Octopus: A Practical and Versatile Wideband MIMO Sensing Platform

Zhe Chen[1,2*], Tianyue Zheng[1*], Jun Luo[1]
[1] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2] Guangxi Wanyun Technology Co., Ltd, Nanning, China
Email: chenz@ssijri.com, {tianyue002, junluo}@ntu.edu.sg

ABSTRACT

Radio frequency (RF) technologies have achieved a great success in data communication. In recent years, pervasive RF signals are further exploited for sensing; *RF sensing* has since attracted attentions from both academia and industry. Existing developments mainly employ commodity Wi-Fi hardware or rely on sophisticated SDR platforms. While promising in many aspects, there still remains a gap between lab prototypes and real-life deployments. On one hand, due to its narrow bandwidth and communication-oriented design, Wi-Fi sensing offers a coarse sensing granularity and its performance is very unstable in harsh real-world environments. On the other hand, SDR-based designs may hardly be adopted in practice due to its large size and high cost. To this end, we propose, design, and implement Octopus, a compact and flexible wideband MIMO sensing platform, built using commercial-grade low-power impulse radio. Octopus provides a standalone and fully programmable RF sensing solution; it allows for quick algorithm design and application development, and it specifically leverages the wideband radio to achieve a competent and robust performance in practice. We evaluate the performance of Octopus via micro-benchmarking, and further demonstrate its applicability using representative RF sensing applications, including passive localization, vibration sensing, and human/object imaging.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**; • **Hardware → Sensor devices and platforms**; *Signal processing systems.*

KEYWORDS

RF-sensing platform, IR-UWB, MIMO, edge computing and sensing.

ACM Reference Format:
Z. Chen, T. Zheng, and J. Luo. 2022. Octopus: A Practical and Versatile Wideband MIMO Sensing Platform. In *The 27th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'21), January 31-February 4, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3447993.3483267

* Both authors contributed equally to this research.

ACM MobiCom'21, January 31-February 4, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-8342-4/22/01…$15.00
https://doi.org/10.1145/3447993.3483267

1 INTRODUCTION

Wireless technologies have changed our world and obtained a significant success in both communication and sensing. On one hand, a wide range of radio frequency (RF) signals have been employed for wireless communication such as Wi-Fi [25] and LTE [6], delivering a ubiquitous connectivity to us. On the other hand, RF signals are further exploited for sensing in recent year, and promising progress has since been achieved. Novel sensing models and algorithms have been developed for user localization or tracking [2, 3, 16, 62], human activities/gesture recognition [1, 13, 18, 55, 63], and vital sign monitoring [15, 61, 66]. Compared with traditional device-based sensing approaches, RF sensing has the unique advantages of being both *device-free* and *contact-free*.

Though promising, most sensing solutions are hosted on Commercial Off-The-Shelf (COTS) chipset such as Wi-Fi cards [16, 21, 61] and Software Defined Radio (SDR) platforms [1–3, 41, 62]. However, the high-end hardware components (e.g., the high sampling rate ADCs) of SDR platforms are prohibitively expensive and hence unlikely to be available on commodity hardware. Therefore, the better performance of SDR can hardly be relevant to real-life deployments. Meanwhile, great efforts have been devoted to Wi-Fi sensing using Intel and Atheros Wi-Fi chipsets. While optimistic results have been achieved in controlled setups under lab environments [16, 42, 55], severe performance degradation may take place in real-life deployments. This stems from the communication-oriented design of Wi-Fi: a clean channel has to be reserved for sensing and transmission settings (e.g., packet size, packet interval, and data rate) need to be precisely controlled. In reality, clean channels rarely exist and all parameters can be out of control, rendering the performance in the wild much worse than that achieved in a laboratory. Last but not least, most up-to-date RF sensing systems do not operate in a *standalone* basis, as they often require an external computing device to perform signal processing. This seriously limits their ability in wide-scale deployments, because they can barely be supported by edge computing. We summarize the key features of existing platforms in Table 1.

In this paper, we propose Octopus as a compact and programmable wideband platform dedicated to RF sensing. As shown in Table 1, Octopus aims to tackle all the aforementioned challenges faced by existing platforms, by exploiting, in particular, ultra-wideband (UWB) radio, large-scale MIMO, and a standalone implementation with inexpensive commercial-grade components. Essentially, Octopus leverages its novel RF components (including both UWB impulse radios and a MIMO antenna array) to obtain RF In-phase and Quadrature (IQ) baseband signal samples, as illustrated by Figure 1. Under the standalone mode, these samples are processed only by the hardware and software components of Octopus. The

Table 1: Comparison Between Different RF Sensing Platforms.

Platform	Standalone	Bandwidth (GHz)	Implementation	Frequency (GHz)	Antenna setting	Antenna arrays	Tx power (mW)
Intel 5300 [21, 42]	No	0.04	COTS	2.4, 5.8	Bistatic	3×3	1000
Atheros 802.11n [57]	No	0.04	COTS	2.4, 5.8	Bistatic	3×3	1000
WiTrack1.0 [3]	No	1.69	USRP+VCO+DDS	6.35	Monostatic	1×3	0.75
WiTrack2.0 [2]	No	1.69	USRP+VCO+DDS	6.35	Monostatic	3×5	0.75
RF-Capture [1]	No	1.69	USRP+VCO+DDS	6.35	Monostatic	4×12	0.75
Octopus	**Yes**	**1.50**	**Commercial-grade**	**7.29**	**Monostatic**	$\mathbf{8 \times 8}$[1]	**0.084**

[1] The antenna array can be customized; it is extensible to larger scales, such as 8×16, 16×16, or even 32×32.

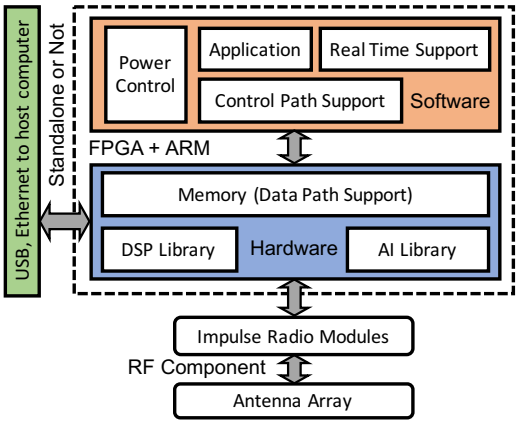


Figure 1: System overview of Octopus.

hardware component contains DSP and AI libraries (e.g., FFT and neural networks) to accelerate essential data processing. The software component executes control function to, for example, toggle Octopus between sleep and active modes for energy saving, and it also allows for deploying standalone applications. If necessary, Octopus can offload IQ baseband samples to a host computer for quickly verifying proof-of-concept prototypes. Therefore, Octopus is capable of closing the gap between novel sensing techniques and realistic deployment requirements, and it may open up new opportunities for RF sensing design and development.

From hardware architecture perspective, Octopus aims to achieve high performance, low-power, flexibility and small form-factor, by exploiting commercial-grade UWB radio chipsets, RF switches, ARM-based MCU, and Field Programmable Gate Array (FPGA). Building upon a UWB RF chip [5] offering low power consumption and full access to baseband data, we design a two-level hybrid antenna array involving multiple RF chips and switches, with each chip operating multiple antennas in a time-division manner at μs granularity, and multiple chips synchronized to form a larger antenna array for achieving finer spatial resolution. We also adopt an FPGA component to provide rich I/O interfaces such as high speed connection bus line, Serial Peripheral Interface (SPI), and Universal Asynchronous Receiver-Transmitter (UART). Meanwhile, we further leverage the FPGA to accelerate real time processing, targeting deep learning based sensing applications that require, for example, Convolutional Neural Network (CNN). As the central "brain", the ARM-based MCU schedules all I/O operations among FPGA, RF switch, and UWB radio chipsets, and it executes all kinds of sensing algorithms under the standalone mode.

As Octopus couples software and hardware closely, software framework of Octopus needs to provide a user-friendly programming model, so as to provide developers full access to hardware features. To this end, we employ a Directed Acyclic Graph (DAG) based work flow scheduler [23] to reconfigure intermediate jobs of an application and to exploit job parallelism. We treat each thread as a node in DAG, and multiple threads can synthesize jobs in hybrid FPGA and ARM architecture. Moreover, software is mapped directly to accelerators in FPGA via direct memory access (DMA). In this way, data swapping between software and hardware accelerator incurs a minimal latency. Since the framework is developed by C/C++ language, developers can readily deploy their application prototypes based on this familiar programming model.

Essentially, we make the following major contributions in designing Octopus as the first standalone RF sensing platform:

- We develop a compact yet scalable sensing platform equipped with a GHz bandwidth and a large-scale antenna array. It can operate in a standalone mode, and its low-power nature is compatible with battery-powered edge computing deployments.
- We design a user-friendly programming model to support efficient job scheduling within FPGA and ARM. Though supporting only one application at runtime due to limited FPGA resource incapable of concurrent low-level task execution, it allows for parallelizing lower-level (in FPGA) and high-level (in ARM processor) tasks.
- We propose sensing algorithms to fully utilize the unique characteristic of wideband signals, so as to achieve a better performance in terms of both accuracy and robustness, and to move RF sensing one step further towards real-life adoptions.
- We showcase the flexibility of Octopus via several representative sensing applications, demonstrating its excellent performance and exceeding capability in serving as both a research and development platform. Our software demonstrations and hardware designs are both publicized at Github [38].

Note that the scope of Octopus is orthogonal and complementary to that of the mmWave-based technologies [35, 39, 52, 65], due to substantially different frequency bands and thus sensing capabilities. In the following, we introduce the hardware architecture, interfaces, and software framework respectively in Sections 2, 3,

and 4. We present sensing algorithms specifically designed for Octopus's wideband MIMO radio in Section 5, and we report the system evaluation and case studies in Section 6. Finally, we discuss the related work in Section 7, before drawing a conclusion in Section 8.

2 BASIC HARDWARE DESIGN

For flexible deployment, Octopus is designed as a *monostatic* sensing platform, where the transmitter and receiver are co-located. As illustrated in Figure 2, the hardware architecture of Octopus has two major parts: the large-scale MIMO and digital processing module. The former transforms the analog RF signals into digital baseband IQ signals, which contain a high spatial sensing resolution due to the impulse radio's wide bandwidth and MIMO's spatial multiplexing. The latter includes a heterogeneous ARM-FPGA architecture for configuring and controlling the MIMO radio, as well as for accelerating digital processing. Moreover, Octopus supports baseband data offloading to host PC, allowing for design and quick verification of new application prototypes.

