ADVANCED RF TECHNIQUES FOR WIRELESS SENSOR NETWORKS: THE SOFTWARE-DEFINED RADIO APPROACH

by

Sándor Szilvási

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
Electrical Engineering

May, 2014
Nashville, Tennessee

Approved:
Ákos Lédeczi
Xenofon Koutsoukos
Mitch Wilkes
Yuan Xue
Miklós Maróti

ABSTRACT

Traditional wireless sensor node designs follow a common architectural paradigm that connects a low-power integrated radio transceiver chip to a microcontroller. This approach facilitated research on communication protocols that focused on the media access control layer and above, but the closed architecture of radio chips and the limited performance of microcontrollers prevented experimentation with novel communication protocols that require substantial physical layer signal processing. Software-defined radios address these limitations through direct access to the baseband radio signals and an abundance of reconfigurable computing resources, but the power consumption of existing such platforms renders them inapplicable for low-power wireless sensor networking.

This dissertation addresses this disparity by presenting a low-power wireless sensor platform with software-defined radio capabilities. The modular platform is built on a system-on-a-programmable chip to provide sufficient reconfigurable computational resources for realizing complete physical layers, and uses flash technology to reduce power consumption and support duty cycling. The direct access the platform provides to the baseband radio signals enables novel protocols and applications, which is evaluated in two ways.

First, this is demonstrated by designing the physical layer of a spread-spectrum communication protocol. The protocol is optimized for data-gathering network traffic and leverages spectrum spreading both to enable an asynchronous multiple-access scheme and to increase the maximum hop-distance between the sensor nodes and the basestation. The performance of the communication protocol is evaluated through real-world experiments using the proposed wireless platform.

Second, a multi-carrier phase measurement method is developed for radio frequency node localization. Compared to existing interferometric approaches, this method offers more than four orders of magnitude measurement speedup and requires no deliberately introduced carrier frequency offset. The operation of the multi-carrier approach is validated using the new platform in various experiments. The analysis of the collected phase measurement data led to a novel approach for phase measurement-based distance estimation. This model is utilized to derive two maximum-likelihood distance estimators and their corresponding theoretical bounds in order to analyze and interpret the experimental results.

ACKNOWLEDGMENTS

In the completion of my research and this dissertation, I owe thanks to several people. First, I would like to thank my academic advisor, Professor Ákos Lédeczi, who gave me guidance and ensured a secure and stable research environment. I would also like to thank the rest of the committee members, especially Professor Xenofon Koutsoukos and Miklós Maróti, for their constructive criticism.

The members of the Institute for Software Integrated Systems (ISIS) research group contributed to my research each in their own unique way. Péter Horváth bootsrapped me on communications theory and Benjámin Babják never hesitated to provide assistance with the outdoor experiments. Tihamér Levendovszky, Róbert Kereskényi and János Máthé provided moral support and kept me grounded throughout the years. János Sallai was always eager to discuss new ideas and Daniel Balasubramanian would regularly volunteer to proofread my papers. However, I am most grateful for having had the opportunity to work with Péter Völgyesi. In short, I learned a great deal from him and he definitely shaped the way I look at science, engineering and life today.

Most importantly, I owe special thanks to my parents, brother, friends and lovely fiancée, Zsuzsanna Vörös. I might have not always been there for them when they needed me. Nonetheless, they were always patient with me, provided support and cheered me up whenever it was called for.

Partial support for this work has been provided by the National Science Foundation under Grants CNS-0964592 and CNS-1035627. Further thanks to the Nashville Metropolitan Parks and Recreation and Edwin Warner Park for allowing to use their space for the morning runs and the outdoor experiments.

TABLE OF CONTENTS

Page

Appendix

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

1 Motivation

Over the past decades, wireless sensor network (WSN) research has primarily relied on computer simulations to test new ideas and communication protocols. While such simulations are an essential and natural first step towards verifying innovative concepts, many radio propagation related simulations were criticized for making oversimplifying assumptions [1]. In response, several low-cost and low-power WSN platforms [2][3] were developed using commercial-off-the-shelf (COTS) radio chips attached to simple microcontrollers. The low power consumption of these platforms enabled the experimental validation of simulation results through long-term deployment in real-world scenarios [4][5]. However, *the closed architecture of the highly-integrated radio transceivers and the limited performance of the microcontrollers prevented experimentation with custom physical (PHY) layers.* Furthermore, while first generation WSN nodes employed a variety of radio chips with different PHY layers, upcoming node designs started to reduce such diversity by converging to a single standardized PHY solution. This, in turn, gradually rephrased the question of "what can be done in PHY and MAC layers?" to "what can be done with the specific COTS radio chip?" Therefore, a platform with reasonable computing performance and direct access to the PHY layer would offer tremendous opportunities to experiment with novel communication protocol stacks and various WSN services, such as node self-localization or time synchronization.

In contrast, software-defined radios (SDR) demonstrate exceptional flexibility when it comes to communication protocol prototyping and verification, as they implement the entire protocol stack in software, reconfigurable hardware or a combination of the two. The most common approach is to configure a field-programmable gate array (FPGA) based digital front-end for the high-speed signal processing tasks of the PHY layer, and connect it to a powerful multi-core desktop computer which implements the upper layers in software. In general, *SDR platforms provide the necessary flexibility and performance to define custom communication protocols, but their size and power consumption prohibit experimentation in realistic deployed WSN scenarios.* The lack of deployable SDR platforms, therefore, calls for finding a good balance between traditional SDR and WSN node architectures.

Motivated by the above observations, ***the goal of this dissertation is to apply the software-defined radio concept to wireless sensor networks.*** First, it proposes the design of a wireless node architecture that combines the low-power capability of wireless sensor nodes with a reasonable amount of computing power and the reconfigurable nature of traditional SDRs. Then, it demonstrates the potential of the SDR approach through the development and experimental evaluation of a communication protocol PHY layer and a multi-carrier phase measurement method for radio frequency node localization using the proposed platform, both of which would be infeasible with traditional WSN nodes.

2 Challenges

The architectural design of the SDR-capable low-power wireless sensor platform has to address the following challenges:

- **Power consumption.** Reduced power consumption and efficient power management techniques are key features that enable the ad-hoc and long-term deployment of WSNs. While in terms of power efficiency, the proposed reconfigurable platform is not expected to directly compete with highly integrated COTS radio transceivers, it should run complete WSN protocol stacks with power consumption much closer to that of WSN nodes than to desktop SDRs. Duty cycling and clock scaling are two essential power saving techniques that existing desktop SDRs lack due to the inherent limitations of SRAM technology-based FPGAs, but the proposed platform needs to offer. To evaluate the performance of such techniques, the platform should also provide the means to monitor and log the power consumption for detailed analysis.

- **Computational resources.** Existing SDR platforms offer design flexibility through direct access to the baseband radio signals along with an immense amount of reconfigurable computing resources to process them. In contrast, existing low-power sensor nodes lack such computational power by definition, and could not process the received radio signals even if they were accessible. Thus, although aimed to be low-power, the proposed platform is required to provide adequate amount of reconfigurable computational resources to define and experiment with novel WSN communication protocols that involve substantial PHY layer signal processing.

- **Development framework.** The development of FPGA applications in hardware description languages (HDL) is generally associated with a steep learning curve and long development times. However, several algorithmic and model-based high-level synthesis (HLS) tools exist that simplify HDL design entry and also allow for extensive model-based simulations. Therefore, to minimize the HDL implementation effort and improve the simulation fidelity of PHY layer components, the platform should be accompanied with a workflow that integrates model-based HLS tools and a set of basic infrastructure components. Furthermore, the framework should provide a means to stream the raw PHY layer signals to a computer for offline analysis.

The design of the sensor network communication protocol that potentially benefits from the use of custom PHY layer waveforms faces the following challenges:

- **Asymmetric radio link.** The vast majority of the proposed WSN communication protocols almost exclusively focuses on the design of energy efficient MAC layers with time-division multiple access (TDMA) or carrier-sense multiple access (CSMA) schemes [6], probably because the PHY layer of the COTS radio chips used on existing sensor platforms naturally supports these approaches but not others. However, there are alternative access schemes that offer a collision free medium at the cost of additional computational complexity, which can be distributed asymmetrically between a simple transmitter and a complex receiver. This asymmetry may be exploited in the vicinity of a resourceful basestation to enable sensor nodes to report sensor data asynchronously and simultaneously, while keeping their complexity low.

- **Rapid synchronization.** The communication in typical WSN applications is characterized by short packet lengths and low communication data rate, in the order of tens of bytes per second [7]. Furthermore, the network traffic is often bursty because neighboring sensor nodes tend to react to the same external events. Therefore, robust detection and demodulation of the quasi-simultaneously arriving messages impose tight synchronization requirements on the receiver.

In the context of existing radio interferometric phase measurement of RIPS [4] and SRIPS [8], the development of novel phase estimation approaches is challenged by:

- **Time synchronization.** The interferometric phase measurement of both RIPS [4] and SRIPS [8] is subject to a $\delta \cdot t_e \cdot 2\pi$ error, where δ is the interference frequency, a deliberately introduced frequency offset between the unmodulated carriers of the two transmitters, and t_e is the timing error between the two receivers. To keep this error term below a certain bound, both RIPS and SRIPS have to rely on μs accurate *external* time synchronization throughout the measurements. However, access to the baseband signal may enable the construction of alternative waveforms that allow to either incorporate the time synchronization into the phase measurement itself, or to compensate for its error.
- **Carrier wavelength.** The RIPS and SRIPS employed different COTS radio chips to perform the interferometric phase measurements in the 433 MHz and 2400 MHz frequency bands, respectively. The two experiments attained an order of magnitude different localization accuracy, however, no analytical expression is given on their dependence on the corresponding carrier wavelengths and phase estimation errors. A comprehensive analysis on the impact of these parameters might give useful insight into the preferable carrier frequency allocation strategies.

