \leq -6$ dB where the decoding probability is 1% lower. When implementing SPIDER, the decoder behaviour match for SNR higher than 5 dB, while the difference in the decoding probability is lower than 2% for SNR higher than -6 dB. Finally, only for SNRs lower than -6 dB, the difference in the decoding probability is 10%. From these results we can conclude that the implementation of SPIDER over the preamble of IEEE 802.11ad control packets does not compromise the decoding performance over a wide range of SNR conditions.

5.3 Accuracy

We evaluate the accuracy of SPIDER under different conditions: i) we study the impact of the number of reception beam pattern changes (M) during the preamble and ii) the number of sectors in the AP SLS used for the AoA estimation. The specific reception beam patterns are chosen heuristically using a structural similarity metric [49] to select sectors that well cover the maximum angular range. For this experiment we capture packets from a single AP. The station is mounted on the pan/tilt platform and rotates between -45° and 45° in steps of 2° . For each angle, we take static measurements comprising 20 SLSs. We compare the phase-coherent estimation of SPIDER with an estimation based solely on RSS measurements [1, 26].

Fig. 9 shows the results for different receive beam patterns. It includes whisker plots with median, quartiles, extremes (± 1.5 IQR) and outliers. SPIDER is able to determine the direction of the AP with a median error of 0.5° with only 3 beam patterns (with one single outlier, and no outliers for more than 3 patterns), as shown in Fig. 9 (top). In com-

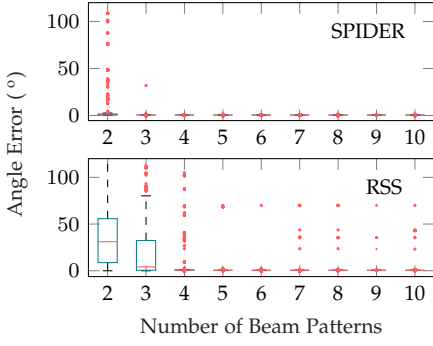
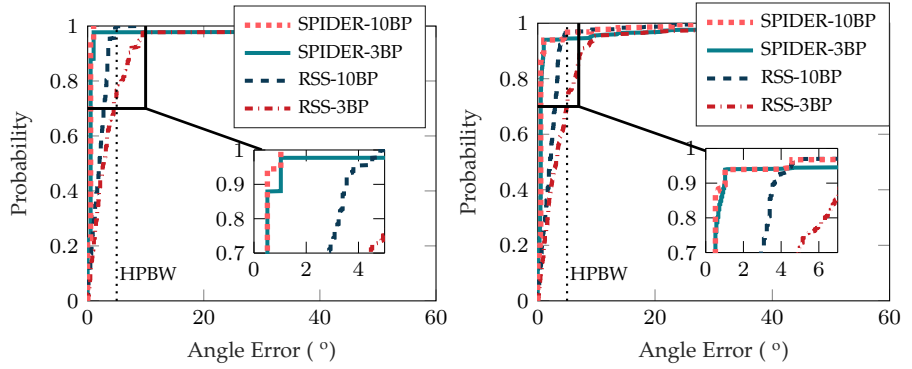


Figure 9: Angle error CDF for different number of reception beam patterns for SPIDER (top) and RSS (bottom)



(a) CDF for AP SLS with 32 sectors (b) CDF for AP SLS with 14 sectors

Figure 10: CDF of angle error for the different methods

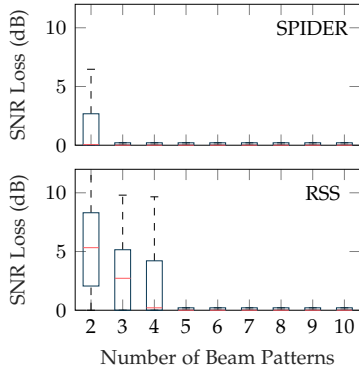


Figure 11: SNR-loss of SPIDER (top) and the RSS-based method (bottom) compared to IEEE 802.11ad beam training

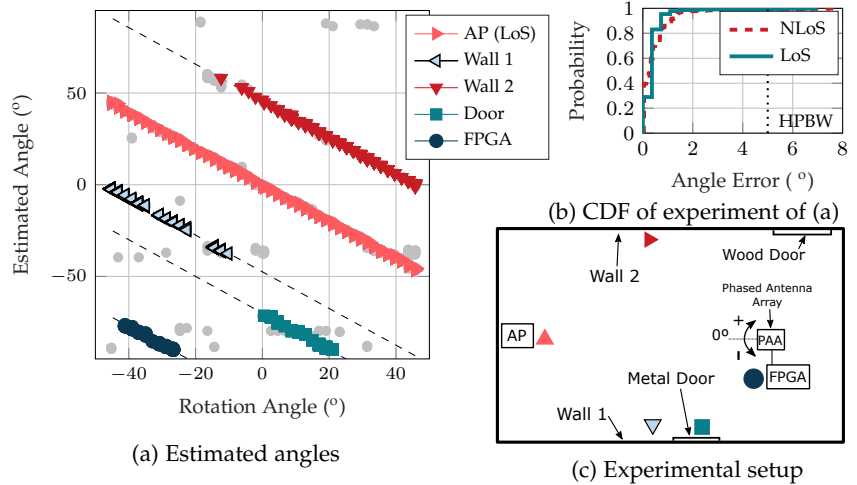


Figure 12: Multi-path angle estimation

parison, the RSS-based method without phase information (bottom graph) requires 4 or more beam patterns for a sufficiently low mean error, but even for a larger number of beam patterns has remaining estimation errors.