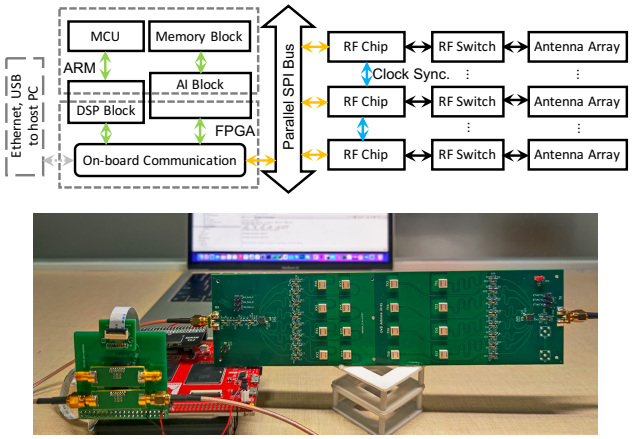


Figure 2: Hardware architecture and image of Octopus.

2.1 Large-Scale MIMO Design

Figure 2 shows a hybrid architecture of Octopus's large-scale MIMO, aiming to achieve both *scalability* and *programmability*. While the former allows for an easy scaling up of the size of antenna array, the latter enables the system to support real-time antenna selection within the array.

Existing large-scale antenna array architectures are divided into three major classes: digital antenna array, switched antenna array, and phased antenna array. A digital antenna array employs one RF chip for every antenna, and can transmit or receive signals

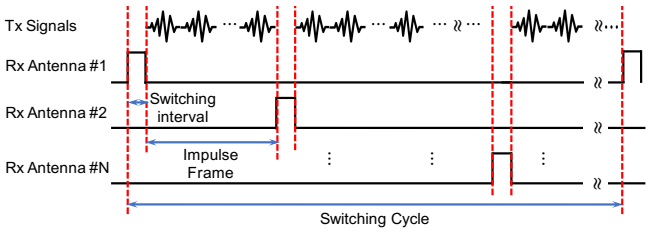
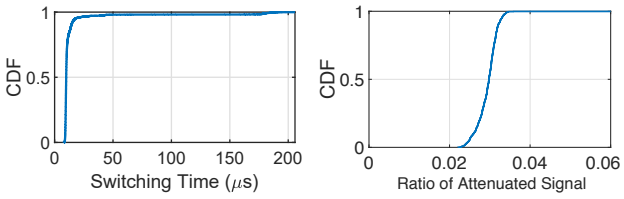


Figure 4: Timing diagram of the Rx RF switch for a single RF chip. When multiple Tx antennas are involved, above operations are repeated for each Tx antenna.

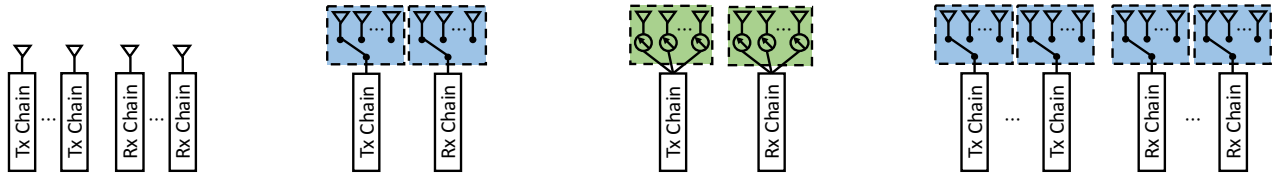
simultaneously as shown in Figure 3a. A switched antenna array has only one RF chain but requires an RF switch to select antennas in a time-divided manner as illustrated in Figure 3b. A phased antenna array consists of multiple antennas with phase shifters; it selects different beamforming patterns to scan pre-defined directions, as shown in Figure 3c. However, all the three classes have respective issues. The digital antenna array needs to synchronize all RF chips, which may not scale well due to the throughput limit of clock I/O. The switched antenna array introduces a considerable time delay, an inherent weakness of time division systems. For the phased antenna array, the phase shifter for wideband signals is extremely expensive and very hard to manufacture [36], preventing it from a wide adoption in practical edge computing.

Considering the aforementioned pros and cons, we propose a two-level hybrid architecture for Octopus as shown in Figure 3d; it comprises a digital antenna array with multiple RF chips at the first level and each RF chip drives a switched antenna array via an RF switch. The Rx switch timing diagram of a single RF chip is shown in Figure 4, and that for Tx switch is similar. Typically, the pulse frame duration T_s is at millisecond level, and the switching interval between antennas T_i should be much shorter. Therefore, we select HMC321 [4] as our RF switch because it has a switching time largely below 10 μs as shown in Figure 5a, satisfying the stringent



(a) CDF of switching time. **(b) CDF of attenuated ratio.**

Figure 5: HMC321 RF switch performance: CDFs of (a) the switching time and (b) the attenuated ratio.



(a) Digital antenna array. **(b) Switched antenna array.** **(c) Phased antenna array.** **(d) Hybrid antenna array.**

Figure 3: Three major classes of large-scale antenna array architecture (a)–(c), vs. that of Octopus (d).

timing requirement between pulse frames and supporting up to 1.2 kHz frame rate given the 0.8 ms frame length. We also verify the amplitude attenuation caused by HMC321; the results shown in Figure 5b confirm a slight attenuation introduced by the switch, where 60% amplitude attenuation is higher than 0.97 (-0.2646 dB) and the minimum value is still 0.94 (-0.5374 dB). Note that the self-interference between co-located Tx/Rx antennas can be handled either by removing the first few samples (thanks to the high range resolution) or by applying self-interference cancellation.

2.2 RF Chip Selection

Extracting RF features from raw complex IQ baseband samples is the most crucial goal for RF sensing. SDR-based platforms with a large bandwidth have demonstrated the capability to obtain richer RF features than Wi-Fi based narrowband solutions [2, 3]. However, as they leverage Frequency-Modulated Continuous-Wave (FMCW) technology for RF sensing and FMCW is not offered by any commercial-grade devices at sub-10 GHz range, a combination of Voltage Controlled Oscillator (VCO), Direct Digital Frequency Synthesizer (DDS), and SDR platform (e.g., USRP) have to be used to achieve the sensing objective. Consequently, the resulting platforms incur both high power consumption and high cost. Moreover, the scalability of FMCW-based platforms in practical applications is also questionable at sub-10 GHz band. For instance, if multiple FMCW platforms operate simultaneously, differentiating signals among them could be a challenge [31].

Octopus is designed to offer a wide bandwidth at a reasonable cost. We carefully analyze all commercially RF chips with a large bandwidth and identify the X4 chip from Novelda [5] as the basis of our platform. The X4 RF chip is an impulse wideband transceiver operated at 7.29 GHz; it supports direct RF signal sampling/generating with a 11 bits Analog-to-Digital Converter (ADC)/Digital-to-Analog Converter (DAC). More importantly, multiple X4 chips can be cascaded to form a MIMO module via hardware synchronization, from which all IQ baseband samples are read to MCU or FPGA via Serial Peripheral Interface (SPI). Finally, we may leverage advanced power management offered by the X4 chip to enable low power sensing operation. It is worth noting that pulse transmissions from different sources are virtually collision-free [43], naturally enabling simultaneous operations of multiple platforms.

2.3 Reconfigurable Hardware Architecture

The reconfigurable hardware is mainly used to retrieve and process data from RF chips, and the requirements it needs to meet are twofold: standalone operation and parallel data exchange/processing. We observe that a heterogeneous ARM-FPGA architecture can fulfill both requirements, as ARM provides flexible usability for users under standalone mode, while FPGA supports large scale parallel I/O operations and accelerates data processing. To build a low-power and cost-effective system, we choose Intel EP4CE15, a SRAM-based Cyclone series FPGA [27] offering not only adequate resources (e.g., logic elements) but also a large number of I/O interfaces for data communication and digital control. This FPGA has up to 343 I/O pins, readily supporting 80 general purpose I/O pins and up to 128 antennas driven by 8 RF chips and 16 (8-port) RF switches. Moreover, Octopus leverages FPGA to accelerate signal processing blocks such as FFT, FIR filters, and CNN.

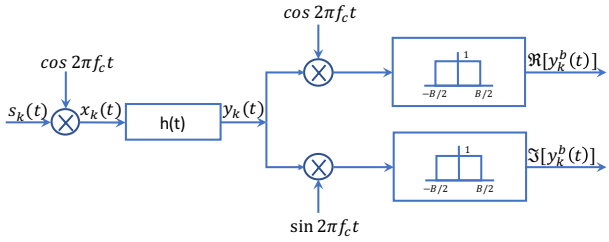


Figure 6: Transceiver diagram of X4 chip from the baseband Tx signal $s_k(t)$ to the baseband Rx signal $y_k^b(t)$.

To further enable standalone operations, we adopt an ARM Cortex-M7 32-bit RISC core chip as the MCU for Octopus [49]; it collaborates with FPGA to handle algorithms imposed by applications and to control individual digital chips (e.g., SDRAM). This high performance chip operates at up to 216 MHz frequency, and it has a 512 KB onboard SRAM and a 2 MB onboard flash memory. Aside from standalone operations, this microprocessor also provides advanced connectivity via USB and Ethernet, allowing Octopus to route raw RF samples from FPGA to a host PC.

3 INTERFACING MODULES

We hereby explain connecting paths (shown in Figure 2 as wide arrows) to interface various modules.