3 Contributions

The fundamental contribution of the this dissertation is a modular wireless research platform that addresses the low-power requirements of WSNs and the high-performance computational demand of SDRs simultaneously. The presented platform provides a means to approach the WSN research from the PHY layer perspective, consequently, to inspire ideas that are free from the architectural constraints imposed by the particular implementation of the few prevailing radio chips. To support this claim, two further contributions include the development and experimental evaluation of a long-hop asymmetric-link communication protocol and a multi-carrier radio frequency distance estimation method, both of which heavily rely on custom PHY layer waveforms. The specific contributions made in this dissertation are as follows:

Low-power Wireless Node Architecture. A novel low-power flash SoPC based SDR architecture has been designed for experimental evaluation of custom PHY and MAC schemes. The corresponding deployable MarmotE SDR wireless research platform has been implemented and fabricated, see Figure 9. A development framework has been provided that leverages existing model-based HLS tools to allow the PHY layer designer to focus on the signal processing aspect by reducing the HDL implementation effort. The resource utilization and power consumption of the MarmotE SDR platform have been evaluated through a GMSK modulation based communication protocol, where the flash FPGA fabric provided sufficient logic resources to host the complete PHY layer of complexity typical in WSNs. The sleep, transmit and receive mode power consumptions have been measured to be more than an order of magnitude below that of typical SDRs, however, sleep mode power consumption fell above the targeted range as the available SoPC model lacked an important power-saving feature common in flash FPGAs. Therefore, MarmotE SDR platform has already enabled multi-day long experimentation with full-custom communication protocols on a single battery charge, justifying the flash FPGA approach. Moreover, the successor flash SoPC device models incorporated the missing low-power mode that further extends the attainable duration of deployed experiments.

Long-Hop Asymmetric Link Communication Protocol. A direct-spread code division multiple access (DS-CDMA) scheme has been designed for data-gathering WSNs that are characterized by short packet lengths and bursty one-way traffic. The communication protocol leveraged the asymmetry in the PHY level waveform processing requirements to keep the transmitter complexity of the sensor nodes low, and shift the computational burden to a resourceful basestation residing at one-hop distance. Consequently, the transmitter has been implemented on the MarmotE SDR platform using HLS tools, while the receiver on a traditional SDR. Real-world experiments showed that the protocol enabled the MarmotE SDR nodes to asynchronously and simultaneously transmit their direct-spread messages to the basestation with arbitrarily reduced collision rates. The communication scheme also allowed to extend the maximum attainable hop-distance for a given transmit power level, therefore, to increase the number of nodes that are able to reach the basestation in a single hop. Neither experiment would have been practically feasible using traditional COTS radio chip based WSN nodes.

RF Node Self-Localization. A radio-frequency phase measurement method and a distance estimation algorithm have been developed for sensor node self-localization. First, the *relative* phase offset estimation problem of the radio interferometric approach [9] has been generalized to emphasize that the underlying problem is related to time difference of arrival (TDOA) estimation, then a multi-carrier phase measurement method has been proposed. The latter operates on the baseband complex signals directly and employs orthogonal frequency-division multiplexing (OFDM) waveforms, for which subcarrier allocation schemes have been proposed and the effects of timing and carrier frequency offsets analyzed. Compared to the single-carrier interferometric approach, it (i) requires no transmitter power calibration, (ii) allows to completely compensate the frequency offset between the transmitters, therefore, to relax the time synchronization requirements, and (iii) offers over four orders of magnitude measurement speedup at the expense of substantial baseband signal processing on both the transmitter and the receiver side. A MarmotE SDR implementation has been synthesized from a HLS model, which formed the basis of the outdoor experiments. For distance estimation, a complex sinusoid-based model has been introduced to inherently address the wrapping problem of the relative phase offset measurements. Based on direct analogies with frequency estimation, two maximum-likelihood distance estimators have been derived, along with their corresponding theoretical performance bounds. The Cramér-Rao lower bounds (CRLB) were then used as performance benchmarks throughout the analysis of the computer simulations and the experimental results, which then gave valuable insight into the roles of the subcarrier spacing, carrier frequency and effective bandwidth.

The organization of the dissertation follows the partitioning of the above contributions. Chapter II provides a survey on WSN and SDR architectures, followed by the description and evaluation of the MarmotE SDR platform. Chapter III reviews the prevailing WSN protocol stacks and the basics of spread-spectrum communications, then it discusses the design, MarmotE SDR implementation and experimental evaluation of the proposed DS-CDMA scheme. Chapter IV gives the background on WSN localization techniques, describes the phase and distance estimation methods and their experimental evaluation using the MarmotE SDR platform. Finally, Chapter V concludes the dissertation.

CHAPTER II

WIRELESS NODE ARCHITECTURES

1 Introduction

The advances of semiconductor process technology since the 70's has enabled continuous decrease in the size, cost and power consumption of silicon integrated circuits. Following Moore's law, each new process node generation allowed to fabricate approximately twice as many transistors on a unit die area, and to reduce their operating voltage gradually.

By the mid 90's, the availability of highly-integrated semiconductor devices reached a level that enabled the development of low-cost, network capable sensor devices. Ideas about ubiquitous networking and possible applications emerged, and the era of wireless sensor networks (WSN) set off. At the same time, the excessive amount of transistors allowed to replace fixed-function hardware solutions with flexibly reconfigurable architectures. In wireless communication technologies, this fostered experimentation with reprogrammable and reconfigurable full-vertical protocol stacks, including custom physical layers, which led to the recent boom of software-defined radios (SDR).

In this chapter, Sections 2 and 3 give an overview of the prevailing WSN and SDR architectures, respectively. Based on their suitability for experimentation with custom communication protocol stacks in typical WSN scenarios, Section 4 presents and evaluates the design of the proposed MarmotE SDR platform, a WSN platform with SDR capabilities. Conclusions on the MarmotE SDR are then drawn in Section 5.

2 Wireless Sensor Nodes

The technological advances in sensing and radio communication and the wide availability of disposably cheap and low-power integrated circuits enabled the emergence of the WSN paradigm: a collection of battery operated small sensing devices, capable of self-organizing a wireless network in order to collaboratively sense the physical environment.

The wireless communication capability provides the means for ad-hoc deployment of the sensor nodes in unattended areas with minimal or no infrastructure support. The ability to deploy large scale WSNs in practically any environment, in turn, offers tremendous potential to a wide variety of existing and novel sensing applications. Such application areas include health care [10], environmental and habitat monitoring [11][12][13], agriculture [14], industrial monitoring [15], military [16] and structural health monitoring [17]. In general, however, the specific WSN application and operating environment set different requirements for the sensor nodes in terms of power consumption, computing performance, sensing modality and communication parameters.

2.1 Sensor Node Architectures

Despite the varying WSN application requirements, the sensor nodes share several characteristics that translate to a common hardware architecture with functionally well isolated components. In the following discussion the sensor node hardware architecture is partitioned into the four subsystems shown in Figure 1: a control subsystem, a sensor subsystem, a communication subsystem and a power subsystem. In general, the control subsystem coordinates the operation of the sensor and communication subsystems to collect, process and forward data, while the power subsystem provides power for all the other subsystems.

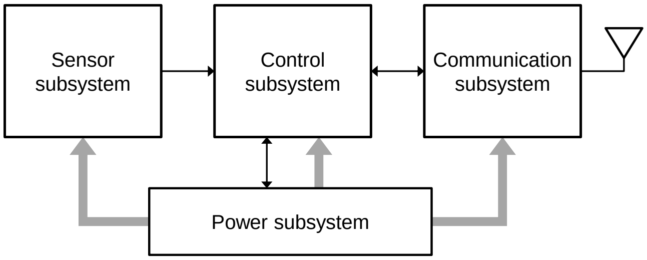
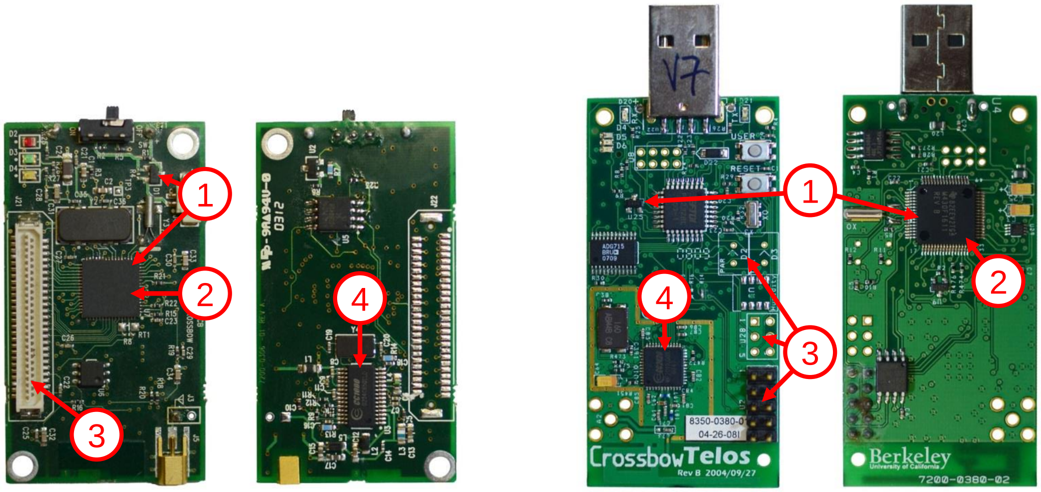


Figure 1: The architectural decomposition of typical wireless sensor nodes.

Highly integrated chips often incorporate several of the subsystem functions on a single die. Therefore, while the above decomposition of the node architecture simplifies its analysis, the actual boundaries of the physical components and that of the subsystems may be different. The subsystem fragmentation is illustrated for two WSN nodes in Figure 2.



(a) Mica2 top and bottom.

(b) TelosB top and bottom.

Figure 2: The power (1), control (2), sensor (3) and communication (4) subsystems of two common sensor node platforms, the Mica2 (a) and the TelosB (b).