Fig. 10a shows the Cumulative Distribution Function (CDF) of the AoA estimation error when the station receives 32 SLS frames from the AP, the standard number of beam training frames sent by the COTS APs we use. We compare SPIDER and RSS approaches for $M = 10$ and $M = 3$ beam pattern changes. For SPIDER with $M = 10$, the CDF curve is well below 5° and thus within the half power beam width of the antenna ($\sim 10^\circ$). The algorithm achieves less than 1° error for 97% of the cases and a median angle estimation error of 0.5° . Therefore, the selected beam pattern for communication would always be the correct one. For $M = 3$, SPIDER's angle error exceeds the half power beam width only in 2.3% of the cases and the selected beam pattern for communication would (temporarily) be sub-optimal. This level of accuracy is more than enough for good link performance. In contrast, the RSS-based approach has significantly more outliers and higher angle errors, highlighting the importance of phase coherent estimation. We also consider estimation using a lower number of SLS frames, as would be the case with compressive beam training at the AP side to reduce overhead [26]. As shown in Fig. 10b, with 14 SLS frames the angle error degrades but SPIDER still works well.

For both cases $M = 10$ and $M = 3$, the mean error remains at 0.5° and the errors exceed 5° only in 3.3% and 5.5% of the cases, respectively.

Whenever the estimated angle leads to the selection of a sub-optimal beam pattern, the SNR of the link will decrease by factor corresponding to the difference of gains of the optimum versus the chosen antenna pattern at the true angle. We compare the SNR loss of SPIDER and the RSS-based approach with 10 receive beam patterns to standard 802.11ad beam training, which tests all beam patterns and thus would always select the optimal one. To determine SNR, we generate traffic from the AP to the station using *iperf*. After angle estimation using the respective algorithms, we configure the station to receive the data packets using the beam pattern that provides the highest gain in the estimated direction. We then decode the data packets to determine the SNR.² The SNR loss of SPIDER and RSS compared to exhaustive search is shown in Fig. 11. The mean SNR loss for the phase-coherent approach is zero for any number of receiver beam patterns, and has no non-zero outliers for 3 beam patterns or more. The RSS approach matches the results of our proposal only for 5 beam patterns or more.

² Since our FPGA implementation is not yet fully real-time capable and frames are decoded offline, the ACK frames for the data packets from the AP are sent by COTS station located next to the FPGA.

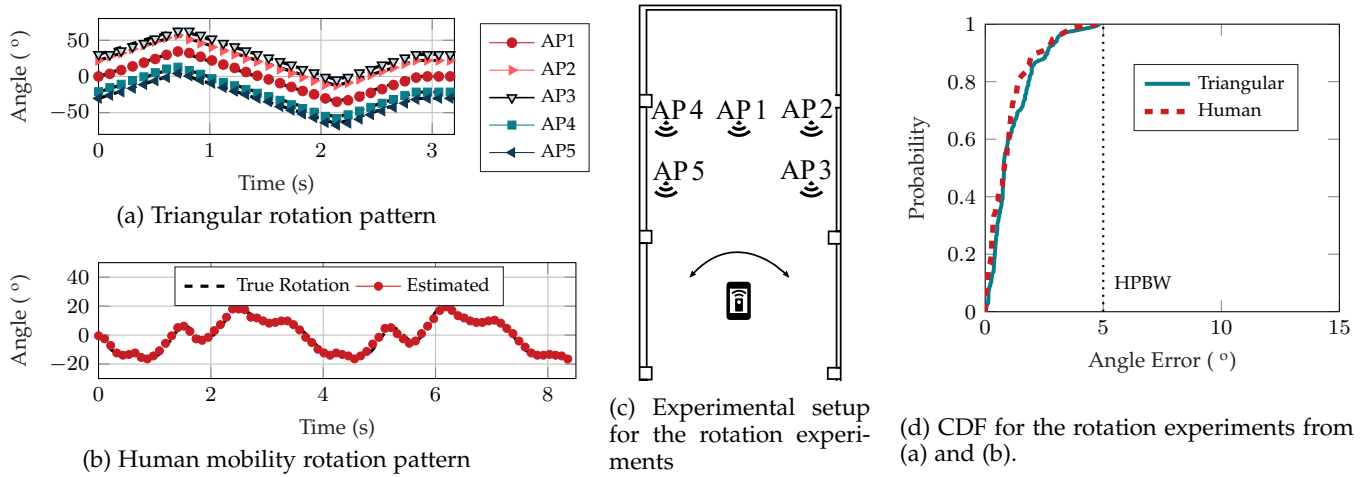


Figure 13: Real-time rotation experiments for different patterns

5.4 Multi-path estimation

As stated in Section 3.4, our phase-coherent algorithm can be extended to iteratively estimate all the multi-path components of the channel. To this end, we set up an AP to station link as described in the previous section, with the phased antenna array mounted on the pan/tilt platform. The system is deployed in an empty room as shown in Fig. 12.b, with the AP located 10 m away from the receiver. Fig. 12 shows that for the different rotation angles, it is possible to identify not only the LOS path from the AP but also reflections that follow a clear pattern for the different multi-path components present in the environment. We are able to identify four sources of such reflections, coming from the walls, a metallic door and the rack in which the FPGA is mounted (Fig. 5b). We also detect some spurious reflections, as can be seen in Fig. 12.a, that likely correspond to paths from the metallic window frames and similar room features, but we were not able to clearly identify the reflectors.

5.5 Mobility

To verify the angle estimation accuracy of SPIDER in real-time, we perform both tests with rotation at a fixed location as well as several trajectories with actual mobility.

5.5.1 Rotation experiments

For these experiments we use a setup where the station is static but rotates, to be able to tightly control the angles. The phased antenna array connected to the FPGA system is mounted on a pan/tilt platform. For this experiment we use 10 receive beam pattern changes. We deploy 5 APs in the room to show that our system can reliably estimate angles of multiple APs at the same time, as depicted in Fig. 13c. We use different APs to show the compatibility with any IEEE 802.11ad device: one Talon AD7200 router (AP1 at a distance of 6 m from the station), three Dell D5000 Wireless Docking stations (AP2 at 6.5 m, AP3 and AP5 at 4.7 m), and a NETGEAR Nighthawk X10 (AP4 at 6.5 m).