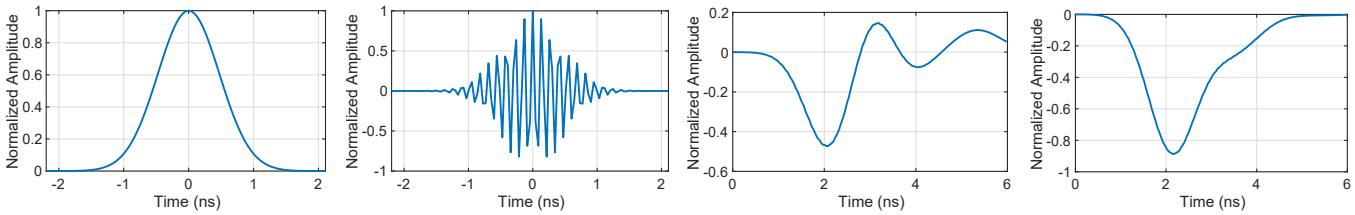
3.1 Transceiver Signal Path

The transceiver of X4 chip can provide raw (real) baseband or IQ baseband samples, and its diagram is shown in Figure 6. The transmitted (Tx) baseband signal $s_k(t)$ shown in Figure 7a as the k-th frame is a Gaussian pulse as follows:

$$s_k(t) = V_{\text{tx}} e^{-0.5(t-0.5T_{\text{tx}}-kT_s)^2 \epsilon_{\text{tx}}^{-2}}, \tag{1}$$

where V_{tx}, T_{tx} ,$\epsilon_{\text{tx}} = \frac{1}{2\pi B_{-10\text{dB}} (\log_{10}(e))^{1/2}}$ and $T_s = \frac{1}{f_p}$ are the pulse amplitude, signal duration, standard deviation determining the -10dB bandwidth, and frame duration with f_p being the pulse repetition frequency, respectively. After up-conversion, the Tx signal becomes $x_k(t) = s_k(t)\cos(2\pi f_c(t - kT_s))$ where f_c is the carrier frequency. Figure 7b demonstrates the actual Tx signal $x_k(t)$ with a single tone cosine carrier wave at $f_c = 7.29$ GHz multiplying the baseband signal $s_k(t)$ with a 1.5 GHz bandwidth. In Octopus, we have customized V_{tx} to have a tunable range of 20.97 dB, and f_c to have four choices: 5.8, 7.29, 8.7, or 9.1 GHz.

The Channel Impulse Response (CIR) $h_k(t)$ is represented by $h_k(t) = \sum_{p=1}^{P} \alpha_p \delta\left(t - \Gamma_p\right)$ where $\Gamma_p = \tau_p + \tau_p^{\text{d}}(kT_s)$. The variable τ_p is the time delay of the p-th reflection path signal, α_p is the channel gain of that, and $\tau_p^{\text{d}}(kT_s)$ is the time delay caused by Doppler frequency shift. For monostatic transceiver, $\tau_p = \frac{2R_p}{c}$, and $\tau_p^{\text{d}}(kT_s) = \frac{2v_p kT_s}{c}$ where R_p, c and v_p are the distance between object and transceiver, the propagation speed of radio and the velocity of the object, respectively. Furthermore, $\Delta r = \frac{c}{2B}$ is the range resolution where B is the bandwidth of the transceiver, and the time



(a) Baseband signal $s_k(t)$. (b) Actual Tx signal $x_k(t)$. (c) Real baseband Rx $\Re(x_k(t))$. (d) Imaginary baseband Rx $\Im(x_k(t))$.

Figure 7: From transmitted (Tx) baseband signal to received (Rx) baseband IQ signal.

delay resolution is $\Delta\tau = \frac{1}{2B}$. Given $x_k(t)$ and $h_k(t)$, the receiving signals $y_k(t)$ become:

$$y_k(t) = \sum_{p=1}^{P} \alpha_p \cos(2\pi f_c(t - kT_s - \Gamma_p))s_k(t - \Gamma_p) + n(t),$$

where $n(t)$ is Gaussian noise with variance σ^2. In practice, as $kT_s \gg t$, the receiving signals $y_k(t)$ are divided into two dimensions: *fast time* t and *slow time* kT_s. The receiving IQ baseband signals $y_k^b(t)$ are obtained via IQ down-conversion:

$$y_k^b(t) = \sum_{p=1}^{P} \alpha_p e^{-j2\pi f_c \Gamma_p} s_k(t - \Gamma_p) + n(t). \quad (2)$$

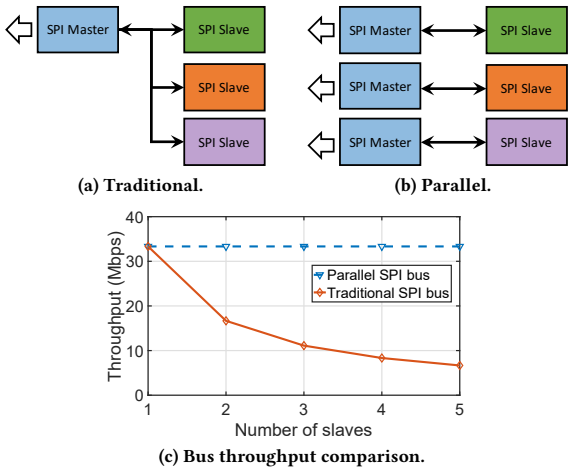
An example of received (Rx) baseband IQ (complex) signals are shown in Figures 7c and 7d. The current model is only for a single chip and antenna; a model specifically for MIMO will be further discussed in Section 5.

3.2 Baseband to FPGA via Parallel SPI Bus

To accelerate the communication with multiple RF chips, we use multiple parallel SPI buses to create a high-speed interface, instead of traditional independent slave configuration of SPI [40]. Traditional setting involves one master and multiple slaves shown in Figure 8a. The SPI master (e.g., FPGA) generates clocks to write and read data from different slaves via Chip Selection (CS) ports. However, such a mode decreases the throughput of each RF chip, hence making the frame rate f_p too low to support a MIMO system. In order to support up to 1.2kHz frame rate with default 138 fast time samples per frame, the throughput requirement per chip should be $1.2 \times 138 \times 8 \times 8 = 10.6$Mbps (4 bytes for I and Q components, respectively). The traditional SPI setting with a 33MHz clock can only support one RF chip, as two chips already decrease the bus throughput below the required 21.2Mbps according to Figure 8c. Therefore, leveraging the rich I/Os of FPGA, we design a parallel SPI bus to keep the throughput stable as illustrated in Figure 8b. This parallel structure allows FPGA to communicate with each RF chip independently. We evaluate the bus throughput under both modes in Figure 8c. Clearly, only the parallel SPI bus can maintain a stable throughput for every RF chip. Data retrieved through the parallel SPI bus are cached in SDRAM.

3.3 Interfacing FPGA to ARM-based MCU

We interface the FPGA with the ARM-based MCU via a Flexible Static Memory Controller (FSMC): it provides a 16-bit data bus and a 26-bit address bus. Given the throughput requirement per RF chip stated in Section 3.2, to support 8 RF chips requires a 84.8Mbps throughput. The MCU's 100MHz clock [49] enables FSMC to achieve 1.6Gbps write/read throughput. Although only a read or



(a) Traditional. (b) Parallel.

(c) Bus throughput comparison.

Figure 8: Comparing traditional and parallel SPI buses in terms of structure and performance.

write can be performed within one clock cycle, the system barely needs to write data to RF chips after initialization. Therefore, FSMC can at least support a throughput up to several hundreds of Mbps even without 100% efficiency, far sufficient to accommodate the 84.8Mbps demand from RF chips. Moreover, Octopus can route data to a host PC for fast prototyping via USB/Ethernet interface. Such operations are enabled by transferring data directly from SDRAM to USB/Ethernet leveraging the MCU's DMA support.

4 PROGRAMMING MODEL

For conveniently developing applications on Octopus, we employ a simple programming abstraction in C/C++ language, based on Directed Acyclic Graph (DAG) [33]. We treat every application as a graph, with each node being a basic unit for modular programming, implementing a processing operation or fundamental function. The processed data are transferred among nodes via edges. Therefore, a DAG graph is a flowchart for an application with multiple nodes as a processing unit and edges as data flows. In particular, each node has four elements: input ports, output ports, function unit, and configuration. Input and output ports are used to interface with other nodes via edges, function unit could be implemented in Verilog or C/C++ language, and configuration is used to set parameters for each node. In Octopus, the whole application following a DAG flowchart is scheduled by the ARM-based MCU, and the FPGA provides both peripherals interfaces and hardware accelerations.

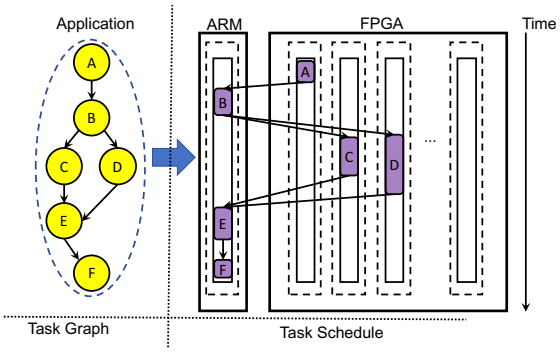


Figure 9: An example application detecting human presence: a DAG graph presentation and actual implementation. A to F denote data load, data existing check, fast motion estimation, slow motion estimation, presence detection, and Ethernet upload, respectively. C and D employ FFT to compute Doppler frequencies for fast and slow motion estimations, accelerated by FPGA. B and E are processed in MCU for flexible implementation, and Ethernet interface F allows for external access for demonstration purpose.

For brevity, we omit the discussion on how individual nodes (e.g., DSP or AI modules in either FPGA or ARM) are constructed, but rather focus only on how to support connections among nodes.

In order to fully support connections (edges) between FPGA and ARM-based MCU, four different cases exist: i) ARM to ARM, ii) FPGA to FPGA, iii) ARM to FPGA, and iv) FPGA to ARM. Obviously, case i) is naturally supported in any C/C++ programming of ARM-based MCU via pointers. Case ii) is also straightforward by integrating different function units into one using Verilog in FPGA. However, for cross-domain message exchanges in cases iii) and iv), one side does not know the function units' addresses on the other side, making it necessary to perform address query before function unit access. We design an address interpreter in FPGA that maps an identifier fpga_id invoked from ARM to the corresponding function unit address in FPGA. Though this interpreter allows ARM to query any function unit in FPGA, the access from the reversed direction can be more complicated, because C/C++ functions are often dynamically linked, causing function unit addresses in ARM to be dynamic too. Therefore, in addition to an identifier arm_id, we further implement an address query function arm_addr_query in ARM for address translation. For example, a function calling from FPGA carries the arm_id of a function unit in ARM, and it triggers an Interrupt ReQuest (IRQ). In response to this IRQ, ARM calls the function arm_addr_query in the IRQ handler to query the right function unit address. In fact, the above procedures described for control flow equally applies to data flow that swaps data among DAG nodes. Figure 9 illustrates an implementation example. The applications used for our later case studies in Section 6.2 are all developed using this programming model.

5 WIDEBAND MIMO SENSING METHODS

In this section, we present the sensing methodology for Octopus, by focusing two parameter estimations, namely angle and range. We start with a MIMO channel model, then we introduce an accurate parameter estimation algorithm and a general joint-angle-range estimation approach, both leveraging the wideband sensing capability of Octopus. At last, we touch upon array calibration to handle system errors.