2.1.1 Power subsystem

Typical WSN applications are deployed in harsh environments, where mains power is either completely unavailable [11][12][13] or too costly to be wired [15], therefore, the nodes have to rely on their own local energy sources. The lifetime requirement of the node, along with the energy scarcity in the operating environment poses a significant power supply design and management challenge, which is usually addressed using some form of energy storage, optionally backed up by an energy harvester. Thus, the main tasks of the power subsystem are to perform efficient energy storage and energy scavenging, furthermore, to provide regulated supply power for the rest of the sensor node.

Power management. Wireless sensor nodes follow two basic approaches to combat energy scarcity and consequently prolong network lifetime: they use low-power components and advanced power management techniques. The use of low-power building blocks for sensing, communication and control promotes longer battery life in general. As the components are selected at design time, their active and sleep mode power consumption are considered static parameters.

Power management techniques, on the other hand, dynamically switch between the various power states of the components to carry out an application task with minimum energy consumption. Efficient power management is generally achieved in WSNs through *duty cycling*, alternating between long sleep and short active mode periods. In turn, duty cycling based power management schemes are broadly categorized as either topology control or sleep-wake up protocols [6].

- **Topology control.** Topology control based duty cycling exploits node redundancy to prolong overall network lifetime by adaptively selecting a minimum required subset of nodes to achieve the application goals. The selection algorithm is driven by two main criteria, physical location and network connectivity. On one hand, a minimum required number of nodes is selected based on their physical location to provide adequate coverage for the application. On the other hand, nodes are also selected according to their location in the network topology to ensure network connection to all active nodes. The selected nodes become active participants of the application, while the rest of the nodes are turned off temporarily to save energy. In dense networks, topology control is reported to increase network lifetime by a factor of 1.5 to 3, compared to the case when all nodes are active participants [18].
- **Sleep-wake up protocols.** Sleep-wake up protocols achieve high power efficiency by utilizing a scheduling algorithm that alternates between sleep and active modes. In sleep mode, most of the sensor node subsystems are turned off or put into a low-power mode to conserve energy. Therefore, sleep mode power consumption is primarily determined by the static current draw of the device. In active mode, the node performs sensing, processing or transmission of data, consequently, the corresponding subsystems draw significantly higher currents. Duty cycle is defined as the ratio of the time spent in active mode compared to the node lifetime, and its efficiency is heavily affected by the wake-up time and the difference between sleep and active mode power consumption. As the latter two typically differ by orders of magnitude, a low duty cycle sleep-wake up schedule proportionally extends the lifetime of the node, and consequently that of the network. The actual sleep-wake up schedule and the achievable lowest duty cycle ratio is determined by the WSN application through the required sampling rate and the selected communication protocol.

Energy storage. The ideal energy storage is low-cost, low-volume, low-weight and is able to power a sensor node throughout the entire application lifetime. These attractive attributes would allow WSN nodes to become small, inexpensive and to operate autonomously. In search for such solutions, a wide variety of energy storage options has been proposed to provide the necessary power for the operation of WSN nodes, including single-charge and re-chargeable batteries, fuel cells [19] and supercapacitors [20].

The most favorable primary energy storages for large-scale, long-term deployed WSNs, where maintenance is an issue, became low-cost batteries with relatively high energy density and low self-discharge. The relatively inexpensive alkaline based batteries with high energy density and very low self-discharge, typically below 2% per year, fall into this category. Advocating this approach, several popular platforms [2][3] provide a battery pack typically for two AA size batteries. High-performance prototype sensor nodes [17] and WSNs that can afford maintenance [21], however, often opt for rechargeable solutions. The choice in such cases predominantly falls on more expensive rechargeable Li-ion batteries with high energy density and reasonably low, 5-8% per month, self-discharge rate. Further alternative energy storage options include fuel cells and supercapacitors. Fuel cells convert chemical energy into electrical energy. The promise of miniaturized fuel cells is several times the energy density of batteries with slightly less power, however, they have not been widely adopted in WSNs. Electric double-layer capacitors, or supercapacitors, on the other hand, are becoming widely used in WSNs as a secondary energy storage. The power density of supercapacitors is generally higher than that of batteries, but they discharge faster and their energy density is also lower [22]. These properties make supercapacitors suitable to store harvested energy for short-term, as an intermediate stage, and to use it to either charge a primary energy source battery or to power a sensor node directly.

Energy harvester. Energy harvesting has long been considered the ultimate power source for ubiquitously deployed sensor nodes. Long-established concepts of energy scavenging have grown into systems that transform wind, ambient light, heat, vibration or radio energy into electrical energy to power wireless electronic devices. Table 1 summarizes the estimated performance of the various approaches based on the survey presented in [23].

Energy source	**Performance**	**Comment**
Ambient airflow	2 mW/cm^3	MEMS device [24]
Ambient light	18 mW/cm^2	direct sunlight [25]
	100 μW/cm^2	office environment
Vibration (mechanical)	1–2 μW/cm^3	human body motion
	200–800 μW/cm^3	machine attached [26]
Thermoelectric	60 μW/cm^2	
Ambient radio	< 1 μW/cm^2	

Table 1: Comparison of various energy harvesting methods.

Devices of considerably different sizes have been proposed for utilizing ambient airflow. MEMS airflow microturbines measure 0.5 cm^3 in volume and achieve 1 mW output power [24]. Wind generators, such as the AmbiMax energy harvesting platform [27], can generate significantly higher output power, however, their several orders of magnitude larger physical size is prohibitive for most WSN applications. Solar power is another attractive environmental energy source utilized in energy harvester prototypes, such as the mentioned AmbiMax, as well as Prometheus [25] and Everlast [20]. These harvesters generally use a supercapacitor to buffer energy and charge the primary energy storage, which is usually a rechargeable battery. The performance of common solar cells varies between 100 mW/cm^2 and 100 μW/cm^2, when lit by direct bright sun or illuminated by office light, respectively [23]. Vibrational energy harvesters scavenge energy usually by exploiting the oscillation of a proof mass tuned to the dominant frequency of the environment. The achievable performance highly depends on the operating environment, peaking at several hundreds of μW/cm^3 in industrial environments when mounted on machines vibrating at few kHz frequencies [26].

Thermoelectric generators produce electrical power from the temperature difference between objects or environments. When a dense array of thermoelectric elements, with good thermoelectric coefficients of around a few hundred μV/K, is connected in series, the result is a compact device that can produce up to 60 μW/cm^2 with only 5°C of temperature difference. Energy scavenging from ambient radio power or deliberately broadcast radio energy is also an attractive approach to power sensor networks [28]. However, due to the rapid decrease of the transmitted signals electrical field and regulatory restrictions, the available harvestable power is severely limited. The practical operating range of such radio energy harvesters is limited to a few meters, making it ideal for passive radio frequency identification (RFID) tags, but not for powering large-scale WSNs.

2.1.2 Controller subsystem

The controller subsystem is primarily responsible to collect data from the sensor subsystem and supervise the other subsystems of the node. The collected data is usually processed by a digital processing unit, then either stored locally or passed to the communication subsystem for transmission. WSN control subsystems most commonly utilize COTS microcontrollers, due to their reasonable *energy efficiency*, *computing performance* and *design flexibility*. However, during the past decades, several alternative architectures were proposed for applications that called for different trade-offs between these three parameters, as shown in Figure 3. The following sections present an in-depth comparison of these architectures .

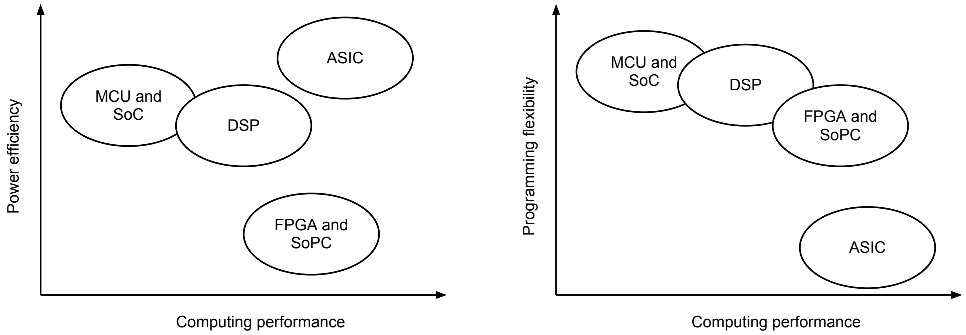


Figure 3: Qualitative comparison of WSN node controller subsystems in terms of computing performance, power efficiency and programming flexibility.

MCU. The microcontroller unit (MCU) is a highly integrated silicon device comprising a processor, a memory, a power management unit and several I/O peripherals, where the single-chip integration of these components has several advantages. First, it allows for compact sensor node designs that require minimal board space and cost. Several widely used MCUs are available in packages with footprint as small as 5x5 mm and require only a few external components, mainly decoupling capacitors [29]. Second, it offers efficient power management schemes that fit well the low duty cycle operation of wireless sensor nodes. Current consumption of ultra low-power MCUs in sleep mode is in the μA range, while in active mode, they can deliver a computation-energy efficiency of 500 μA/MHz, with the total consumption kept usually below 1 mA, see Table 2. Finally, the processor and memory of MCUs provide sufficient computing performance, storage and programming flexibility to host the processing algorithm of most WSN applications. The MCU processor architectures typically vary by vendor and model, with bus widths spanning from 8 bits to 32 bits. Memory capacity of low-power MCUs are constrained in general, but vary significantly, with typical program memory sizes in the range from a few to tens of kilobytes, and RAM capacities in the range of a few hundred to a few thousand bytes.

MCU model	ROM	RAM	Wake-up time	Sleep current	Active current
Atmel ATmega128L	128 kbytes	4 kbytes	200 μs	8 μA	2000 μA/MHz
TI MSP430F1611	48 kbytes	10 kbytes	6 μs	2 μA	500 μA/MHz

Table 2: Resource and power consumption comparison of two MCUs widely used in WSN nodes.

In summary, the compact design, low-power features, low-cost and ease of programmability made the MCU the most popular choice for sensor node processing unit.