First, the receiver is steered following a triangular angle pattern sweeping to angles of 35° and -35° and back to 0° . The complete cycle of the movement takes 2.8 seconds. After that, the system remains with the original orientation,

pointing towards AP1 at 0° . As can be seen from Fig. 13a, the station is able to follow the triangular angle rotation pattern for the 5 APs even at the extreme angles. We also include a CDF of the angle error for the 5 APs in Fig. 13d. The median error is 0.8° and all of the values are below the HPBW of the antenna array. Given that our pan/tilt system and FPGA setup are not perfectly time synchronized and 1° error corresponds to only a 18 ms time shift, these results are again within the measurement accuracy of our setup.

We perform an additional rotation test for a single AP-station pair (AP1 from Fig. 13c). In this case, the station is rotated following a pattern given by a gyroscope trace of a walking person [50], while the AP is in a fixed position. The results are presented in Fig. 13b, showing that the station is able to follow the trajectory over the whole duration of the experiment (8.3 s). As can be seen from the CDF shown in Fig. 13d, the results are similar to the triangular pattern case despite the *randomness* of the trajectory, achieving 0.8° of median error and with all errors below the HPBW.

5.5.2 Simultaneous AP and station mobility

We perform experiments in a smaller laboratory of 7.5×6.5 m, where windows and office furniture (chairs, desks, computers, etc.) create a richer multi-path environment. We test the performance of the system for the extreme case of both AP and station mobility. While the station is rotating following the human walking pattern trace [50], the AP is moving towards the direction of the station from P1 to P2 as shown in Fig. 14c. The results in Fig. 14a show that the angle pattern is similar to the one with only rotation (Fig. 13b) but with an additional angle increase due to the AP movement. Even with the additional challenge of movement at both sides of the link, the CDF of the angle error shows a median error of 0.67° , with the 96% of values below 2.4° and a single outlier value beyond the HPBW of the antenna array.

5.5.3 Multi-AP trajectory

We also perform experiments with real-time human mobility in which the station is moved manually at walking speed along a trajectory that covers part of a corridor and then goes into a laboratory, as shown in Fig. 15c. For this experiment we deploy four APs, with three of them placed

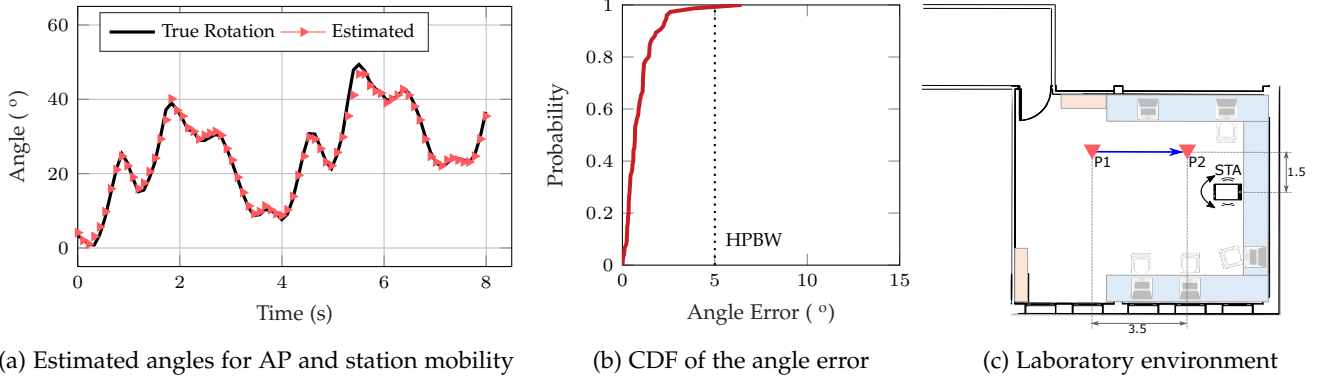


Figure 14: Simultaneous AP and station mobility in a laboratory environment

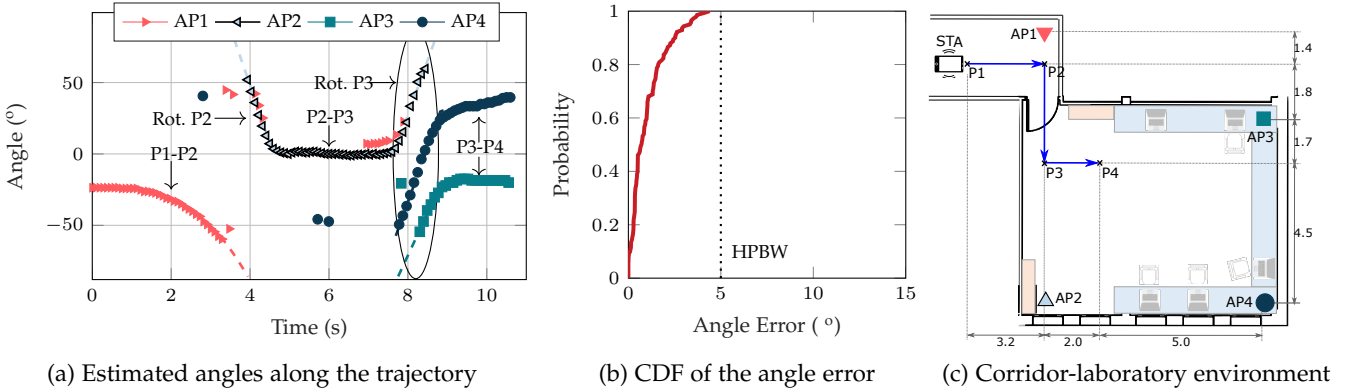


Figure 15: Multi-AP trajectory in a corridor-laboratory environment

in three of the corners of the laboratory while the fourth one is located outside in the corridor. Due to the beam width and the different orientations of the station, different APs are observed during different parts of the trajectory, as shown in Fig. 15a. We include in Fig. 15a the measured ground truth angles for each AP as dashed lines. During the movement from P1 to P2, the station is only able to locate AP1 (in the corridor) until it is beyond -60° . In this part of the trajectory, the station is also briefly able to detect AP4 (bottom-right corner in Fig. 15c) through the open door.