5.1 A MIMO Channel Model

We start by extending the single channel model in Section 3.1 to MIMO channel. For simplicity, we consider a linear antenna array, whose receiving switching cycle is denoted as $T_{\text{sc}}^{\text{rx}} = NT_s$, and the transmitting switching cycle is $T_{\text{sc}}^{\text{tx}} = MNT_s$ where N and M are the cardinalities of receiving and transmitting antennas, respectively. In light of Eq. (2), the delays Γ_p of the p-th reflection path in the received baseband signals $y^b_{k,n,m}(t)$ can be modeled as following:

$$\begin{aligned}
\Gamma_p &= \tau_p + \tau_p^{\text{d}}\left((kMN + mN + n)T_s\right) + \tau_p^{\text{a}_\text{r}}(n) + \tau_p^{\text{a}_\text{t}}(m),\\
\tau_p^{\text{a}_\text{r}}(n) &= \frac{d\cos(\theta_p)}{c}n, \qquad n = 0, 1, \cdots N-1,\\
\tau_p^{\text{a}_\text{t}}(m) &= \frac{d\cos(\phi_p)}{c}m, \qquad m = 0, 1, \cdots M-1,
\end{aligned} \tag{3}$$

where d denotes the distance between consecutive antennas, and θ_p and ϕ_p are respectively the AoA (Angle-of-Arrival) and AoD (Angle-of-Departure) of the p-th reflection path. With a two-level MIMO architecture, the time delays caused by different antennas at the i-th Rx chain and the l-th Tx chain are $\tau_p^{\text{a}_\text{r}}(i,n) = \frac{d\cos(\theta_p)}{c}(iN+n)$ and $\tau_p^{\text{a}_\text{t}}(l,m) = \frac{d\cos(\phi_p)}{c}(lM+m)$, respectively. Note that it is the monostatic nature of our radio that enables a delay-based model directly inferring distance, rather than a commonly used phase-based model representing only difference in distance.

5.2 Wideband Parameter Estimations

We first briefly justify the benefit of wideband sensing against narrowband schemes based on, e.g., Wi-Fi. For simplicity, we focus only on AoA estimation as other estimations are related. Given Eq. (3), the *phase difference vector* $\Delta\boldsymbol{\phi}$ between the n-th Rx antenna and the reference (0-th) antenna becomes

$$\Delta\boldsymbol{\phi}(n) = \frac{2\pi\mathbf{f}d\cos(\theta_p)}{c}n \tag{4}$$

where $\mathbf{f} = [\cdots, f_i^-, \cdots, f_1^-, f_c, f_1^+, \cdots, f_i^+, \cdots]$ can be deemed as the OFDM-equivalent representation of the frequency band, with all these frequency components belonging to $(f_c - B/2, f_c + B/2)$. As a wider bandwidth leads to more orthogonal frequency components, Eq. (4) implies more independent equations under wideband sensing. In other words, wideband sensing intrinsically offers a larger frequency diversity in parameter estimation. In practice, wideband signals are mostly processed in time domain. As shown in Figure 10, a narrowband $s_k(t)$ (produced by, e.g., Wi-Fi) has similar envelopes sampled by the antenna array. Consequently, any estimation algorithm would suffer from both low accuracy and coarse resolution, as the antenna diversity of MIMO cannot be fully exploited. Therefore, the frequency diversity offered by wideband sensing manifests itself in time domain as the waveform's time-sensitivity, thus allowing every obervation made by an antenna to be effectively counted.

To take into account Γ_p in Eq. (3), we construct a receiving matrix $\mathrm{Y} = [\boldsymbol{y}^b_{1,k,1,m}, \boldsymbol{y}^b_{1,k,2,m}, \cdots, \boldsymbol{y}^b_{I,k,N,m}]^T$ with $\boldsymbol{y}^b_{i,k,n,m}$ being a row

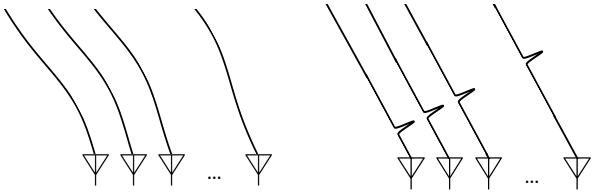


Figure 10: Narrowband vs wideband.

vector with a number of T samples for the i-th RF chain and the n-th antenna. Then we can jointly estimate $\tau_p^{\text{a}_\text{r}}(i,n)$ (thus θ_p) and τ_p by solving the following optimization problem, instead of estimating them separately.

$$\arg\min_{\boldsymbol{\tau}} \left\| \sum_{t=1}^{T} \sum_{p=1}^{P} \alpha_p e^{-j2\pi f_c \Gamma_p} s_k(t-\Gamma_p) - \Upsilon(:,t) \right\|_2, \tag{5}$$

where $\boldsymbol{\tau} = [\tau_1^{\text{a}_\text{r}}, \cdots, \tau_P^{\text{a}_\text{r}}, \tau_1, \cdots, \tau_P]^T$. We employ an iterative EM (Expectation-Maximization) algorithm to solve the above problem, so as to obtain $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_P]^T$ and also the range information derivable from $\boldsymbol{\tau}$. In fact, other parameters such as Doppler shift (or velocity) can also be estimated, but we omit them for the sake of brevity.

5.3 General Wideband Sensing

The EM algorithm described in Section 5.2 is only applicable to scenarios where the number of reflection paths P is known. However, more complex cases may involve i) unknown number of targets in the space, or ii) a target of unknown shape (which is of interest and hence cannot be deemed as a point). As these cases all entail an unknown P, we hereby explore a more general wideband sensing algorithm to handle them.

We define the *data acquisition plane* as the plane on which the radio transceiver resides. The basic idea of this algorithm is to back-project the received signals on the data acquisition plane to the target point. For brevity, we use a 1D Rx array to illustrate the process in Figure 11. It can be geometrically shown that the reflected signals from a specific target, when viewed from different Rx antennas, form a hyperbola in the plane defined by the Rx antenna indices and fast time t. Consequently, reflection from the target point, which is exactly located at the vertex of the hyperbola, can be obtained by summing up the signals along the curve. Likewise, this method can be readily extended to the case of 2D antenna arrays by summing up received signals on a 3D hyperboloid.

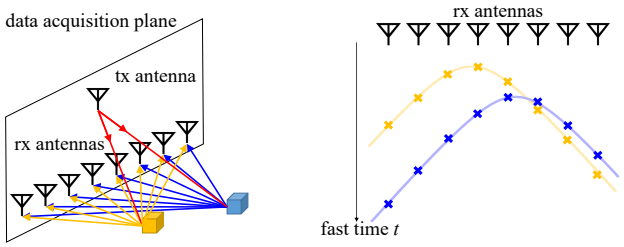


Figure 11: Signal-reflecting objects fall on hyperbolas.

However, the sensing performance will suffer from issues such as high sidelobes if we implement the algorithm using this naive geometric approach. Denoting the AoA and AoD of the m-th Tx and n-th Rx antennas by θ_n and ϕ_m, and the distances from the target to these respectively antennas by d_m^{t} and d_n^{r}, a more rigorous algorithm computing the scalar wavefront intensity $\iota(\phi_m, \theta_n, d_m^{\text{t}}, d_n^{\text{r}}, \tau)$ can be derived from the wave equation [68]:

$$\iota(\phi_m, \theta_n, d_m^{\text{t}}, d_n^{\text{r}}, \tau) = \frac{4}{\cos\phi_m \cos\theta_n} \times \left[\frac{d_m^{\text{t}} d_n^{\text{r}}}{c^2} \frac{\partial^2 s_{mn}(t)}{\partial t^2} + \frac{d_m^{\text{t}} + d_n^{\text{r}}}{c} \frac{\partial s_{mn}(t)}{\partial t} + s_{mn}(t) \right] \Bigg|_{t=\tau+\Gamma_{mn}}, \tag{6}$$

where $s_{mn}(t)$ is the Rx signal strength at time t, τ denotes the starting time, and $\Gamma_{mn} = \frac{d_m^{\text{t}} + d_n^{\text{r}}}{c}$ is the concerned time delay. The partial derivative terms come from differentiation along the surface normal when deriving the solution of a homogeneous wave equation [7].

The algorithm finally estimates the reflection intensity of an arbitrary point (x, y, z) at a given time τ. Assuming a pair of Tx-Rx antennas positioned at $(x_m, y_m, 0)$ and $(x_n, y_n, 0)$, the tuple $(\phi_m, \theta_n, d_m^{\text{r}}, d_n^{\text{r}})$ in Eq. (6) can be substituted by $\cos\phi_m = z/d_m^{\text{t}}$, $\cos\theta_n = z/d_n^{\text{r}}$, $d_m^{\text{t}} = \sqrt{(x-x_m)^2 + (y-y_m)^2 + z^2}$, as well as $d_n^{\text{r}} = \sqrt{(x-x_n)^2 + (y-y_n)^2 + z^2}$. Therefore, $\iota(x, y, z, \tau)$ is obtained by summing up the converted version of Eq. (6) over all antenna pairs.

$$\iota(x, y, z, \tau) = \sum_{n=1}^{N} \sum_{m=1}^{M} \iota(x, y, z, \tau, x_m, y_m, x_n, y_n). \tag{7}$$

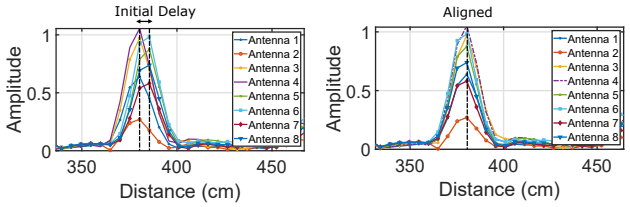
A straightforward application of this general algorithm is RF imaging, which projects the intensity $\iota(x, y, z, \tau)$ to the pixel (x, y) on a 2D image plane. Moreover, Octopus can use this algorithm to scan an unknown indoor space, similar to [34] but in a more energy efficient manner. Since traversing the whole space is computationally expensive, we fall back to the EM algorithm in Section 5.2 for angle and range estimations when the number of targets is known.