SoC. The System-on-Chip (SoC) approach represents an increased level of chip integration as it incorporates both a complete MCU subsystem and a dedicated radio-frequency transceiver on a single die. SoCs are available in as small as 7x7 mm packages, comparable in size to regular MCUs, thus they further reduce cost and the required board area. SoCs share the benefits of efficient power management features and deliver the same computation-energy efficiency as traditional MCU-based controller subsystems. However, the integration between control and radio subsystems is tighter as the latter is usually mapped into the processor memory space, which in turn allows for faster response times both in sleep and active modes. The most common SoCs host MCU architectures widely used in earlier MCU-based control subsystems. Hence, they provide the same computing performance, storage capacity and programming flexibility. That is, low-power SoC architectures also come with bus widths from 8 bits to 32 bits, program memory capacities of a few kB to a few hundred kB and RAM sizes of a few hundred to a few thousand bytes. Similarly to MCU architectures, commodity SoCs utilize radio transceiver architectures that have already been used on WSN nodes as a separate chip. This gives an option to migrate existing standalone-MCU-based WSN designs to a SoC without the need to change the radio related specifications.

DSP. Digital signal processors (DSP) offer a low-power, embedded solution for WSN applications that require computationally intensive processing of the sensor data. Running digital signal processing algorithms on the sensor nodes themselves comes at the price of increased power consumption. However, in data-intensive applications, this is often the only feasible option as streaming of raw data is prohibitively expensive in terms of communication bandwidth and the power consumption associated with it.

Embedded DSPs are available in small 10x10 mm footprint packages [30], which is comparable to MCUs and SoCs. However, DSPs generally incorporate less peripherals and require more external components, which results in a considerably higher cost and larger board area. The power management unit of embedded DSPs supports similar low-power modes as standalone and SoC embedded MCUs, which allows for duty cycle operation. Although, the sleep mode current consumption of DSPs is typically a few hundred μA, two orders of magnitude higher than that of MCUs. Furthermore, despite DSPs having a computation-power efficiency similar to MCUs, see Table 3, their clocks run at tens of MHz frequency, yielding an active mode current draw of tens of mA. The processor and memory architecture of DSPs is tailored to processing streams of data, by supporting efficient single-instruction-multiple-data (SIMD) operations and direct memory access (DMA) transfers, providing large data buffers and executing most instructions in a single clock cycle. The data memory capacities in DSPs are typically at least 64 kB, and the data bus widths are at least 16 bits.

DSP model	**Wake-up time**	**Sleep current**	**Active current**
ADSP-2188	25 μs	100μA	26 mA @ 80 MHz
Marvell PXA271	few ms	390μA	31 mA @ 13 MHz

Table 3: List of embedded DSPs used in WSNs.

WSN applications that require high-speed and complex processing of data streams, such as audio or video signals, often utilize embedded DSPs for the controller subsystem. Even though DSP chips have power efficiency per performance ratios comparable to those of MCUs and SoCs, they are clocked at significantly higher speeds resulting in higher active mode current. More importantly, however, they have higher static power consumption in sleep mode which also shortens the sensor node lifetime.

FPGA. Field programmable gate arrays (FPGA) offer a reasonable alternative to DSPs for complex on-node processing of sensor data. Though the architecture of FPGAs is radically different from that of DSPs, using an FPGA in the controller subsystem means a similar trade-off of computing performance for power consumption. On the other hand, the common design entry method for FPGAs is through the use of hardware description languages (HDL), which generally requires different skills and longer development than programming DSPs. However, with high-level synthesis (HLS) tools, the HDL code may be generated from a high-level algorithmic or model-based descriptions, and the development effort of FPGA-based control subsystems becomes comparable to the programming of MCUs, SoCs or DSPs.

Low-density FPGA devices are available in small packages, occupying only 8x8 mm board area [31][32], but they require the similar amount of external components as DSPs, such as external clock source and decoupling capacitors. Moreover, the wake-up time and static power consumption of FPGAs varies highly depending on the fabrication technology and the device size, as suggested by the list of devices in Table 4. Therefore, a brief description of each FPGA type follows evaluating their applicability to WSN nodes.

- **SRAM FPGA.** Traditional SRAM FPGA-based sensor nodes have been used in several WSN applications [17][16] despite some unfavorable power management properties in duty cycle operation. SRAM technology FPGAs are associated with higher static (quiescent) current draw than MCUs, which leads to increased sleep mode power consumption unless the power

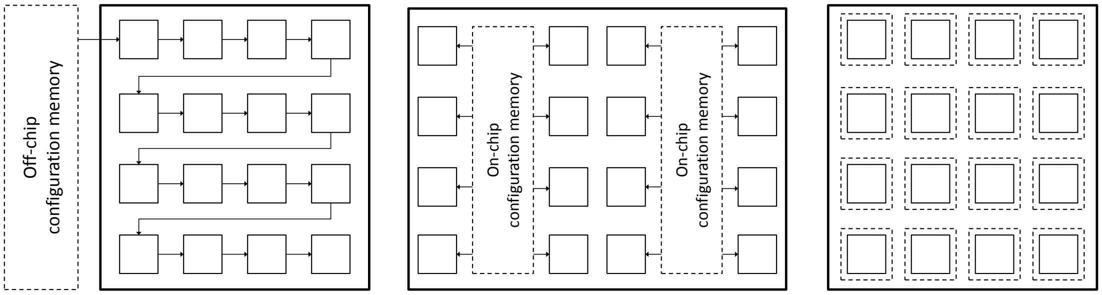
FPGA model	Fabric CMOS technology	Wake-up time	Sleep current
Xilinx Spartan-3 XC3S1000	SRAM	>10 ms	37 mA
Xilinx Spartan-6 LX4	SRAM	>10 ms	6 mA
Altera Cyclone V 5CEA2	SRAM	>10 ms	52 mA
Lattice LatticeXP2-5	Hybrid	few ms	14 mA
Microsemi Igloo AGL600	Flash	few μs	1.7 μA

Table 4: Wake-up time and sleep current comparison of FPGAs used in existing wireless sensor nodes. Note that the sleep currents correspond to 1.2 V core voltage, and that the size of the FPGAs is considerably different.

rails are turned off. Turning off the device to eliminate leakage current losses is impractical, however, as the active FPGA configuration is stored in volatile SRAM cells that lose their content upon device power-down. Thus, the FPGA needs to be reconfigured on every wake-up event, which has several disadvantages. First, the configuration bitstream is typically stored in a serial accessible off-chip non-volatile memory, see Figure 4(a). Therefore, reading and loading its content takes time in the order of tens of ms, as illustrated in Figure 5(a), with the actual reconfiguration time largely determined by the size of the SRAM FPGA device. Second, reconfiguration takes energy and introduces significant in-rush current [33]. Third, the application state is also lost with the content of the SRAM memory cells. Clearly, all of these reduce the effectiveness of duty cycling.

The architecture of the SRAM FPGA comprises a uniform array of configurable logic, interconnect, embedded memory and hardware multiplier or DSP block that allows the creation of massively parallelized data paths. The processing of high sample-rate, multi-channel audio or video data usually maps more naturally to such configurable data paths than to an essentially sequential DSP architecture, therefore, it allows the control subsystem to run on reduced frequency clocks.

Despite the several disadvantages of SRAM FPGAs in low duty cycle operation, often required in long-term deployed WSNs, if the application calls for high-performance, multi-channel signal processing at the sensor node, such FPGAs are either the only viable control subsystem option, or are reasonable alternative to DSPs.


Off-chip configuration memory
On-chip configuration memory
On-chip configuration memory

(a) SRAM FPGA.

(b) Hybrid FPGA.

(c) Flash FPGA.

Figure 4: Programming architecture comparison of SRAM (a), hybrid (b) and flash (c) FPGAs.

- **Hybrid FPGA.** Hybrid FPGAs store the active configuration for the interconnects and the function of programmable logic elements in SRAM cells, similarly to SRAM FPGAs. The main difference is, however, that hybrid FPGAs address the issue of long start-up time associated with the configuration process of SRAM FPGAs. Attributing the slow speed of configuration mainly to the serial access of an external memory device, hybrid FPGAs incorporate on-chip non-volatile memory to store device configuration [34]. Furthermore, hybrid FPGAs provide a means to transfer the configuration data from the on-chip memory to the configuration SRAM cell in a parallel manner, see Figure 4(b), as opposed to traditional SRAM FPGAs that load the configuration serially. The parallel loading of the configuration decreases the startup time by a factor of approximately 50, resulting in a wake-up time of a few ms. Although this is still three orders of magnitude slower than that of the MCUs and SoCs, it is a significant improvement for FPGAs to operate in duty cycle mode.

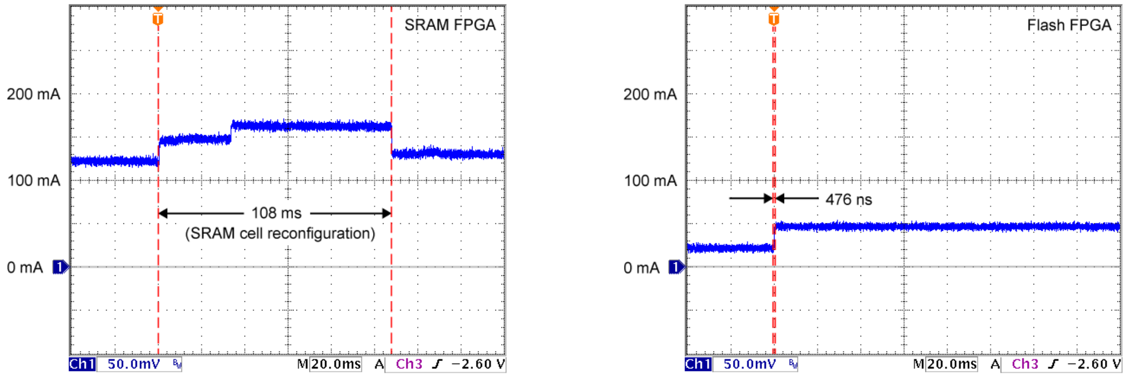
  Once configured, the general architecture of hybrid FPGAs is basically identical to SRAM FPGAs. Therefore, assuming same process node technology, they draw similar amount of static and dynamic current in active mode. Also, they provide the same density of programmable logic and embedded hardware resources to define massively parallel distributed signal processing datapaths in the controller subsystem.