Once the station reaches P2, it rotates by 90° . It starts receiving frames from AP2 and AP1 through NLOS reflection coming from the metal locker next to AP2. AP2 is visible at 0° during the P2-P3 trajectory and the angle changes when the station rotates at P3. At this point, it starts locating AP3 and AP4 which are then successfully tracked until the end of the trajectory. We include a CDF of the angle error corresponding to the whole trajectory in Fig. 15b. Despite the difficulties to get ground truth angle information and the possible errors introduced by vibrations from the rack wheels (Fig. 5b), we achieve 0.68° of median error and 100% of the values below the HPBW of the antenna. We tried multiple trajectories obtaining similar results, which demonstrates the accuracy of the proposed algorithm.

5.6 Beam training overhead reduction

We cannot test SPIDER in a dense deployment of stations, since we only have two FPGA transceiver nodes. However, since SPIDER is fully passive as discussed in Section 3, we

can we get a precise estimate of the overhead by means of analyzing large standard IEEE 802.11ad deployments and discounting the station overhead from such setups. The beam training frequency is vendor specific, and most of the time it depends on the data traffic and medium occupancy. To account for this, we set up deployments with up to 22 Talon AD7200 routers, with 4 configured as APs, 16 as stations and the other 2 are configured in monitor mode to redundantly capture the beam training messages. All APs use the same channel and use the standard CSMA/CA for medium access. The beam training of the devices uses 32 beacon frames, one per beam pattern in the default codebook, and takes around $1.6ms$ in total for one AP/STA pair. The Talon routers are designed for static scenarios and do beam training every 10 beacon intervals, but in scenarios with mobility training might be needed each beacon interval (i.e., every 100 ms). We add background traffic of 250 MBit/s per station and compare the measured and extrapolated overhead. To this end, we capture the traffic between 1 AP/station pair and multiply the AP overhead (DMG Beacon frames) by the number of APs and the station overhead (Sector Sweep during A-BFT and DTI, Sector Sweep Feedback, and Sector Sweep ACK) by the number of stations. This latter setup corresponds to deployments where the frequency of beam training cannot be reduced due to network dynamics. To obtain the overhead reduction achieved by SPIDER, we divide the total overhead by the overhead caused by a station implementing SPIDER (DMG Beacon, Sector Sweep Feedback, and Sector Sweep ACK).

Fig. 16a shows measured and extrapolated beam train-

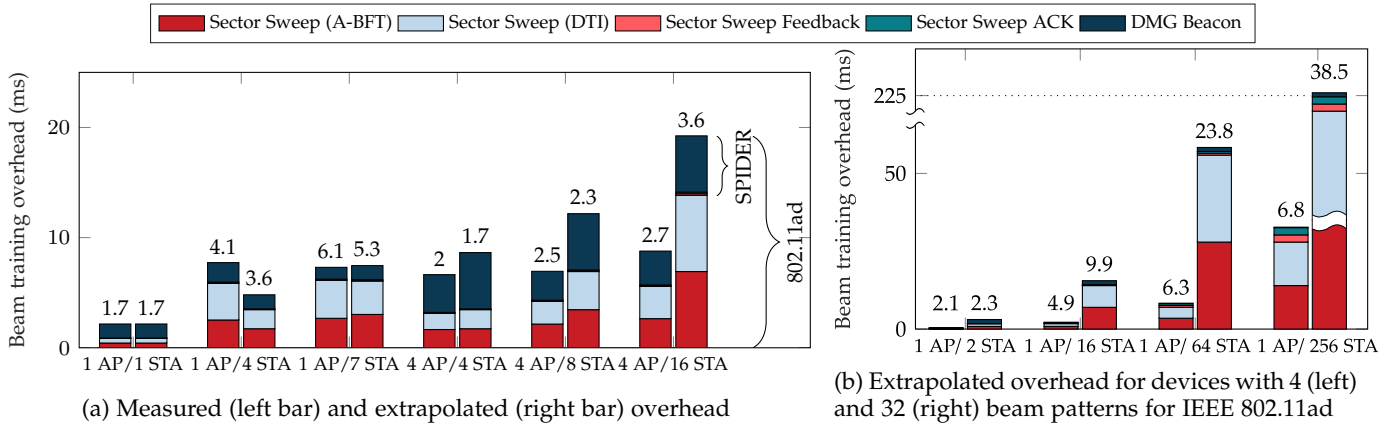


Figure 16: Beam training overhead (in ms) of IEEE 802.11ad compared to SPIDER (ratio on top of the bars)

ing overhead. For 1 AP and 1 station, AP overhead is slightly larger than station overhead and IEEE 802.11ad around 1.7 times the overhead of SPIDER). For 4 stations, the extrapolated station overhead is exactly 4 times larger and thus IEEE 802.11ad overhead is $3.6\times$ that of ours (left bar). However, the Talon routers both slightly increase the DMG Beacon frequency and significantly increase the station beam training frequency so that actually measured IEEE 802.11ad overhead is higher by a factor of 4.1. For 7 stations, the Talon routers reduce the beam training frequency due to the increase in data traffic with that many stations, so that measured and extrapolated overhead almost match. Note that due to issues with the newest Talon firmware it is not possible to connect 8 (or more) stations to one AP. For 4 APs with 1, 2, and 4 stations per AP, when extrapolating the overhead we obtain an IEEE 802.11ad overhead of 9, 12 and 19 ms, respectively, whereas the Talon routers try to maintain a relatively constant overhead of 6, 7 and 9 ms. For these cases, SPIDER reduces the overhead by a factor of 1.6 to 3.6. We see the effects of a reduction in beaconing frequency as is done by the Talon routers in the time it takes stations to connect to the network. For 4 APs and 4, 8, and 16 stations, the time it takes until the last station connects increases from 6 to 10 to 13 seconds, respectively.