5.4 Array Calibration

The main purpose of calibration is to obtain the time delay or phase mismatches caused by hardware. In particular, unlike narrowband array calibration that only phase align antennas of an array to that of a reference antenna, wideband array calibration needs to handle two parts: delay offsets and phase offsets. Considering the MIMO model in Section 5.1, we further have:

$$y_{i,k,n,m}^{\text{hw}}(t) = e^{j\omega_{i,n}} y_{i,k,n,m}^{b}(t - \tau_{i,n}^{\text{rx}}) \tag{8}$$

where $\tau_{i,n}^{\text{rx}} = \tau_{0,0}^{\text{rx}} + \Delta\tau_{i,n}^{\text{init}}$, with $\tau_{0,0}^{\text{rx}}$, $\Delta\tau_{i,n}^{\text{init}}$, and $\omega_{i,n}$ respectively referring to the delay of the reference antenna (e.g., the first antenna of the first RF chain), the initial delay offset and phase; they are caused by hardware at the i-th RF receiving chain and the n-th antenna. To perform array calibration, a metal reflector is positioned in the far-field (3.8 m and 0° direction in our case) of the antenna array. Since the round-trip distances for different antenna pairs are approximately the same, we align each antenna with $\Delta\tau_{i,n}^{\text{init}}$ and $\omega_{i,n}$ to the reference antenna. The results before and after calibrations are shown in Figure 12. Due to the finite range resolution, the initial time delays $\{\Delta\tau_{i,n}^{\text{init}}\}$ are rounded to two discrete distances (380 cm and 385 cm) when plotting only the magnitudes of baseband signals in Figure 12a, whereas finer-grained distance information is further conveyed by carrier phases $\{\omega_{i,n}\}$. Thanks to the high quality of the



(a) Without calibration. (b) With calibration.

Figure 12: While the initial delays cause offsets in all antennas before calibration (a), all reference distances of eight antennas are well aligned after calibration (b).

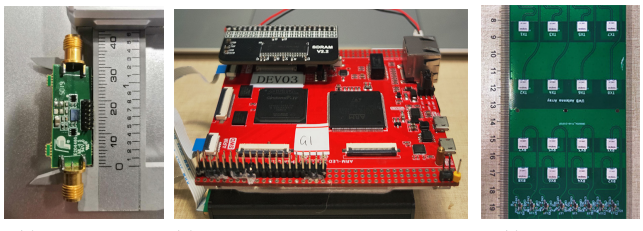
Phase Lock Loop (PLL) in X4, we only need to perform a one-time calibration for all these quantities. Note that the minor randomness caused by an RF switch does not affect the calibration at all, because the interactions between a certain pair of Tx-Rx antennas take place only after the Rx antenna is switched on, as explained in Figure 4.

6 EVALUATION

We evaluate the performance of our Octopus prototype, whose full image and individual components are respectively shown in Figures 2 and 13, by conducting experiments in a 30 m^2 lab. The prototype mainly consists of three parts: the transceiver, the RF switches, and the digital processing platform. Up to 16×16 Tx-Rx antennas are used in the experiments; though the default 8×8 antenna array has been designed as a rectangular shape (Figure 13c), multiple arrays can be supported by our hybrid mode (see Figure 3d). In addition, we may use detachable antennas to form other array shapes (e.g, linear or L-shape). This compact and monostatic design makes Octopus readily deployable in real-life scenarios. The cost of building this prototype is less than 300 USD, only about one-tenth of the SDR-based design [1–3], rendering wideband RF sensing much more affordable. In the lab, Octopus is placed against a wall and one meter above the ground to get a large field of view (FoV). Background objects (e.g., walls, desks, chairs, and cabins) remain static when conducting experiments, and this static background is subtracted later for detection of the real target.

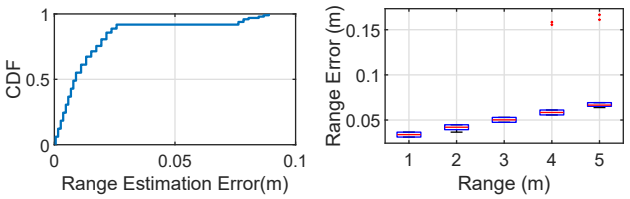
6.1 Micro-benchmarking

We start the evaluations by benchmarking several basic aspects of Octopus, including range/angle estimations, the effect of the number of antennas, the effect of antenna element spacing, and



(a) Transceiver. (b) ARM-FPGA processing module. (c) Antennas.

Figure 13: Octopus hardware components.



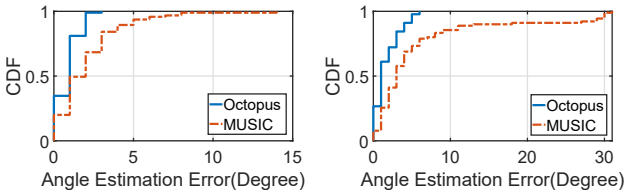
(a) Overall error. (b) Errors on different ranges.

Figure 14: Range estimation evaluation.

the performance of the AI library. To complete these tasks, Octopus simply retrieves the raw RF readings from its transceiver (via FPGA) and processes them in ARM, except the last aspect where computations in both FPGA and ARM are involved.

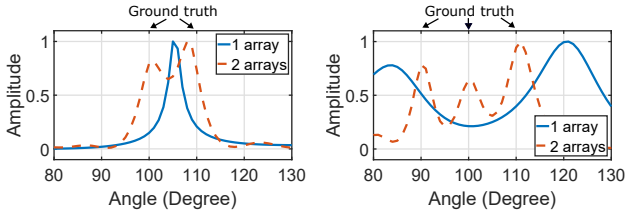
6.1.1 Range Estimation. Octopus uses the algorithm described in Section 5.2 for range estimation. We obtain the range accuracy by comparing the estimated ranges of a solid metal block against the ground truth manually set in advance. As shown in Figure 14a, the median range estimation error of Octopus is only 0.01 m and 90% of the errors are below 0.03 m. The error also increases slightly with an increasing range, as shown in Figure 14b. Therefore, the general performance of Octopus in ranging can be deemed as excellent. Nonetheless, because Octopus has a resolution of about 10 cm (see Δr in Section 3.1), we still observe a long tail (albeit minor) in the CDF of the range estimation error, which may be a concern when performing precise measurements. Note that the 5 m maximum range is confined by the Tx power of X4, and we have not applied a power amplifier in this experiment.

6.1.2 AoA/AoD Estimations. Octopus leverages the power of its MIMO antenna array to estimate AoA and AoD, again using the algorithm explained in Section 5.2. To evaluate the performance of AoA/AoD estimations, we manually set a solid metal block at various angles with respect to the normal vector of the antenna face. As shown in Figure 15, the AoA estimation errors of Octopus are all below 3° (with a median value of only 1°), but the MUSIC algorithm (commonly used for narrowband estimation [58]) suffers from a very large variance with its maximum error almost reaching 15°. A similar situation takes place for AoD estimation, where the maximum error of MUSIC is more than 3 times of Octopus. We have tried other conventional narrowband algorithms (e.g., MVDR [11] and ESPRIT [44]), but they fail to produce meaningful estimations as they cannot handle wideband scenarios. The step-function shape of the CDFs stems from the 1° granularity configured for our algorithm to reduce complexity.



(a) AoA estimation error. (b) AoD estimation error.

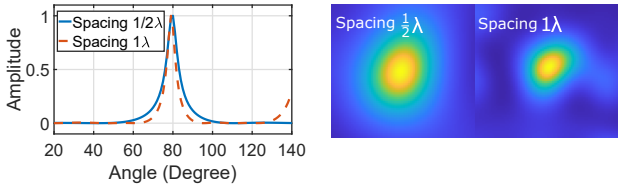
Figure 15: Angle estimation evaluation.



(a) Signals from 2 directions. (b) Signals from 3 directions

Figure 16: Multi-array improvement.

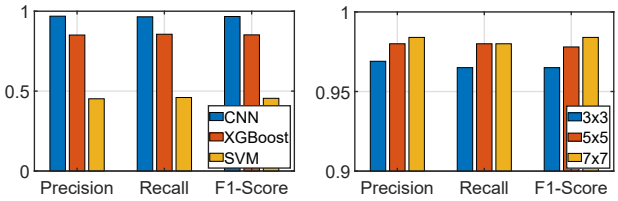
6.1.3 Effect of Multiple Arrays. Increasing the number of antenna elements in the array enables Octopus to increase angular resolution, and this can be achieved by once adding a whole array of 8 Rx antennas (only 1 Tx antenna is needed). To showcase the enhanced resolution of Octopus with multiple arrays, we place multiple reflectors in front of the antenna array and verify if they can be differentiated. In Figure 16a, two reflectors are placed at 100° and 110°, and in Figure 16b, three reflectors are placed at 90°, 100° and 110°. We test Octopus with 1 array (8 antennas) and 2 arrays (16 antennas) to estimate the AoAs of the signals echoed from the reflectors. When there is only 1 array, the reflectors cannot be differentiated, since the angular resolution of 1 array is not enough to separate signals reflected from close-by reflectors. Fortunately, adding another array apparently allows the reflectors to be differentiated. Therefore, Octopus is a flexible MIMO sensing platform that is readily extensible if we demand better performance.



(a) 1D angle estimation. (b) 2D angle estimation.

Figure 17: Effects of antenna spacing.

6.1.4 Effect of Antenna Element Spacing. By increasing the spacing between the antenna elements, we can increase the aperture of the antenna array, and hence enhance the accuracy of angle estimation. However, increasing the antenna spacing will reduce the spatial sampling rate, hence causing spatial aliasing (i.e., multiple peaks in the angular domain for a single target) if the antenna spacing is larger than 1/2 wavelength. As our default antenna array has a fixed antenna spacing, we resort to detachable antennas to obtain a variable spacing. Figure 17a illustrates two cases where the spacing between antenna elements are 1/2 wavelength and 1 wavelength of the center frequency. It can be seen that Octopus with 1 wavelength antenna spacing has a sharper peak, suggesting a higher accuracy. However, it does introduce an aliasing component. Similar results can also be observed in Figure 17b showing the result of 2D angle estimation with the same two antenna element spacings. The rule of thumb for choosing the spacing is that, if we have prior knowledge of the angular position of the target, (e.g., RF imaging for an object facing the antenna array), we can make the spacing between the antenna elements larger. Otherwise, if we perform angle estimation or localization, we should make the spacing between the antenna elements smaller.



(a) Different classifiers. (b) Different kernel sizes.

Figure 18: Evaluation of the AI modules.

6.1.5 AI Library. There is a strong trend in both industry and academia to develop more high-level applications on top of RF sensing platforms. This further entails the need for employing AI modules (including various machine learning algorithms and deep neural network models) to extract insights from RF sensing data, forming the ground from which various applications can be built. To cater to this need, Octopus offers an embedded AI library. Due to the page limit, we hereby use one application, human activity recognition (HAR), to evaluate three AI classifiers in the library, namely CNN [32], XGBoost [12] and SVM [50], with CNN implemented in FPGA and another two in ARM. To evaluate the performance of these classifiers on HAR, we let 20 people performing seven activities: bending, falling, lying down, standing up, sitting down, squatting down, and walking. The precision, recall, and F1 score of these classifiers are shown in Figure 18a. As CNN performs the best, we also study the impact of kernel size in Figure 18b.