  In summary, hybrid FPGAs offer the same computational performance as SRAM FPGAs with reasonable wake-up time improvement that is crucial for low-power duty cycling operation. Despite their respectable ms order startup-time, however, hybrid FPGAs are underrepresented in the existing arsenal of wireless sensor networks.

- **Flash FPGA.** Flash FPGAs take a completely different approach compared to SRAM and hybrid FPGAs, as they load the active configuration directly into non-volatile flash memory cells that directly define the interconnects and the function of programmable logic elements, see Figure 4(c). FPGAs based on flash technology are claimed to achieve lower static power consumption [33], thus allow for low-power sleep modes and efficient clock-scaling. As flash memory cells retain their configuration in power off mode, no reconfiguration is needed when turning on the device. Furthermore, certain flash FPGA devices offer advanced power saving features for sleep mode, which reduce the power consumption to as low as a few μW while retaining block RAM and register data [32]. Entering and exiting sleep mode takes less than 1 μs, which makes flash FPGAs the most favorable FPGA choice for duty cycle operated sensor nodes. The measurements presented in Figure 5 give a quantitative comparison of the startup process of SRAM and non-volatile flash FPGA devices. Observe that the 108 ms wake-up time associated with the SRAM FPGA, measured from requesting wake-up until the PLL is locked, is reduced to 476 ns using flash FPGA fabric due to the absence of the reconfiguration phase.

  Apart from the storage of the active configuration, the architecture of flash FPGAs is essentially the same as SRAM or hybrid FPGAs, a dense programmable interconnect network surrounding large arrays of configurable logic blocks and various embedded resources. Though theoretically flash and SRAM FPGAs could incorporate the same embedded blocks and have comparable logic densities, current flash FPGAs lack embedded multipliers and multiply-and-accumulate (MAC) units. Furthermore, current state-of-art flash FPGAs are manufactured at 130 nm process node, whereas modern SRAM FPGAs are currently fabricated at 28 nm, which inherently reduces the achievable logic density per unit area by orders of magnitude.

In summary, flash FPGAs have unique abilities, such as μA order sleep current and sub-microsecond wake-up time, that are crucial for efficient duty cycle sensor node operation and are not possessed by SRAM or hybrid FPGAs. Although the programmable logic resources in flash FPGAs are less abundant than in SRAM devices, they still prove to be sufficient for many WSN applications [17].



(a) SRAM FPGA.

(b) Flash FPGA.

Figure 5: Wake-up time comparison of SRAM (a) and flash (b) FPGAs, measured with development kits that utilize similar size SRAM and flash FPGAs.

SoPC. System on a programmable chip (SoPC) refers to a compact, highly integrated silicon device that embeds a complete MCU subsystem with a hard CPU core, several peripheral units, a reasonable sized array of general purpose programmable logic, and even complete analog or mixed-signal subsystems. Several SoPC devices are available under different brand names, such as Programmable System on Chip (PSoC) or customizable System-on-Chip (cSoC), each referring to similar SoPCs architectures. Current SoPCs are compared in Table 5, in terms of CPU core type, programmable logic technology and size. Package footprint sizes of smaller SoPC devices start at 7x7 mm, rendering them a strong competitor of MCUs and FPGAs with the added features of programmable logic and low-power hard CPU, respectively.

SoPCs offer a wide palette of devices with different trade-offs between low-powerness and computational resources. On one end of the spectrum, the ultra low-power devices are similar to MCUs and SoCs supplemented with a limited amount of general purpose programmable logic. Such devices are the Cypress Semiconductor PSoC 3 and PSoC 5 [35] that host microcontroller subsystems (MSS) based on a 8051 and an ARM Cortex-M3 CPU, respectively, along with a moderate number of universal digital blocks (UDB). The advanced power saving features of PSoC devices, along with the mA range active and μA range sleep current draw, promote their low-duty-cycle, battery based operation as a sensor node controller subsystem. While the programmable UDBs provide additional flexibility to peripheral management and aid simple signal processing tasks, the computational performance of PSoC devices is mainly provided by the CPU.

SoPC model	**CPU core**	**Fabric CMOS technology**	**Logic density**	**Sleep current**	**Active current**
Cypress Semiconductor PSoC 3	8051	SRAM (130 nm)	Small	1 μA	1.2 mA
Cypress Semiconductor PSoC 5	Cortex-M3	SRAM (130 nm)	Small	2 μA	6 mA
Microsemi SmartFusion	Cortex-M3	Flash (130 nm)	Medium	15 mA†	21 mA
Xilinx Zynq-7000	Cortex-A9	SRAM (28 nm)	Medium	-	>200 mA
Altera Cyclone V	Cortex-A9	SRAM (28 nm)	Medium	-	>150 mA

Table 5: Current consumption comparison of SoPC devices with hard microcontroller subsystem. †SmartFusion model A2F500 with sleep mode defined by running at 32 kHz.

Similarly, the current flash FPGA-based SoPC devices are built around an ARM Cortex-M3 MSS, incorporate similar analog peripherals and an array of uniform programmable logic. The feature set, performance and power consumption of the SmartFusion MSS is comparable to that of PSoC devices, however, the programmable array is based on a reasonably complex flash FPGA fabric [36], rather than on a small PLD. The embedded flash FPGA fabric shares the same characteristics as stand-alone flash FPGAs [32], that is, the capability to perform highly parallel signal processing tasks while still supporting duty-cycle operation via low static power consumption and short wake up times.

On the other end of the spectrum are the SoPC devices with high-performance hard microprocessors surrounded by a rich set of peripherals and a sea of programmable logic. The FPGA fabric of such SoPC devices, including the Altera Cyclone V [37] and Xilinx Zynq-7000 [38], is SRAM based and built with 28 nm CMOS technology. Even the smallest models of these devices offer excessive amount of reconfigurable resources for typical WSN applications and feature powerful embedded DSP blocks, consequently, they outperform the MCU, SoC and FPGA devices discussed thus far. However, the price for this performance is the high static and dynamic power consumption ranging in the order of hundreds of mA and the several ms wake up time associated with re-configuration.

Similarly to SoC devices, SoPCs represent the current state-of-art semiconductor technology, but with an additional array of uniform programmable logic. The amount of the programmable logic varies significantly by the various device models, as well as the power-awareness, which ranges between that of simple stand-alone MCUs and mid-sized FPGAs. While the power consumption of most SRAM FPGA-based SoPCs is still prohibitively large to be used in WSNs, several lower-end SoPC devices offer an thought-provoking trade-off between processing performance and current draw.

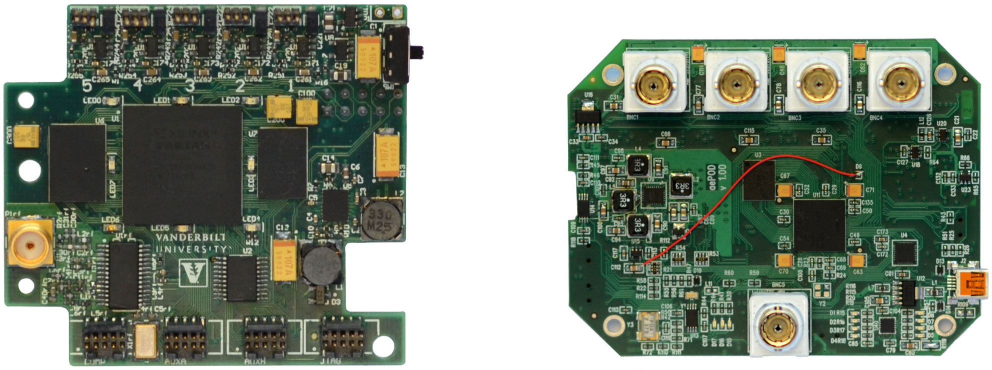
ASIC. An application specific integrated circuit (ASIC) implementation of a control subsystem provides the most compact and power-efficient solution [39]. The achievable processing performance per unit power is inherently higher than with FPGAs, as the die size for the same *fixed* functionality is significantly smaller. However, the non-returning engineering cost, development and manufacturing time associated with ASIC production are usually unacceptable for prototyping in WSNs.

Therefore, an ASIC approach is generally unsuitable for prototyping and experimenting with sensor node architectures, whereas it has unrivaled advantages when it comes to high-volume production of sensor nodes with fixed functionalities.

2.1.3 Sensing subsystem

The primary task of WSNs is to gather information on certain aspects of the environment such as temperature, wind, humidity, air pollution, acceleration, pressure, detect fires, earthquakes or landslides, or determine the location of acoustic sources. Regardless of what physical phenomena the WSN nodes monitor, their sensing subsystems share the common characteristics.

The typical WSN node sensing subsystem consists of three main components, a sensor, a signal conditioner and an analog-to-digital converter (ADC). The sensor is essentially a transducer that transforms heat, pressure, light or some type of energy into electrical energy. The output of the transducer is usually weak and distorted by noise, therefore, a signal conditioning stage follows to amplify and filter it, then to match its level to the ADC dynamic range. The required ADC resolution and sampling rate are also application dependent. Simple MCU-integrated ADCs, offered by several platforms [2][3][40][41], suffice in many WSN applications, such as for temperature and humidity monitoring, or fire detection. However, certain acoustic source localization problems, such as shooter localization[42] and acoustic emission detection [17], call for increased precision or sampling rate that is generally achieved by separate high-resolution, high-speed ADCs. Figures 6(a) and (b) show two such multi-channel WSN platforms with ADCs operating at MSPS order sampling rates.



(a) SRAM FPGA-base sensor node for acoustic localization and tracking.