Fig. 16b shows the extrapolated overhead for the standard setting with 32 beacon frames per beam training for IEEE 802.11ad and 10 beacon frames for SPIDER (right columns), as well as a scenario for low end IoT devices with only 4 beam patterns and thus 4 beacon frames (left columns) for both the IEEE 802.11ad baseline and SPIDER. We consider one AP and 2, 16, 64 and 256 stations. As expected, the higher the number of stations, the higher the beam training overhead for the baseline 802.11ad and the higher the relative overhead reduction with SPIDER. For standard devices with 32 beam patterns, the total beam training time with 256 stations is 227 ms, and thus beam training would only be feasible every few seconds. In dynamic scenarios, such infrequent beam training may lead to significant outage time. With SPIDER, beam training can still be done every 100 ms without excessive overhead, consuming only 6 ms. For low end devices with only 4 beam patterns, the overhead in general is substantially lower, as shown in the bars on the left. Nevertheless, with the standard approach the total beam training time would

consume 32% of the total communication time (since the beam training lasts 32 ms every 100 ms) with 256 stations. Although this does not prohibit communication, the time it takes significantly reduces the communication efficiency. Our approach would reduce this by a factor of 6.8 to only 4%.

6 RELATED WORK

We first discuss available mm-wave testbeds and then prior work on beam training. Since the advent of mm-wave communication systems, different experimentation platforms have been proposed, each trying to cover different types of experiments. These solutions differ vastly in cost, performance, and flexibility. Platforms using USRP software-defined radios [39, 51] are bandwidth constrained which limits their applicability to narrow-band experiments. Nevertheless, their low cost makes them a good solution for simple experiments that do not require high performance. The X60 testbed [52] fulfills the bandwidth requirements of the IEEE 802.11ad standard, but with limited flexibility regarding the fast real-time configuration of the antenna array and relatively high device cost. OpenMili from [53] increases the bandwidth capabilities compared to [39, 51], but still does not meet the requirements of IEEE 802.11ad. Besides, the custom phased array only includes very coarse phase control and its antenna reconfigurability is limited. Also COTS devices have been modified to perform mm-wave experiments [26]. While they are both low cost and standard compliant, they do not provide access to the physical and lower MAC layers, which are important to develop and validate new MAC and signal processing solutions. In contrast, our experimental platform with its fast real-time antenna reconfiguration provides unique capabilities for mm-wave experimentation with high mobility, where fast beam steering is important. It offers the flexibility to implement custom algorithms such as SPIDER, which no other available testbed platform can support. The platform allows to implement not only IEEE 802.11ad-based systems but also other mm-wave standards such as IEEE 802.11ay, which requires fast beam steering for its beam training mechanism and to track station movement during the data communication phase [31].

We now review existing beam training approaches. The IEEE 802.11ad beam training mechanism [2] selects the antenna sector that provides the highest SNR by exchanging SLS frames using beam patterns from a predefined codebook, as discussed in more detail in Section 2.1. Commercial devices typically have codebooks with up to 64 predefined sectors [5]. Beam training for such codebook sizes typically takes several milliseconds, but can take up to seconds in dense AP deployments [17, 32, 54]. It also incurs unacceptable latency and a very high beam training overhead in mobile scenarios [55, 56, 57].

The IEEE 802.11ay amendment [3] introduces the concept of *in-packet* training, where TRN fields are appended to the beacon frames. These fields allow to test different beam pattern configurations within a single packet [31]. However, each TRN field comprises several complementary Golay sequences, i.e., they are longer than the single Golay sequence per beam pattern change used in SPIDER. Besides, IEEE 802.11ay training requires additional fields, whereas SPIDER uses the existing frame preamble, preserving 802.11ad compatibility.

More efficient beam training mechanisms are mainly based on compressed beam training techniques, since the sparsity of the mm-wave channel makes it amenable to sparse estimation. Compressed sensing approaches reduce the number of sectors that are probed to $\mathcal{O}(K \log N)$, where N is the size of the codebook and K is the number of dominant paths in the mm-wave channel [28, 29]. Phase coherent compressive beam training mechanisms [23, 24] require phase coherence among the training frames, which is not supported by current COTS devices. For this reason, the authors only present simulation results for large antenna arrays. More recent approaches [25, 26, 27] use RSS information, which avoids the phase coherence requirement across frames and makes them implementable on current hardware. The approach in [27] divides the beam search space into narrow sectors and uses carefully designed multi-armed beam patterns which explore the different sectors similar to a hashing algorithm. A voting process returns the desired direction with a probability that depends on the specific hash functions and on the beam pattern shapes used. Current consumer devices with low resolution phase shifters, hardware imperfections and strong side-lobes will incur significant errors in this estimation. Compressed beam training can even be implemented on COTS devices using the (imperfect) beam patterns of the default codebook [26]. A subset of M out of the N beam patterns are probed ($M \ll N$), and the RSS measurements are correlated with the known radiation patterns of the default sectors of the devices. The authors report fluctuations in measuring the SNR which they address by estimation over multiple RSS measurements. The algorithm determines the maximum of the weighted sum of the radiation patterns over the search space, and the angle where this maximum is reached corresponds to the estimated direction. They show that 14 beam patterns suffice for reliable angle estimation.

Recent work proposes a new method for channel estimation using a linear block code [58], which reduces the number of measurements required. However, the algorithm is only tested in simulations with ideal beam patterns and therefore does not take into account hardware limitations.

It may fail in multi-path channels where paths have comparable strength. A similar idea is presented in [59] with a two-phase algorithm that can estimate a sparse multi-path channel from RSS measurements, as opposed to the authors' own prior work [25] that estimates only the dominant path. However, again only simulations with ideal beam patterns are presented. The components of a sparse multi-path channel can even be estimated with an overhead that only scales with the number of multi-path components of the channel but not the number of antenna patterns [30]. However, this approach requires solving a complex non-convex optimization problem via a genetic algorithm that is difficult to run on consumer hardware in real-time. It also requires multiple packet transmissions from both the AP and the stations, whereas with SPIDER the measurements taken over a single packet preamble from the AP suffice.

An approach that changes the beam pattern once in the middle of a preamble to a custom multi-lobe beam pattern with a phase shift among the lobes is presented in [60]. This mechanism allows to track the direction of simple device rotation and achieves zero-overhead device tracking with an angle error of less than 5° in most cases.