6.1.6 Power Consumption and Cost Breakdown. To benchmark Octopus's power usage, we conduct multiple measurements to derive the typical power consumption of its individual components, and the detailed breakdown is shown in Table 2. One may readily observe that each component consumes only a few hundreds of mW and the total power consumption is capped below 1W. Such a low-power profile goes very close to a smartphone running its screen, it hence surely allows for battery-powered edge computing deployments. Alongside the power consumption, we also provide a rough cost breakdown in terms of these components in the table; the total budget below 300USD could be further reduced should we go for a large-scale production.

Table 2: Power Consumption and Cost Breakdown.

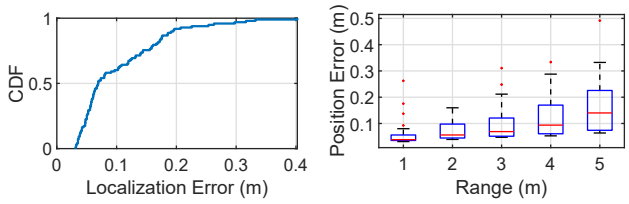
	Power (mW)	Cost (USD)
RF front-end	201.08	180.00
FPGA	313.53	35.00
ARM	289.10	18.00
Peripheral	91.50	28.00

6.2 Case Studies

We perform case studies to evaluate the performance of Octopus in real-life applications. These applications include localization, trajectory tracking, motion detection, vital signs monitoring, occupancy awareness, and RF imaging. Since these applications all

require parameter fine-tuning by individual users, their main functions are implemented in ARM, but they leverage basic modules in FPGA (e.g., FFT and CNN) for vital signs monitoring and occupancy awareness. Due to the lack of compact implementation of the RF imaging algorithm, the imaging results are obtained by offloading baseband RF signals to a host PC and processing them with Matlab. Therefore, apart from the offline RF imaging, all the other applications operate in an online manner.

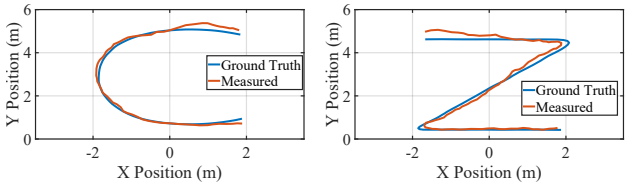
6.2.1 Localization. By combining range and angle information (see Sections 6.1.1 and 6.1.2), a unique point in the space can be identified and thus the target can be located. As explained earlier, Octopus simply relies on its transceiver to gather baseband RF signals and then performs online computations in ARM to infer locations. To evaluate the localization performance of Octopus, we use a solid metal block to emulate a target, and perform 200 experiments at various locations (whose distances towards Octopus are set so that they vary between 1m and 5m with a stepsize of 1m). In this evaluation, each experiment lasts for 60 s. Since the frame rate of Octopus is 30frame/s, and Octopus has 16 antennas in total, we get a total of 360,000 measurements. The overall localization error is presented as CDF in Figure 19a. It can be seen that the median localization error is only 0.067 m, and 90% of the error is below 0.2m. In addition, we show the localization errors at different ranges in Figure 19b. The decreasing accuracy in range attributes to the signal attenuation that follows an inverse-square law as the range increases. For typical indoor scenarios, a single Octopus may reach up to 5 m, with a median error of 0.13 m. We could extend the system coverage by employing multiple Octopus prototypes. Although the underlying algorithm is similar to many RF-based localization proposals [2, 3, 16, 62], using the wideband radio has endowed Octopus with a very high accuracy, while its compact design has made the implementation very efficient.



(a) Localization error. (b) Errors on different ranges.

Figure 19: Localization evaluation.

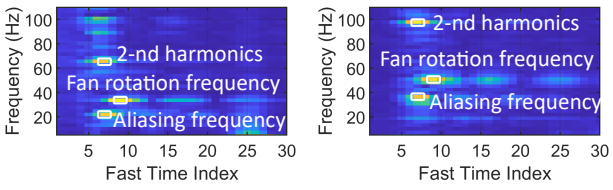
6.2.2 Trajectory Tracking. Octopus is able to not only locate a target, but also to keep track of target in motion. Target trajectory tracking can be seen as an extension of localization, as the target is moving and the position has to be updated constantly. There are some interesting applications such as tracking human movement [10], recording a person's writing in air [1], and recording a person's writing on blackboard [14]. In this section, we study the trajectory tracking performance of Octopus by tracking a person walking with a constant speed. The person walks according to preset paths, whose ground truth trajectories are measured beforehand. Octopus uses its localization function to infer and record the person's position every 0.03 s, then it applies a Kalman filter (in the DSP library) to correct the walking trajectory. To maintain the brevity of our presentation, we only give two examples in Figure 20, where the person walks along two trajectories of letters "C" and "Z", respectively. We observe that the measured trajectories and the ground truth are very close, confirming the proficiency of Octopus in target trajectory tracking. We refrain from comparing Octopus's tracking accuracy with existing proposals in the literature, as the outcome is rather obvious: accurate localization surely leads to accurate tracking.



(a) Walking in "C". (b) Walking in "Z".

Figure 20: Tracking human walking trajectory.

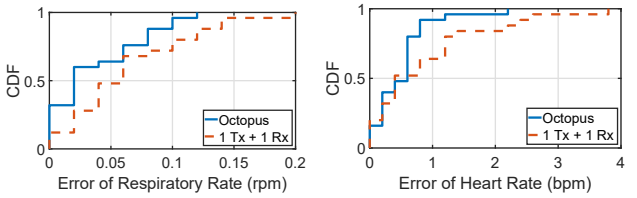
6.2.3 Motion Detection. Besides estimating target positions, Octopus also excels in detecting Doppler shift. To evaluate this ability, we employ Octopus to measure the rotational speed of an electric fan. Electric fans use Pulse Width Modulation (PWM) to control rotation speed, and the fan we employ has two speed levels: low speed (33 Hz) and high speed (50 Hz). By observing the first- and second-order harmonics, the frequencies of low speed and high speed are estimated to be 33 Hz and 49.5 Hz respectively. One may believe that the first-order harmonics alone would suffice, but we cannot remove the aliasing frequency (visual effect of "inverse rotation") without verifying against the second-order harmonics. This demonstration is rough as Octopus simply stacks multiple frames of the baseband signals and performs column-wise FFT (in the DSP library). We leave further developments of specific analytical tools for interpreting such "RF images" to future users of Octopus.



(a) Low speed. (b) High speed

Figure 21: Doppler heatmaps of an electric fan at two different rotational speeds.

6.2.4 Vital Signs Monitoring. To further evaluate the fine-grained Doppler detection capability of Octopus, we employ Octopus to measure human vital signs, i.e., respiratory rate and heart rate. We let a test subject sit down at 1 m distance from Octopus and perform 100 measurements. Octopus leverages Variational Mode Decomposition (VMD) [19] (in the AI library) to decompose the baseband signal into to its respiratory and heartbeat components and then relies on FFT (in the DSP library) to extract the respiratory and heart rates. The ground truth is obtained by NeuLog respiration

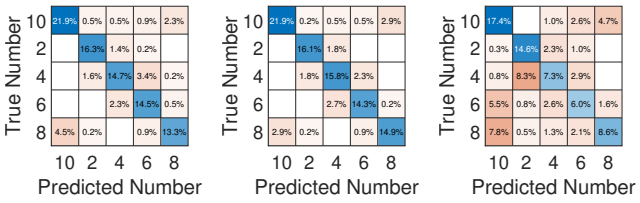


(a) Respiratory rate errors. (b) Heart rate errors.

Figure 22: Vital signs estimation errors.

monitor belt logger sensor NUL-236 [37] and Heal Force PC-80B portable ECG monitor [22] for breath and heartbeat, respectively. We show the CDF plots of vital signs estimation errors of Octopus in Figure 22a and Figure 22b, while comparing with Octopus using only one pair of antennas. It can be observed that Octopus achieves a median respiratory rate error of 0.02rpm (Respirations Per Minute) and 90% of the heart rate errors below 1 bpm (Beats Per Minute), much better than the simplified version with a median respiratory rate error at 0.04rpm and a 90-th percentile heart rate error beyond 2 bpm. As this application involves the whole suite of Octopus hardware, we choose it as one of the software demonstrations publicized at Github [38]. We also refer readers to [15, 66] for more extensive vital signs monitoring enabled by Octopus.

6.2.5 Occupancy Awareness. Thanks to the high spatial resolution of the UWB signal and the high angular resolution of the MIMO antenna array, Octopus has the ability to detect multiple targets. To evaluate the multi-object detection performance of Octopus, we choose occupancy awareness as a test. Occupancy awareness [17, 47, 60] is an active research area thanks to its potential applications in smart environment, so it should be a relevant application of Octopus. In this evaluation, we let up to 10 people standing at random locations in a $5 \times 5\,\mathrm{m}^2$ area, while Octopus gathers reflected signal from the crowd, whose quantity varies from 2 to 10 with a step size of 2. We train three machine learning modules to infer the occupancy, namely random forest, XGBoost, and CNN. The results show that the precision scores are 81%, 83%, and 53%, the recall scores are 81%, 83%, and 54%, and the f1-score are 81%, 83%, and 53% for random forest, XGBoost, and CNN model, respectively. We also report the confusion matrices of these modules in Figure 23. Although this evaluation does demonstrate the decent ability of Octopus in detecting multiple targets, the performance of occupancy inference is below our expectation (especially for the CNN model). We attribute this to the insufficient capacity of our compressed CNN model (necessary to fit it into FPGA), which we shall continue improving in follow-up studies. The codes of this application are also chosen to be publicized at Github [38].