(b) Flash FPGA-based sensor node for acoustic emission monitoring.

Figure 6: Two FPGA-based WSN platforms with high sampling rate ADCs and multi-channel capability.

Power requirements imposed on the WSN node inherently apply to the sensing subsystem too. Thus, accuracy is often traded off for low-power consumption in active mode operation (measurement) and for various power saving features. Such feature is the ability to put the sensing subsystem into sleep mode or to keep it actively sampling and have it wake up the controller subsystem upon an external event.

2.1.4 Communication subsystem

The most common task of the communication subsystem is to transmit the collected and preprocessed sensor data to a basestation, where post-processing, data fusion and evaluation takes place. For wireless transmission, WSN nodes typically utilize low-cost COTS packet radios that operate in the 433 MHz, 915 MHz or 2400 MHz industrial, scientific and medical (ISM) radio bands. These radios are commonly designed with transmit powers in the -20 to 0 dBm and receiver sensitivity in the -95 to -100 dBm ranges, which limits their communication range to a few tens of meters in practical WSN scenarios. Their PHY layer employs amplitude-shift keying (ASK), frequency-shift keying (FSK) or quadrature phase-shift keying (QPSK) modulation schemes that occupy few hundred kHz to few MHz wide frequency bands to achieve the targeted 50 to 250 kbps data rates.

The communication subsystem is associated with three main distinct power modes, transmit, receive and sleep. In a single-hop network, the communication subsystem typically enters transmit mode to send measured sensor data to the basestation and receive mode to receive configuration parameters, while in a multi-hop network nodes also forward data to each other. In either scenario, listening to the radio channel consumes energy comparable to transmitting on it, as illustrated in Table 6. Therefore, only sleep mode offers reasonable power savings that can be exploited for efficient duty cycling operation.

The relatively low transmit and receive mode power consumption of the transceiver chips listed in Table 6 is largely due to their highly integrated ASIC implementation. SoC radio solutions, shown in Table 7, take silicon integration to the next level by incorporating the radio core circuitry and the MCU on a single die. The radio cores of both the standalone ASIC and SoC radio chips contain analog and digital circuits that implement the analog radio front-end, the complete PHY layer and parts of the MAC layer of a given radio stack.

Radio model	**Modulation**	**Transmit current**	**Receive current**	**Sleep current**
CC1000	FSK	10.4 mA	7.4 mA	0.2 μA
CC1101	ASK, FSK	17.2 mA	15 mA	0.2 μA
CC2420	O-QPSK	17.4 mA	18.8 mA	0.02 μA
AT86RF230	O-QPSK	14 mA	15.5 mA	0.2 μA

Table 6: Comparison of COTS radio chips for WSN communication subsystems. Transmit mode assumes 0 dBm transmit power, sleep mode assumes turned-off voltage regulator.

SoC radio model	**Radio core**	**Transmit current**	**Receive current**	**Sleep current**
CC2430Fxxxx	CC1101	17 mA	15 mA	0.5 μA
ATmega128RFA1	AT86RF231	10 mA	15.5 mA	0.2 μA
nRF24AP2	nRF24L01+	15 mA	17 mA	2 μA

Table 7: Current consumption comparison of SoC radio chips for WSN communication subsystems. Transmit mode assumes 0 dBm transmit power, sleep mode assumes turned-off voltage regulator.

The COTS radio chip approach clearly results in a power-efficient ASIC implementation of the lower stack layers and usually provides packet level interface convenient to access from the controller subsystem, which is preferred for the design of most WSN applications. However, the rigid ASIC implementation severely limits experimentation with the lower layers.

2.2 Existing Platforms

Several dozen different research and commercial WSN platforms have been proposed and developed in the past decades. Among the first WSN nodes were the WeC and the Rene motes, both featuring on-board sensors and an 8-bit MCU with a few kilobytes of program memory. These were followed by the pioneering Mica and Telos series. The first Mica mote hosted an Atmel ATmega103 MCU with an RFM TR1000 ASK radio transceiver [40]. As successors, the Mica2, shown in Figure 2(a), and the MicaZ [2] were both built around an 8-bit ATmega128L MCU. The Mica2 platform used a simple CC1000 FSK transceiver, and became one of the most popular platforms in the WSN community for several years. The next generation MicaZ mote followed the trend of using the 2.4 GHz band and utilized the 802.15.4 compliant CC2420 O-QPSK radio chip. Similarly, the TelosB [3], shown in Figure 2(b), relied on the CC2420 transceiver, and backed it up with an even lower-power MSP430 MCU. Subsequent WSN platforms also converged to the 2.4 GHz radio band. The IRIS and BTnode [43] platforms were both based on the ATmega128 MCU too, however, while the former was equipped with an AT86RF230 radio chip, the latter utilized both a Bluetooth and a CC1000 transceiver.

WSN platform	**Microcontroller**	**Radio**
Mica	ATmega103/ATmega128	TR1000
Mica2	ATmega128L	CC1000
MicaZ	ATmega128L	CC2420
TelosB	TI MSP430F1611	CC2420
IRIS	ATmega1281	AT86RF230
BTnodes	ATmega128L	CC1000 and Zeevo ZV4002
XSM	ATmega128L	CC1000
iMote 1	Zeevo TC2001P	Bluetooth
iMote 2	Marvell PXA271	CC2420

Table 8: Comparison of existing WSN platforms.

The platforms listed in Table 8 exhibit the characteristics of traditional WSN nodes with a battery operated MCU and radio transceiver along with an extension header to connect sensors. First generation nodes utilized a diverse set of radio chips with different physical layers, showing the signs of early experimentation. However, with the introduction of the 802.15.4 protocol and the associated integrated COTS radio chips, platform designs started to converge to a common set of lower stack layers. Further technology advances fostered the integration of the MCU and the radio on a single die both in research prototypes [44][39] and commercial radio chips.

Most WSN hardware platforms gained popularity through commercialization by companies like Crossbow or Moteiv (Sentilla), and today some offer complete modular hardware and software solutions for a wide range of WSN applications [45]. Nonetheless, several of the platforms [17][46] departed from the traditional architecture due to highly specialized application demands.

In the past decades several node architectures have been proposed to address the tight energy, communication and computational requirements of the various WSN applications. While most architectures promoted a general platform approach with interchangeable sensor subsystem, some responded to unique application requirements with specialized control subsystem demonstrating increased signal processing capabilities at the expense of power efficiency. Such control subsystems mainly rely on SRAM-based FPGAs that lack efficient power saving modes. Therefore, the question is still open whether there exist better architectural compromises between power consumption and flexibility for experimentation using different technologies.

3 Software-Defined Radios

A software-defined radio (SDR) is a wireless communication system that implements a substantial portion of the lower communication protocol stack layers in software, as opposed to the fixed function hardware components used in traditional radios. The software implementation of the lower layers provides flexibility to prototype and experiment with novel radio communication protocols at the expense of a significant amount of computational resources.

Such flexibility in radio communication is desirable in several scenarios, including military, commercial and research. Two major initiatives for SDR were the Speakeasy [47] and the Joint Tactical Radio System [48] military projects. The primary goal of Speakeasy was to develop a communication system that could emulate several existing radios over a wide range of frequencies by exploiting the latest advancements in DSP technologies. Similarly, the JTRS project envisioned a modular and scalable radio system that is capable to interoperate with existing legacy military radio systems. The SDR concept also finds application in commercial wireless communication systems, as it allows to prototype and evaluate new standards for mobile cellular phones, and possibly to remote-update the SDR basestation configuration and software to implement such upcoming standards without actual hardware replacements. Moreover, wireless communication researchers and developers also benefit from the SDR approach, as it provides means to validate concepts and obtain performance metrics of the lower protocol stack layers with reasonable engineering effort.

Software implementation of wireless communication protocol layers and the design of the underlying platform pose several challenges. The most stringent requirements can be summarized according to [49] as follows: high system throughput, intensive computation and hard real-time constraints (low latency). As several platforms exemplify in Section 3.1.3, achieving high system throughput and providing adequate amount of computational resources are generally not limiting factors for traditional SDRs. However, adhering to the tight latency requirements posed by several wireless protocols is usually challenging for all-software SDR approaches.

3.1 Software Radio Architectures

Most SDR systems exhibit the common architecture depicted in Figure 7, which involves an *analog radio front-end*, a *digital front-end* and a *digital processing unit*. The former generally employs tunable and precision analog radio-frequency components, while the latter two are commonly realized based on a CMOS SRAM technology based FPGA and a desktop class computer, respectively.

Such architectural approach directly benefits from the advances of semiconductor process technology through high programmable gate count FPGAs, suited for massively parallel baseband signal processing, and modern multi-core CPUs that host the upper layers of the radio stack.

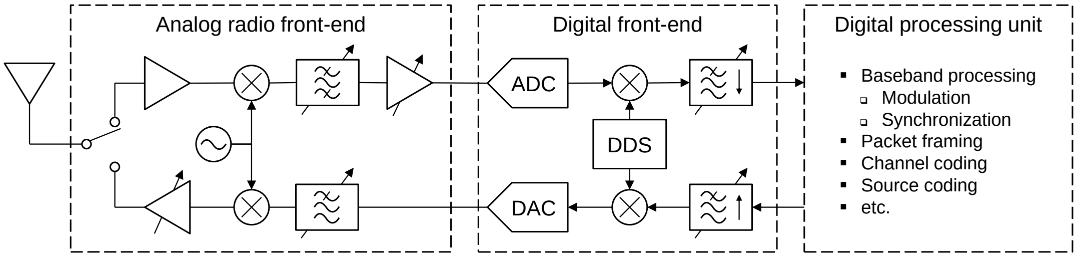


Figure 7: The typical architecture of SDRs.

The following sections describe the roles of the three main architectural components in detail, categorize the existing SDRs and examine their propriety for deployed WSN scenarios.

3.1.1 Analog Radio Front-End

The purpose of the analog radio front-end is to convert the radio frequency (RF) signal impinging on the antenna to a lower intermediate frequency (IF), where the received signal is simpler to process. Analogously, its role during transmission is to mix the modulated IF signal to the RF band prior to radiating it through the antenna.