Few prior works address simultaneous station beam training [61, 62]. In [61], the authors propose a modification to 802.11ad where for station beam training, APs transmit omnidirectionally while stations switch through their different receive beam patterns. However, such an approach is not compatible with 802.11ad. Beam training with multiple APs is feasible but requires coordinating the beam training of all nodes of the network. Also [62] proposes a simultaneous beam training mechanism which is not compatible with legacy 802.11ad devices.

The main difference to our work is that SPIDER provides better scalability by eliminating station beam training overhead entirely, whereas prior work requires active station beam training. It is orthogonal to compressive beam training mechanisms that reduce the number of sectors trained by the AP, and can be used jointly with them to further reduce the overhead. Furthermore, our approach is compatible with current 802.11ad deployments, does not affect legacy stations, and can be implemented on current devices.

7 CONCLUSIONS

In this paper we introduce and implement SPIDER, a beam training mechanism that improves the scalability of IEEE 802.11ad systems for dense deployments. It makes use of rapid receive beam pattern changes during preamble reception for passive channel estimation at the stations, while an AP is performing standard beam training. The preamble measurements are phase-coherent, which allows to rapidly estimate the full multi-path channel based on dimensionality reduction methods and joint estimation over all received frames. To implement and validate SPIDER, we design a flexible full-bandwidth experimentation platform that is compatible with IEEE 802.11ad and allows to test our SPIDER station implementation with commercial APs. The testbed design and source code are made available to the research community as open source. Our evaluation shows that SPIDER provides extremely accurate angle measurements at the station that indicate the best beam pattern to

be used. Besides, it reduces the beam training overhead by an order of magnitude compared to IEEE 802.11ad. SPIDER also works for asymmetric arrays where transmit and receive array geometry differ, as long as the estimated angle can be mapped to an antenna sector.

Our mechanism can be further improved by designing custom beam patterns on the fly, that maximize the antenna gain in the estimated direction, rather than selecting from a pre-defined codebook. Furthermore, SPIDER can be used during data transmission in addition to the SLS, to continuously update the beam direction in highly dynamic scenarios. While data frame preambles are shorter than those of control frames, we show that even with a very low number of receive beam pattern changes it is possible to accurately track the AoA of the received signal. SPIDER can also be integrated at the AP side to further reduce the overhead. Angle estimation is then carried out whenever control messages from a station are received, and AP beaconing would only be necessary to allow new stations to join the network within a reasonable amount of time and to support legacy stations.

ACKNOWLEDGMENTS

This research work was sponsored in part by the European Union's Horizon 2020 research and innovation programme under Grant No. 871249 (LOCUS), the Spanish Ministry of Science and Innovation (MICIU) grant RTI2018-094313-B-I00 (PinPoint5G+) and the Region of Madrid through TAPIR-CM (S2018/TCS-4496).

REFERENCES

- [1] J. O. Lacruz, D. Garcia, P. J. Mateo, J. Palacios, and J. Widmer, "mm-FLEX: An Open Platform for Millimeter-Wave Mobile Full-Bandwidth Experimentation," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–13. [Online]. Available: <https://doi.org/10.1145/3386901.3389034>
- [2] IEEE 802.11 working group, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band." *IEEE Standard 802.11ad*, 2012.
- [3] —, "IEEE Draft Standard for Information Technology-Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications-Amendment: Enhanced Throughput for Operation in License-Exempt Bands Above 45 GHz," *IEEE P802.11ay/D3.0*, 2019.
- [4] W. Roh, J.-Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results," *IEEE communications magazine*, 2014.
- [5] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: Directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi," *IEEE Communications Magazine*, 2014.
- [6] I. Hwang, B. Song, and S. S. Soliman, "A holistic view on hyperdense heterogeneous and small cell networks," *IEEE Communications Magazine*, 2013.
- [7] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5G era: Enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, 2016.
- [8] M. Cheffena, "Industrial wireless communications over the millimeter wave spectrum: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 66–72, 2016.
- [9] K. Zrar Ghafoor, L. Kong, S. Zeadally, A. S. Sadiq, G. Epiphaniou, M. Hammoudeh, A. K. Bashir, and S. Mumtaz, "Millimeter-wave communication for internet of vehicles: Status, challenges, and perspectives," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8525–8546, 2020.
- [10] B. P. S. Sahoo, C. Chou, C. Weng, and H. Wei, "Enabling millimeter-wave 5g networks for massive iot applications: A closer look at the issues impacting millimeter-waves in consumer devices under the 5g framework," *IEEE Consumer Electronics Magazine*, vol. 8, no. 1, pp. 49–54, 2019.
- [11] B. Manz, "5G Cellular Networks Are the Future of Robotics," *Mouser Electronics*, 2016.
- [12] C. Zhang, D. Guo, and P. Fan, "Tracking angles of departure and arrival in a mobile millimeter wave channel," in *IEEE International Conference on Communications - ICC*, 2016.
- [13] V. Va, H. Vikalo, and R. W. Heath, "Beam tracking for mobile millimeter wave communication systems," in *IEEE Global Conference on Signal and Information Processing - GlobalSIP*, 2016.
- [14] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "BeamSpy: enabling robust 60 GHz links under blockage," in *13th Symposium on Networked Systems Design and Implementation*, 2016.
- [15] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering," in *IEEE Conference on Computer Communications - INFOCOM*, 2017.
- [16] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with eyes closed: mm-wave beam steering without in-band measurement," in *IEEE Conference on Computer Communications - INFOCOM*, 2015.
- [17] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "WiFi-Assisted 60 GHz Wireless Networks," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017.
- [18] V. Va, X. Zhang, and R. W. Heath, "Beam switching for millimeter wave communication to support high speed trains," in *IEEE 82nd Vehicular Technology Conference - VTC*, 2015.
- [19] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Beam alignment for millimetre wave links with motion prediction of autonomous vehicles," *IET*, 2017.
- [20] Z. Xiao, T. He, P. Xia, and X.-G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Transactions on Wireless Communications*, 2016.
- [21] S. Noh, M. D. Zoltowski, and D. J. Love, "Multi-resolution codebook and adaptive beamforming sequence design for millimeter wave beam alignment," *IEEE Transactions on Wireless Communications*, 2017.
- [22] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 GHz Indoor Networking Through Flexible Beams: A Link-Level Profiling," in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2015.
- [23] Z. Marzi, D. Ramasamy, and U. Madhow, "Compressive channel estimation and tracking for large arrays in mm-wave picocells," *IEEE Journal of Selected Topics in Signal Processing*, 2016.
- [24] D. Ramasamy, S. Venkateswaran, and U. Madhow, "Compressive adaptation of large steerable arrays," in *Information Theory and Applications Workshop*, 2012.
- [25] M. E. Rasekh, Z. Marzi, Y. Zhu, U. Madhow, and H. Zheng, "Noncoherent mmWave Path Tracking," in *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, 2017.
- [26] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, "Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.11Ad Devices," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies - CoNEXT*, 2017.
- [27] H. Hassanieh, O. Abari, M. Rodriguez, M. Abdelghany, D. Katabi, and P. Indyk, "Fast Millimeter Wave Beam Alignment," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication - SIGCOMM*, 2018.
- [28] P. F. M. Smulders, "Statistical Characterization of 60-GHz Indoor Radio Channels," *IEEE Transactions on Antennas and Propagation*, 2009.
- [29] T. S. Rappaport, F. Gutierrez, E. Ben-Do, J. N. Murdock, Y. Qiao, and J. Tamir, "Broadband Millimeter-Wave Propagation Measurements and Models Using Adaptive-Beam Antennas for Outdoor Urban Cellular Communications," *IEEE Transactions on Antennas and Propagation*, 2013.
- [30] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "Towards Scalable and Ubiquitous Millimeter-Wave Wireless Networks," in *Proceed-*