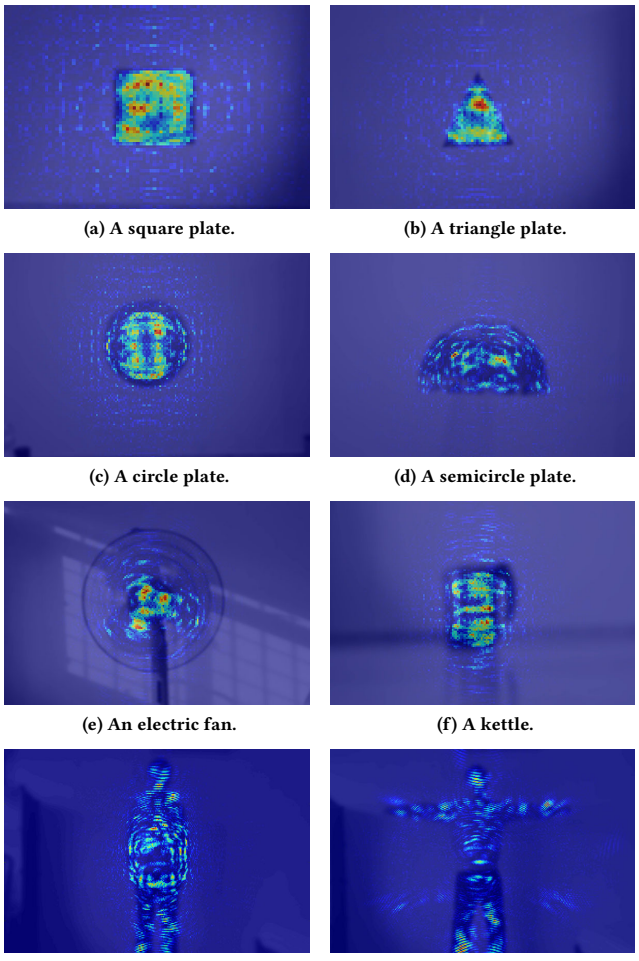


(a) Random forest. (b) XGBoost. (c) CNN.

Figure 23: Confusion matrices of occupancy inference.

6.2.6 RF Imaging. In this section, we showcase the wideband RF imaging capabilities of Octopus, with the help of the general sensing algorithm in Section 5.3. Leveraging the high temporal and spatial resolution of the radio and corresponding analyzing algorithm, Octopus can generate images of higher resolution than previous solution based on COTS devices such as Wi-Fi [24]. To validate our claim, we use two default antenna arrays to form a 16×16 array for Octopus, then we use it to image different objects placed in front of the antenna array. In reporting the results, we overlay the (deliberately blurred) object photo behind an RF image whose bright areas indicate the concentration of reflected energy. We interpret these RF images on three aspects:

(1) *Shape recognition*: We perform imaging of solid metal plates with a diameter of roughly 40 cm at a 2 m distance from the antenna array. By comparing the square in Figure 24a, triangle in Figure 24b, circle in Figure 24c, and semicircle in



(a) A square plate. (b) A triangle plate.
(c) A circle plate. (d) A semicircle plate.
(e) An electric fan. (f) A kettle.
(g) A standing person. (h) A posing person.

Figure 24: Sample RF images obtained by Octopus.

Figure 24d, it is clear that shapes of the targets can be readily recognized. This is in stark contrast with previous imaging solution based on COTS devices [24], which only generates obscure images without clear-cut shapes. Furthermore, our commercial-grade Octopus matches the SDR-based solutions [1] in performance.

(2) *Object recognition*: To illustrate that Octopus can perform imaging of common objects, we place an electric fan and a kettle 1 m away from the antenna array. The imaging results of these two objects are shown in Figure 24e and Figure 24f. It can be seen that the RF images characterize distinct contours of the corresponding targeted objects.

(3) *Pose estimation:* Octopus can capture different human poses although the human body is only a weak reflector. To demonstrate this, our human subject stands 1.5 m away from the antenna array. Figure 24h and Figure 24g show RF images of this person standing straight and standing with arms outstretched, respectively. In these images, the torso, head, and limbs of the human can be clearly observed.

To summarize, all these results confirm that the wideband MIMO sensing capabilities, together with the algorithm in Section 5.3, enable Octopus to perform static RF imaging with previously unobtainable precision on such a compact implementation. Note that, as a wide band radar system, Octopus is certainly capable of performing dynamic RF imaging by exploiting either the Doppler effect [29] or signal strength fluctuation [30], but we leave this task to interested future users of Octopus.

7 RELATED WORK

Existing proposals on RF sensing platforms can be roughly categorized into three classes: i) Wi-Fi based solutions [20, 46, 57], ii) SDR based testbeds [1–3] and iii) specially designed hardware [26]. We will refrain from discussing the industry-driven mmWave-based technologies [35, 52, 65], as their substantially different frequency bands have led to sensing capabilities largely orthogonal to what Octopus can achieve.

Wi-Fi based solution: To the best of our knowledge, Phaser [20] is the first sensing platform solely based on commodity Wi-Fi, following the seminal works in [45, 46, 58] that all leverage CSI information for sensing (localization in particular). To improve the quality of CSI information, Atheros CSI tool [57] is developed to achieve a precisely spliced CSI, and a frequency combining algorithm is proposed in ToneTrack [59] to increase the effective bandwidth. Later proposals [42, 48, 53] reduce system complexities by utilizing only a single Wi-Fi link or access point (AP), while multiple APs are explored to improve localization performance [51]. CrossSense [64], EI [28], WiPose [29], and Person-in-WiFi [54] further shed light on large-scale Wi-Fi sensing from deep learning perspective. Zhang et al., in their latest proposals [63], leverage the Fresnel model to explore the theoretical limit of Wi-Fi sensing. Last but not least, the through-wall sensing capability of Wi-Fi has been exploited for various applications such as occupancy awareness and imaging [17, 30]. While achieving significant progresses in RF sensing, Wi-Fi based solutions suffer low resolution and they are also largely handicapped by their default communication functions, as we have explained in Section 1.

SDR-based sensing platforms: These platforms move some radio components traditionally implemented in hardware to software, allowing developers and researchers to quickly design and prototype RF solutions. The authors in [1–3] adopt USRP (an SDR platform) to implement FMCW-based RF sensing not available on commodity devices. Although these SDR-based sensing testbeds are powerful and flexible, they may hardly be applied in practice due to its large size and cost: each of them costs above 2,000 USD. According to our experience with Octopus, integrating the SDR-based testbeds into commodity hardware still faces a lot of practical difficulties. In parallel with the development reported in this paper, we have also worked out an FMCW version [56] (similar to [2] but far more compact) and an acoustic sensing version [8, 9]; they share similar computation modules as Octopus but compliment it in application scope with different transceivers.

Dedicated sensing products: Apart from academic endeavors, industrial developers have also come up with commercial RF sensing products. A typical one is Walabot [26], claiming useful for in-wall scanning and fall detection. However, these products are not relevant to researchers, as their proprietary techniques (e.g. baseband property and digital processing) are unknown. Although advertised as "programmable", Walabot only provides few APIs and the algorithms are not customizable. Moreover, the intended applications of these platforms are very limited (e.g., near-range imaging and coarse-grained human monitoring). These have made their extensibility very questionable. In fact, we have already leveraged Octopus to develop novel applications similar to what Walabot has claimed, but our developments are highly customizable and can thus accommodate innovative deep learning modules [67].

8 CONCLUSION

In this paper, we present the design, implementation and evaluation of Octopus, a wideband MIMO sensing platform built with cheap COTS hardware. This is a general-purpose platform facilitating quick implementation and validation of a broad range of RF sensing designs, while enabling straightforward application deployments for edge intelligence. Therefore, we believe that Octopus will benefit not only the research community but also the industrial development. The novelty of our platform spans across both hardware and software, including i) wideband MIMO radio front-end for versatile sensing, ii) heterogeneous ARM-FPGA architecture for reconfigurable processing, iii) DAG-based programming model for convenient developments, and iv) UWB-compatible algorithm for high-resolution sensing. We conduct comprehensive micro-benchmarking to validate the effectiveness and high performance of Octopus, and we also use several case studies to demonstrate its wide applicability and remarkable flexibility in enabling a variety of sensing applications including localization, tracking, motion detection, vital sign monitoring, occupancy awareness, and imaging.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their valuable and constructive comments. This work was partly granted by the National Natural Science Foundation of China (Project No. 61971145), and we also thank Energy Research Institute @ NTU (ERI@N) and NTU-IGP for supporting the PhD scholarship of Tianyue Zheng.

REFERENCES

[1] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. 2015. Capturing the Human Figure through a Wall. *ACM Trans. Graph.* 34, 6 (2015), 1–13.

[2] Fadel Adib, Zachary Kabelac, and Dina Katabi. 2015. Multi-Person Localization via RF Body Reflections. In *Proc. of USENIX NSDI*. 279–292.

[3] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C. Miller. 2014. 3D Tracking via Body Radio Reflections. In *Proc. of USENIX NSDI*. 317–329.

[4] Analog Devices. 2020. HMC321 Switch. https://www.analog.com/media/en/technical-documentation/data-sheets/hmc321a.pdf.

[5] Novelda AS. 2021. X4 Radar. https://novelda.com/ or https://www.etesters.com/catalog/BC0629DC-0AD4-4C65-896E-71B099764DD9/novelda-as.

[6] B. A. Bjerke. 2011. LTE-Advanced and the Evolution of LTE Deployments. *IEEE Wireless Communications* 18, 5 (2011), 4–5.

[7] Max Born and Emil Wolf. 1999. *Principles of Optics*. Cambridge University Press.

[8] Chao Cai, Zhe Chen, Henglin Pu, Liyuan Ye, Menglan Hu, and Jun Luo. 2020. AcuTe: Acoustic Thermometer Empowered by a Single Smartphone. In *Proc. of the 18th ACM SenSys*. 28–41.

[9] Chao Cai, Henglin Pu, Menglan Hu, Rong Zheng, and Jun Luo. 2021. Acoustic Software Defined Platform: A Versatile Sensing and General Benchmarking Platform. *IEEE Transactions on Mobile Computing* (2021), 1–15.

[10] Chao Cai, Henglin Pu, Peng Wang, Zhe Chen, and Jun Luo. 2021. We Hear Your PACE: Passive Acoustic Localization of Multiple Walking Persons. In *Proc. of the 23rd ACM UbiComp*. 55:1–24.

[11] Jack Capon. 1969. High-Resolution Frequency-Wavenumber Spectrum Analysis. *Proc. of the IEEE* 57, 8 (1969), 1408–1418.

[12] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proc. of the ACM SIGKDD*. 785–794.

[13] Zhe Chen, Chao Cai, Tianyue Zheng, Jun Luo, Jie Xiong, and Xin Wang. 2021. RF-Based Human Activity Recognition Using Signal Adapted Convolutional Neural Network. *IEEE Transactions on Mobile Computing* (2021), 1–13.

[14] Zhe Chen, Zhongmin Li, Xu Zhang, Guorong Zhu, Yuedong Xu, Jie Xiong, and Xin Wang. 2017. AWL: Turning Spatial Aliasing From Foe to Friend for Accurate WiFi Localization. In *Proc. of the ACM CoNEXT*. 238–250.