The architecture of a typical single-antenna analog radio front-end is illustrated in Figure 7. To enable half-duplex communication, the antenna is connected to a transmit–receive antenna switch that selects the active path. In the receive path, the switch is closely followed by a fixed-gain, or coarsely adjustable, low-noise amplifier (LNA) to increase the received signal power without significantly corrupting it with noise. The amplified RF waveform is then multiplied by the output of the tunable local oscillator to down-mix the received signal to the IF band. Note that the actual IF frequency value is a design parameter, where zero is also a valid choice, that is, zero-IF receivers mix directly to the baseband. The signal is then driven through a relatively wide IF passband filter to suppress strong out-of-band noise and to perform anti-aliasing. Finally, the filtered IF-band analog waveform is amplified by an optional fine-adjustable variable-gain amplifier (VGA) prior to passing it to the digital front-end for digitization.

The transmit path of the analog radio front-end in Figure 7 contains elements with functions complementary to the receive path, albeit in reverse processing order. The modulated analog IF signal first passes through an IF bandpass or lowpass filter, which acts as an anti-imaging filter and smooths the DAC output of the digital front-end. The filtered IF-band waveform is then mixed to the RF band determined by the local oscillator frequency and its power level is boosted by a variable gain power amplifier (PA). With the antenna switch set to transmit mode, the amplified signal current then drives the antenna.

In practice, the analog and digital front-ends are designed either onto separate daughter and mother-boards or onto a single printed circuit board. The former approach, followed by the USRP [50] or KUAR [51], employs interchangeable analog radio front-ends, each targeting a specific *relatively* narrow band of the RF spectrum. However, with the proliferation of integrated, precision and flexibly tunable RF solutions, recent analog front-end daughterboards and most single-board designs enable tuning over a particularly wide RF band that commonly covers the range from 400 MHz to 4 GHz.

3.1.2 Digital Front-End

According to Figure 7, the digital front-end is responsible for the conversion between the analog and digital domains, as well as for the transition between the IF and baseband.

On the receiver path, the ADC digitizes the IF signal at a relatively high, 10–100 MHz, sampling rate. Next, the digitized IF-centered signal is multiplied by an IF-frequency complex sinusoid, generated by the direct digital synthesizer (DDS). The low-pass filter (LPF) then passes the resulting baseband signal and rejects the components at twice the IF frequency. Moreover, it cleans up the spectrum to prevent aliasing during decimation. Finally, the decimator, generally integrated into the LPF, reduces the sampling rate to relax the computational requirements for the following baseband signal processing blocks. The DDS-based mixer, the LPF and the decimator together are commonly referred to as digital down converter (DDC).

Correspondingly, the transmit path in Figure 7 starts with the interpolation of the relatively low sampling rate baseband signal. The combination of the interpolator with the LPF ensures that the spectrum images are suppressed in the upsampled signal. The baseband waveform is then mixed to the IF band by multiplication with the DDS generated IF-frequency sinusoid. Together, the interpolator, the LPF and the DDS-based mixer are called the digital up converter (DUC). Eventually, a DAC converts the high sample rate DUC output into an analog IF signal before passing it to the analog front-end.

Although the partitioning shown in Figure 7 is common, see USRP N210 [50] or KUAR [51], the actual boundaries between the hardware components vary in practice. Third-party radio front-end developers prefer to place the ADCs and DACs adjacent to the analog components for compatibility with a wide range of FPGA and DSP development boards through standardized connectors. On most front-end boards, the DDC and DUC functions are most commonly implemented in an FPGA or a separate ASIC due to their high sampling frequency requirement. In such case, the DDS is usually based on a CORDIC core [52], and the decimation and interpolation on either only a FIR filter, or its combination with a CIC filter. However, the DDC and DUC operations may also get shifted into digital processing unit, in which case the digital front-end does not appear as a separate hardware entity. Furthermore, zero-IF radio front-ends convert directly from RF to baseband and and back. Therefore, the down-mixed analog signal is already centered at zero frequency and the DDC and DUC blocks become unnecessary.

3.1.3 Digital Processing Unit

The fundamental task of the digital processing unit is to provide the necessary computing resources to implement the software-defined functions of the radio. Regardless of the communication protocol stack height, such functions include the PHY layer and its corresponding signal processing. Baseband processing, in turn, generally incorporates both transmitter and receiver related tasks, such as modulation, demodulation and receiver synchronization, that make the PHY layer the computationally most intensive portion of the stack. Note that the corresponding operations can always be implemented in software. However, the high bandwidth and tight latency requirements of recent communications protocols, along with the inherent data-level parallelism in the streamed, low-level signal processing, make certain digital processing architectures preferable. For this reason, the term *software-defined* is used in a broader sense in the following, referring to both software implementation and hardware configuration of some generic logic blocks.

Numerous architectures have been proposed for efficient implementation of SDR class radio systems as demonstrated by the various research and commercial SDR platforms. The various platforms can be categorized based on the architecture of the underlying digital processing unit, as summarized in Table 9. The rest of this section discusses details of the general purpose processor, multiprocessor and reconfigurable hardware based approaches.

Architecture	**Platform**
General purpose processor	USRP/GNU Radio [50][53], Vanu [54], Sora [49]
Multiprocessor	Sora [49], SODA [55], SODA-II [56], PicoArray [57], HyperX [58], SandBlaster [59]
Reconfigurable hardware	XiSystem [60], WARP [61], KUAR [51], AirBlue[62], commercial platforms [63][50][64][65]

Table 9: Comparison of existing SDR platforms based on their underlying digital processing unit.

General purpose processor. The classical architecture for general SDR hardware is best represented by the general purpose processor (GPP) based USRP/GNU Radio [50][53], Vanu [54] and Sora [49] platforms. All three platforms use a separate analog and digital radio front-end hardware, but implement the entire wireless physical layer in software and run it on the GPP of a desktop computer. The RF front-end mainly consists the analog RF circuitry, the converters to interface between the analog and digital domains, and further high-speed interface logic to stream the baseband samples between the RF board and a desktop computer. The stream is then processed by the GPP mainly using portable high-level software components.

The main advantages of the GPP approach are that GPP-based commodity desktop computers are relatively inexpensive and most users are already familiar with the architecture and the programming environment. Several major drawbacks exist, however. The interface between the RF front-end and the desktop computer creates a bottleneck, which makes throughput and latency requirements of several wideband protocols difficult to meet. Furthermore, the architecture of the GPP and its cache system lacks support for highly parallel DSP applications. The Sora platform attempts to overcome this issues by utilizing several Gbps buses, and multi-core GPPs along with cache optimization and core dedication techniques. However, according to [66], Sora fails to meet the requirements of certain widely used wideband wireless protocols.

Multiprocessor. The signal processing of the wireless physical layer is essentially a dataflow, where the dataflow tasks can be efficiently run in parallel on multiprocessor architectures. The Sora platform takes one step in this direction by dedicating processor cores of a multi-core GPP to a specific signal processing task. Since several dataflow blocks can further benefit from parallel or vector computations, specialized multiprocessor architectures with optimized data path routing and single instruction multiple data (SIMD) instruction set processor cores are more suited for energy efficient baseband radio signal processing. The SODA [55] and its successor SODA-II [56] platform are two such architectures that use four and two processing cores, respectively, to meet the throughput and timing requirements of various wideband radio protocols (e.g. W-CDMA and 802.11a) in a power efficient manner. Both SODA platforms rely on static multi-core scheduling and were simulated using wireless protocols implemented in all-software. Finer grained multiprocessor architectures are the HyperX [58] and picoArray [57] platforms, which utilize hundreds of statically scheduled processor cores along with sophisticated bus interconnects.

The Sandblaster [59] platform exploits parallelism inherent in wireless physical layer processing through multithreading. Its microarchitecture hosts several SIMD vector processing units with each having its own dedicated data memory. Multiple copies of the data are available in the dedicated memories, which allows the processors to execute all hardware threads simultaneously.

Reconfigurable hardware. The picoArray platform can also be considered a coarse grained reconfigurable hardware due to its programmable inter-processor bus network, where the static schedule of the time division multiplexed bus switches is determined at compile time. A finer grained approach is to use field programmable gate arrays (FPGA) to define the routing network, accelerate processor cores or realize the entire communication protocol stack directly. The FPGA-based SDR platforms are often argued to be difficult to program and to lack debugging support. However, with the evolution of algorithmic and model-based high-level synthesis tools, the required engineering effort becomes comparable to that of software programming.

The various SDR platforms leverage the flexibility offered by FPGAs at different degrees in different ways. The GPP-based Sora [49] and USRP [50] platforms both utilize an FPGA, but its role is essentially to interface the digital baseband and control signals of the radio front-end with the GPP and, therefore, it implements only a negligible portion of the physical layer. However, the WARP [61], KUAR [51] and other research platforms [62][66] implement the vast majority of the wireless protocol stack in the FPGA fabric preprocessor, and most commercial SDR platforms [63][65][64] follow the same approach.

Another use of FPGA resources is to configure them as hardware accelerators for processor cores. One example is the XiSystem [60] System-on-Programmable-Chip (SoPC), where the XiRisc processor uses dedicated FPGA fabric, the Pipelined Configurable Gate Array (PiCoGa), as a customizable pipelined execution path. The XiSystem architecture features another dedicated FPGA fabric, the eFPGA, specialized to implement various digital interfaces. Some existing FPGA-based SDR platforms [61][51] already contain hard processor cores and, therefore, allow for SoPC approaches similar to that of XiSystem. However, recently introduced commercial SoPC families [38] are expected to gain further ground in both high-performance and low-power SDR platforms.