- ings of the 24th Annual International Conference on Mobile Computing and Networking - MobiCom, 2018.
- [31] Y. Ghasempour, C. R. C. M. da Silva, C. Cordeiro, and E. W. Knightly, "IEEE 802.11ay: Next-Generation 60 GHz Communication for 100 Gb/s Wi-Fi," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 186–192, 2017.
- [32] J. Palacios, P. Casari, H. Assasa, and J. Widmer, "LEAP: Location Estimation and Predictive Handover with Consumer-Grade mmWave Devices," in *The 38th IEEE International Conference on Computer Communications - INFOCOM*, 2019.
- [33] O. Tervo, T. Levanen, K. Pajukoski, J. Hulkkonen, P. Wainio, and M. Valkama, "5G new radio evolution towards sub-THz communications," in *2nd 6G Wireless Summit (6G SUMMIT)*. IEEE, 2020, pp. 1–6.
- [34] J. O. Lacruz and J. Widmer. (2020) mm-FLEX: Millimeter Wave Mobile Full-Bandwidth Experimentation Platform. [Online]. Available: <http://wireless.networks.imdea.org/mmflex>
- [35] W.-C. Liu, T.-C. Wei, Y.-S. Huang, C.-D. Chan, and S.-J. Jou, "All-Digital Synchronization for SC/OFDM Mode of IEEE 802.15.3c and IEEE 802.11ad," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2015.
- [36] N. Preyss and A. Burg, "Digital synchronization for symbol-spaced IEEE802.11ad Gigabit mmWave systems," in *IEEE International Conference on Electronics, Circuits, and Systems - ICECS*, 2015.
- [37] C.-Y. Liu, M.-S. Sie, E. W. J. Leong, Y.-C. Yao, C.-W. Jen, W.-C. Liu, C.-F. Wu, and S.-J. Jou, "Dual-Mode All-Digital Baseband Receiver With a Feed-Forward and Shared-Memory Architecture for Dual-Standard Over 60 GHz NLOS Channel," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2017.
- [38] P. Meher, J. Valls, K. Juang, Sridharan, and K. Maharatna, "50 Years ofCORDIC: Algorithms, Architectures, and Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 9, pp. 1893–1907, Sep. 2009.
- [39] A. Moreno, J. O. Lacruz, and J. Widmer, "Open Source RFNoC-Based Testbed for Millimeter-Wave Experimentation Using USRP Software Defined Radios," in *International Symposium on Circuits and Systems*, ser. ISCAS-2020, 2020.
- [40] B. Sadhu, Y. Tousi, J. Hallin, S. Sahl, S. K. Reynolds, Ö. Renström, K. Sjögren, O. Haapalahti, N. Mazar, B. Bokinge *et al.*, "A 28-GHz 32-element TRX phased-array IC with concurrent dual-polarized operation and orthogonal phase and gain control for 5G communications," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 12, pp. 3373–3391, 2017.
- [41] W. Liu, F. Yeh, T. Wei, C. Chan, and S. Jou, "A Digital Golay-MPIC Time Domain Equalizer for SC/OFDM Dual-Modes at 60 GHz Band," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 10, pp. 2730–2739, 2013.
- [42] P. J. Huber, "Projection pursuit," *The annals of Statistics*, 1985.
- [43] SIVERSIMA. (2019) EVK02001 28 GHz Development Kit. [Online]. Available: <https://www.siversima.com/product/evk-02001-00/>
- [44] Vadatech. (2019) AMC599 board. [Online]. Available: https://www.vadatech.com/media/AMC599_AMC599_Datasheet.pdf
- [45] ——. (2019) AMC726 board. [Online]. Available: https://www.vadatech.com/media/AMC726_AMC726_Datasheet.pdf
- [46] SIVERSIMA. (2019) EVK06002 Development Kit. [Online]. Available: <https://www.siversima.com/product/evk-600200>
- [47] JEDEC Solid State Technology Association. (2019) Serial Interface for Data Converters (JESD204B). [Online]. Available: <http://www.jedec.org/standards-documents/results/jesd204>
- [48] The MathWorks, Inc. (2020) WLAN Toolbox. [Online]. Available: <https://www.mathworks.com/products/wlan.html>
- [49] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [50] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition," *Sensors*, vol. 14, no. 6, p. 10146–10176, 2014.
- [51] O. Abari, H. Hassanieh, M. Rodriguez, and D. Katabi, "Poster: A millimeter wave software defined radio platform with phased arrays," in *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '16. New York, NY, USA: ACM, 2016, pp. 419–420. [Online]. Available: <http://doi.acm.org/10.1145/2973750.2985258>
- [52] S. K. Saha, Y. Ghasempour, M. K. Haider, T. Siddiqui, P. De Melo, N. Somanchi, L. Zakrajsek, A. Singh, R. Shyamsunder, O. Torres *et al.*, "X60: A programmable testbed for wideband 60 GHz WLANs with phased arrays," *Elsevier Computer Communications*, 2019.
- [53] J. Zhang, X. Zhang, P. Kulkarni, and P. Ramanathan, "Openmili: A 60 ghz software radio platform with a reconfigurable phased-array antenna," in *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '16. New York, NY, USA: ACM, 2016, pp. 162–175. [Online]. Available: <http://doi.acm.org/10.1145/2973750.2973760>
- [54] Y. Zhu, Z. Zhang, Z. Marzi, C. Nelson, U. Madhow, B. Y. Zhao, and H. Zheng, "Demystifying 60GHz Outdoor Picocells," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking - MobiCom*, 2014.
- [55] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Cutting the Cord in Virtual Reality," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks - HotNets*, 2016.
- [56] A. Loch, A. Asadi, G. H. Sim, J. Widmer, and M. Hollick, "mm-Wave on wheels: Practical 60 GHz vehicular communication without beam training," in *9th International Conference on Communication Systems and Networks - COMSNETS*, 2017.
- [57] Robert W. Heath. (2019) Vehicular Millimeter Wave Communications: Opportunities and Challenges. [Online]. Available: <http://users.ece.utexas.edu/~rheath/presentations/2015/DSTOPvehicularMmWave2015Heath.pdf>
- [58] Y. Shabara, C. E. Koksal, and E. Ekici, "Linear block coding for efficient beam discovery in millimeter wave communication networks," *arXiv*, 2017.
- [59] M. E. Rasekh and U. Madhow, "Noncoherent compressive channel estimation for mm-wave massive MIMO," *arXiv*, 2018.
- [60] A. Loch, H. Assasa, J. Palacios, J. Widmer, H. Suys, and B. Debaillie, "Zero Overhead Device Tracking in 60 GHz Wireless Networks using Multi-Lobe Beam Patterns," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies - CoNEXT*, 2017.
- [61] S. Shao, H. Zhang, D. Koutsonikolas, and A. Khreishah, "Two-dimensional reduction of beam training overhead in crowded 802.11 ad based networks," in *IEEE Conference on Computer Communications Workshops - INFOCOM WKSHPS*, 2018.
- [62] J. Qiao, X. Shen, J. W. Mark, and Y. He, "MAC-layer concurrent beamforming protocol for indoor millimeter-wave networks," *IEEE Transactions on Vehicular Technology*, 2015.