[15] Zhe Chen, Tianyue Zheng, Chao Cai, and Jun Luo. 2021. MoVi-Fi: Motion-robust Vital Signs Waveform Recovery via Deep Interpreted RF Sensing. In *Proc. of the 27th ACM MobiCom*. 1–14.

[16] Zhe Chen, Guorong Zhu, Sulei Wang, Yuedong Xu, Jie Xiong, Jin Zhao, Jun Luo, and Xin Wang. 2021. M^3: Multipath Assisted Wi-Fi Localization with a Single Access Point. *IEEE Transactions on Mobile Computing* 20, 2 (2021), 588–602.

[17] Saandeep Depatla and Yasamin Mostofi. 2018. Crowd Counting Through Walls Using WiFi. In *Proc. of the 16th IEEE PerCom*. 1–10.

[18] Shuya Ding, Zhe Chen, Tianyue Zheng, and Jun Luo. 2020. RF-Net: A Unified Meta-Learning Framework for RF-enabled One-Shot Human Activity Recognition. In *Proc. of the 18th ACM SenSys*. 517–530.

[19] Konstantin Dragomiretskiy and Dominique Zosso. 2013. Variational Mode Decomposition. *IEEE Trans. on Signal Processing* 62, 3 (2013), 531–544.

[20] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. 2014. Phaser: Enabling Phased Array Signal Processing on Commodity WiFi Access Points. In *Proc. of ACM MobiCom*. 153–164.

[21] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11n Traces with Channel State Information. *SIGCOMM Comput. Commun. Rev.* 41, 1 (2011), 53–53.

[22] Heal Force. 2020. Easy ECG Monitor – Prince-180B (B0). http://www.healforce.com/en/html/products/portableecgmonitors/healthcare-portable-ECG-monitors-Prince-180B-B0.html.

[23] Maurice Herlihy and Nir Shavit. 2011. *The Art of Multiprocessor Programming*. Morgan Kaufmann.

[24] Donny Huang, Rajalakshmi Nandakumar, and Shyamnath Gollakota. 2014. Feasibility and Limits of Wi-Fi Imaging. In *Proc. of the 12nd ACM Sensys*. 266–279.

[25] IEEE. 2020. 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 Enhancements for High Efficiency WLAN. *IEEE P802.11ax/D7.0, September 2020* (2020), 1–822.

[26] Vayyar Imaging. 2020. Walabot. https://walabot.com/.

[27] Intel. 2020. Cyclone IV Device Handbook. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf.

[28] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proc. of ACM MobiCom*. 289–304.

[29] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Wang Shiyang, Lin Sen, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. 2020. Towards 3D Human Pose Construction Using WiFi. In *Proc. of ACM MobiCom*. 23:1–14.

[30] Chitra R. Karanam and Yasamin Mostofi. 2017. 3D Through-Wall Imaging with Unmanned Aerial Vehicles Using WiFi. In *Proc. of the 16th ACM/IEEE IPSN*. 131–142.

[31] Francesco Laghezza, Feike Jansen, and Jeroen Overdevest. 2019. Enhanced Interference Detection Method in Automotive FMCW Radar Systems. In *2019 20th International Radar Symposium (IRS)*. IEEE, 1–7.

[32] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. 1997. Face Recognition: A Convolutional Neural-Network Approach. *IEEE Trans. on Neural Networks* 8, 1 (1997), 98–113.

[33] Chun-Xun Lin, Tsung-Wei Huang, Guannan Guo, and Martin D. F. Wong. 2019. A Modern C++ Parallel Task Programming Library. In *Proc. of the ACM MM*. 2284–2287.

[34] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A. Stankovic, Niki Trigoni, and Andrew Markham. 2020. See through Smoke: Robust Indoor Mapping with Low-Cost mmWave Radar. In *Proc. of the 18th ACM MobiSys*. 14–27.

[35] Mistral Solutions Pvt. Ltd. 2020. mmWave Technology. https://www.mistralsolutions.com/product-engineering-services/expertise/mmwave-radar-modules/.

[36] Mouser Electronics. 2020. Phase Detectors / Shifters. https://www.mouser.com/Semiconductors/Integrated-Circuits-ICs/Wireless-RF-Integrated-Circuits/Phase-Detectors-Shifters/_/N-73tyl.

[37] NeuLog. 2020. Respiration Monitor Belt Logger Sensor NUL-236. https://neulog.com/respiration-monitor-belt/.

[38] Octopus. 2021. https://github.com/DeepWiSe888/Octopus.

[39] Joan Palacios, Daniel Steinmetzer, Adrian Loch, Matthias Hollick, and Joerg Widmer. 2018. Adaptive Codebook Optimization for Beam Training on Off-the-Shelf IEEE 802.11 ad Devices. In *Proc. of ACM MobiCom*. 241–255.

[40] Barry Peter and Crowley Patrick. 2012. *Modern Embedded Computing*. Elsevier.

[41] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-Home Gesture Recognition Using Wireless Signals. In *Proc. of ACM MobiCom*. 27–38.

[42] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. 2018. Widar2.0: Passive Human Tracking with a Single Wi-Fi Link. In *Proc. of ACM MobiSys*. 350–361.

[43] B. Radunovic and J.-Y. Le Boudec. 2004. Optimal Power Control, Scheduling, and Routing in UWB Networks. *IEEE Journal on Selected Areas in Communications* 22, 7 (2004), 1252–1270.

[44] Richard Roy and Thomas Kailath. 1989. ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques. *IEEE Trans. on Acoustics, Speech, and Signal Processing* 37, 7 (1989), 984–995.

[45] Souvik Sen, Romit Roy Choudhury, Bozidar Radunovic, and Tom Minka. 2011. Precise Indoor Localization Using PHY Layer Information. In *Proc. of ACM SIGCOMM*. 1–6.

[46] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. 2013. Avoiding Multipath to Revive Inbuilding WiFi Localization. In *Proc. of ACM MobiSys*. 249–262.

[47] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. 2017. Poster: Occupancy State Detection using WiFi Signals. In *Proc. of ACM MobiSys*. 161–161.

[48] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. 2018. Multipath Triangulation: Decimeter-Level WiFi Localization and Orientation with a Single Unaided Receiver. In *Proc. of ACM MobiSys*. 376–388.

[49] STMicroelectronics. 2020. Arm® Cortex®-M7 32b MCU+FPU. https://www.st.com/resource/en/datasheet/stm32f767ig.pdf.

[50] Johan AK Suykens and Joos Vandewalle. 1999. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters* 9, 3 (1999), 293–300.

[51] Tzu-Chun Tai, Kate Ching-Ju Lin, and Yu-Chee Tseng. 2019. Toward Reliable Localization by Unequal AoA Tracking. In *Proc. of ACM MobiSys*. 444–456.

[52] Texas Instruments Incorporated. 2020. mmWave Sensors: Intelligent Autonomy at the Edge with Single-Chip Millimeter-Wave Sensors. http://www.ti.com/sensors/mmwave/overview.html.

[53] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-Level Localization with a Single WiFi Access Point. In *Proc. of USENIX NSDI*. 165–178.

[54] Fei Wang, Sanping Zhou, Stanislav Panev, Jinsong Han, and Dong Huang. 2019. Person-in-WiFi: Fine-Grained Person Perception Using WiFi. In *Proc. of IEEE ICCV*. 5452–5461.

[55] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proc. of ACM MobiCom*. 65–76.

[56] WiRUSH/AIWiSe. 2019. Guangxi Wanyun and Guangzhou AIWiSe Technology Co., Ltd. https://www.wirush.ai and https://aiwise.wirush.ai.

[57] Yaxiong Xie, Zhenjiang Li, and Mo Li. 2015. Precise Power Delay Profiling with Commodity WiFi. In *Proc. of ACM MobiCom*. 1342–1355.

[58] Jie Xiong and Kyle Jamieson. 2013. ArrayTrack: A Fine-Grained Indoor Location System. In *Proc. of USENIX NSDI*. 71–84.

[59] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. 2015. Tonetrack: Leveraging Frequency-Agile Radios for Time-Based Indoor Wireless Localization. In *Proc. of ACM MobiCom*. 537–549.

[60] Yanbing Yang, Jie Hao, Jun Luo, and Jialin Pan. 2017. CeilingSee: Device-Free Occupancy Inference through Lighting Infrastructure based LED Sensing. In *Proc. of IEEE PerCom*. 247–256.

[61] Youwei Zeng, Dan Wu, Jie Xiong, Enze Yi, Ruiyang Gao, and Daqing Zhang. 2019. FarSense: Pushing the Range Limit of WiFi-based Respiration Sensing with CSI Ratio of Two Antennas. *Proc. of the ACM UbiComp* 3, 3 (2019), 1–26.

[62] Chi Zhang, Feng Li, Jun Luo, and Ying He. 2014. iLocScan: Harnessing Multipath for Simultaneous Indoor Source Localization and Space Scanning. In *Proc. of the 12th ACM SenSys*. 91—104.

[63] Fusang Zhang, Kai Niu, Jie Xiong, Beihong Jin, Tao Gu, Yuhang Jiang, and Daqing Zhang. 2019. Towards a Diffraction-based Sensing Approach on Human Activity Recognition. *Proc. of the ACM UbiComp* 3, 1 (2019), 1–25.

[64] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. 2018. CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. In *Proc. of ACM MobiCom*. 305–320.

[65] Renjie Zhao, Timothy Woodford, Teng Wei, Qian Kun, and Xinyu Zhang. 2020. M-Cube: A Millimeter-Wave Massive MIMO Software Radio. In *Proc. of ACM MobiCom*. 15:1–14.

[66] Tianyue Zheng, Zhe Chen, Chao Cai, Jun Luo, and Xu Zhang. 2020. V2iFi: in-Vehicle Vital Sign Monitoring via Compact RF Sensing. In *Proc. of the 22nd ACM UbiComp*. 70:1–27.

[67] Tianyue Zheng, Zhe Chen, Jun Luo, Lin Ke, Chaoyang Zhao, and Yaowen Yang. 2021. SiWa: See into Walls via Deep UWB Radar. In *Proc. of the 27th ACM MobiCom*. 1–14.

[68] Xiaodong Zhuge, Alexander G Yarovoy, Timofey Savelyev, and Leo Ligthart. 2010. Modified Kirchhoff Migration for UWB MIMO Array-Based Radar Imaging. *IEEE Trans. Geosci Remote Sens* 48, 6 (2010), 2692–2703.