Dolores Garcia is a PhD student in Telematics Engineering at IMDEA Networks and University of Carlos III, Spain. Her research interest are in the intersection between machine learning and mm-wave communications. Previously, she completed an MSc in Theoretical Physics at Imperial College London in 2018 and a BSc in Mathematics at the University of Valencia in 2017.



Jesus O. Lacruz is a Research Engineer at IMDEA Networks, Spain since 2017. He received his Bachelor degree in Electrical Engineering from Universidad de Los Andes, Venezuela in 2009 and the PhD degree in Electronic Engineering from Universidad Politecnica de Valencia, Spain in 2016. His research interests lie in the design and implementation of fast signal processing algorithms for digital communication systems in FPGA devices.



Pablo Jimenez Mateo is currently pursuing a PhD in Telematics Engineering at IMDEA Networks and University of Carlos III of Madrid, Spain. He holds two Bachelor degrees, one in Computational Mathematics and another in Computer Engineering, both from Universitat Jaume I at Castellon de la Plana, Spain and 2 Master degrees, one in Artificial Systems from the same university and one in Telematics Engineering from Carlos III University at Madrid.



Joan Palacios received the BSc degree in mathematics from Universitat de Valencia, Spain, in 2015, and the MSc degree in multimedia and communications in 2016 and PhD degree in 2020, both from IMDEA Networks and Carlos III University, Spain. His areas of interest lie in mm-wave communications with focus on the development of efficient beam training and tracking schemes, hybrid analog-digital beamforming, and localization systems for quasi-optical communication networks.



Rafael Ruiz Ortiz received his Bachelor degree in Industrial Electronics and Automation Engineering from the Universidad Politecnica de Cartagena. He is currently pursuing a MSc in Industrial Electronics Engineering from the Universidad Politecnica de Madrid. He is a Research Engineer at IMDEA Networks Institute. His main research interests are focused on digital embedded system design and the implementation of accelerators based on FPGA devices.



Joerg Widmer (M'06-SM'10-F'20) is Research Professor and Research Director of IMDEA Networks in Madrid, Spain. Before, he held positions at DOCOMO Euro-Labs in Munich, Germany and EPFL, Switzerland. He was a visiting researcher at the International Computer Science Institute in Berkeley, USA, University College London, UK, and TU Darmstadt, Germany. His research focuses on wireless networks, ranging from extremely high frequency millimeter-wave communication and MAC layer design to mobile network architectures. Joerg Widmer authored more than 150 conference and journal papers and three IETF RFCs, and holds 13 patents. He was awarded an ERC consolidator grant, the Friedrich Wilhelm Bessel Research Award of the Alexander von Humboldt Foundation, a Mercator Fellowship of the German Research Foundation, a Spanish Ramon y Cajal grant, as well as eight best paper awards. He is an IEEE Fellow and Distinguished Member of the ACM.