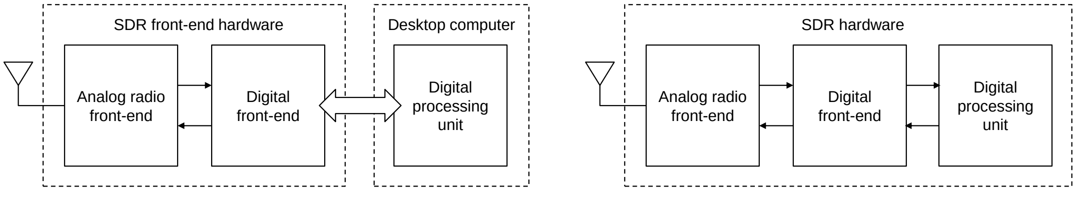
3.2 Deployability

Considering the use of SDR systems in typical WSN scenarios immediately calls for the analysis of the ease of node deployment. In this context, deployability is directly related to the portability of the node, its self-contained, battery-based operation capability in remote environments. Therefore, portability is primarily determined by the power consumption and physical size of the platform, which generally contradicts the classical SDR philosophy, where the exact same parameters are traded-off for protocol design flexibility. Based on the above definition of deployability, the following sections categorize the existing platforms as either a desktop- or an embedded SDR.

Desktop SDRs. Desktop SDRs represent the high-end platforms that heavily rely on large FPGAs and powerful desktop computers. The abundance of computational resources promotes experimentation with the physical layer of complex wireless protocols that utilize wideband signals. The USRP N210, the Sora [49] and several FPGA development board based platforms [66] demonstrated their performance by implementing high subcarrier-count and on-the-fly configurable OFDM-based protocols. However, the price for the high computing performance provided flexibility is limited in portability. The above platforms are radio front-ends connected to a desktop class computer through a high-speed bus interface, see Figure 8(a). This scheme prescribes a complete computer for each SDR node, which clearly limits the scale of practical deployments. Furthermore, the vast majority of the front-end devices require mains power. Therefore, even with the use of laptop computers, the total power consumption of desktop SDRs generally prohibits their lifetime in battery based operation, hence their use in outdoor scenarios.

Embedded SDRs. Embedded SDR platforms, on the other hand, are stand-alone devices optimized for compact size and low power operation. The stand-alone operation is achieved by processing the wireless physical and upper layers on an embedded processing unit that resides on the same board as the radio front-end, see in Figure 8(b), rather than on a desktop class computer. Several processor architectures have been proposed for embedded SDRs, including the picoArray [57], SODA [55], SODA-II [56], Sandblaster [59] and XiRisc [60], that provide significant amount of computational resources in a power efficient manner. However, these conceptual processor architectures almost never get mass-fabricated, which severely limits their availability for deployed real-world experimentation.

Other self-contained SDR platforms rely on reconfigurable hardware in the form of either FPGAs or SoPCs. The USRP Embedded series [67] use both an FPGA and an embedded CPU to implement the full vertical wireless stack. Similarly, the KUAR radio [51] relies on a SoPC and an embedded processor. While both of these platforms are designed to be portable, their high power consumption severely limits their battery based operation in deployed scenarios.



(a) Desktop SDR.

(b) Embedded SDR.

Figure 8: Comparison of the desktop SDR and embedded SDR architectures.

Recent advances in semiconductor and wireless technology allowed the SDR concept to become a challenging practical architectural design problem, rather than a purely theoretical one. Several general and specialized hardware architectures with varying levels of computational capability and power efficiency have been proposed to flexibly implement full vertical wireless communication stacks. The most promising approaches rely on reconfigurable hardware in the form of FPGAs and SoPCs. Such reconfigurable platforms are suitable to meet the computing, throughput and latency requirements of SDRs, however, their relatively large size and high power consumption severely limit their large scale outdoor deployment to experiment with radio stack specialized for wireless sensor networks.

4 MarmotE SDR Platform

Traditional low-power wireless sensor node designs follow a common architectural recipe that connects a low-range integrated radio transceiver chip to a microcontroller. This approach facilitated research on communication protocols that focused on the MAC sublayer and above, but the closed architecture of radio chips and the limited performance of microcontrollers prevented experimentation with novel communication protocols that involve substantial PHY layer signal processing. SDRs address these limitations through direct access to the baseband radio signals and a vast amount of reconfigurable computing resources, but the power consumption of existing SDR platforms renders them inapplicable for low-power networking in the WSN domain.

Driven by the above disparity and our experience with FPGA and SoPC-based sensor nodes [16][42][68][69][17], this section proposes a flash SoPC-based flexible WSN node architecture that attempts to find a balance between low-power operation and processing capabilities. Therefore, the goal of the proposed MarmotE platform is to enable experimentation with both power saving techniques, such as duty-cycling and energy harvesting, and sensor node applications that require high-speed, parallel processing of multi-channel sensor outputs. Observe that in this aspect the analog *radio* front-end is essentially regarded as a specific type of analog *sensor* front-end, where the received baseband signal is treated as the sensor output.

The MarmotE platform follows a modular three-layer approach, where the analog sensor front-end, the mixed-signal flash FPGA-based processing unit and the power management unit are separated into three different modules as shown in Figure 9. The stacked architecture allows to seamlessly replace the top-layer sensor front-end, and the bottom-layer power management modules, while keeping the same mixed-signal processing module intact. Consequently, the MarmotE can be used for structural health monitoring [17] with acoustic emission sensors, or for acoustic source localization [16][42] with a multi-microphone analog front-end. However, the following discussions assume the use of an *analog radio front-end* as the top-layer and emphasize the SDR aspect of the platform by referring to this particular node stack configuration as MarmotE SDR.

4.1 Hardware Architecture

The following MarmotE SDR configuration includes a 2.4 GHz radio front-end[1], a flash FPGA SoPC-based mixed-signal processing module and a rechargeable battery-based power management module. The photo and block diagram of these modules are presented in Figure 9.

[1] The 2.4 GHz radio front-end was designed by Benjamin Babjak, and the analog interface components of the mixed-signal module were selected with his assistance.

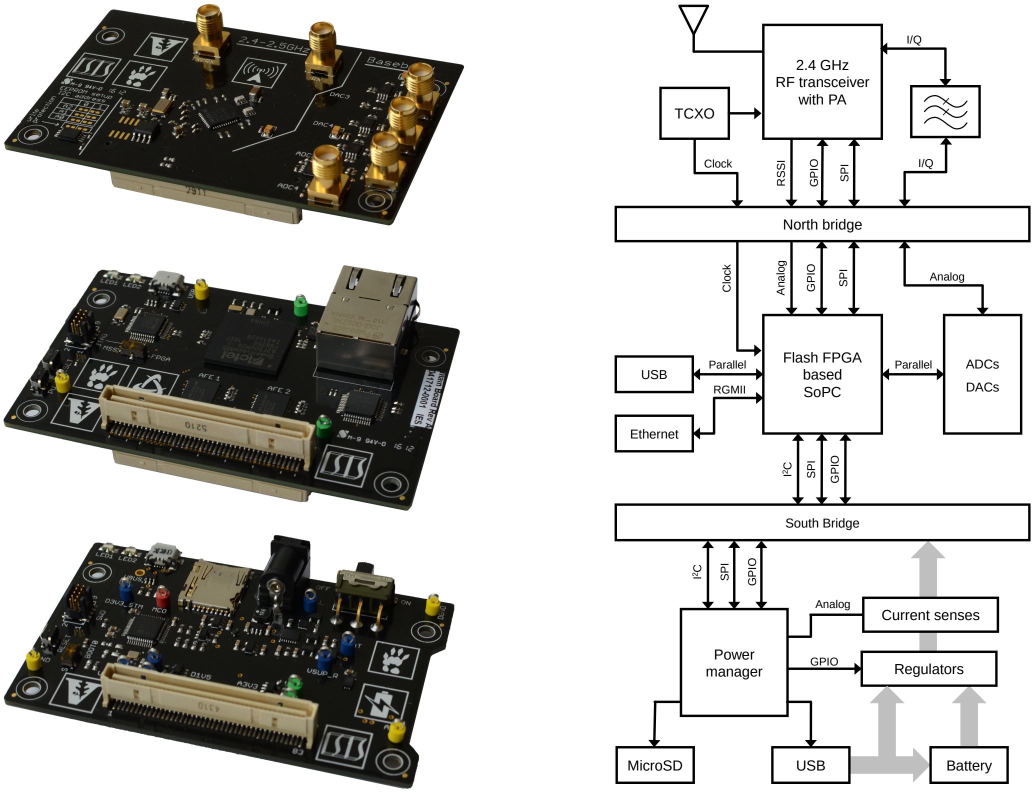


Figure 9: Photo (left) and block diagram (right) of the modular MarmotE SDR platform comprising a 2.4 GHz radio front-end (top), a flash SoPC-based mixed-signal processing (middle) and a battery operated power management (bottom) module.

4.1.1 2.4 GHz radio front-end

The analog radio front-end module at the top-layer is designed to operate in the 2.4 to 2.5 GHz ISM frequency band and to interface with the middle-layer mixed-signal module through analog baseband I/Q signals, both for transmission and reception. Providing such direct access to the baseband complex signals allows the definition of full-custom waveforms and, consequently, to customize the lower PHY layers of 802.11 and 802.15.4 protocols, as well as to enable experimentation with various types of channel access methods, such as TDMA, FDMA and CDMA, in addition to different modulation techniques.

The radio front-end is built around the integrated Maxim MAX2830 RF transceiver, power amplifier, transmit–receive and antenna diversity switch and, thus, supports both single and dual-antenna setups. The MAX2830 was primarily chosen, for it was one of the few models that makes both the receiver and transmit baseband I/Q signals accessible. The single die integration of most RF functions saves board space and reduces the overall power consumption. However, the analog components of the MAX2830 are designed for wider bandwidth, higher linearity and dynamic range requirements than the commodity low-cost RF chips, consequently, they draw significantly more current. The transceiver chip also incorporates a voltage controlled oscillator and a fast settling, 20 Hz step adjustable RF synthesizer. While the original goal of the precise digital tuning capability is to allow use of low-cost crystals, we chose to use the integrated crystal oscillator as a buffer and to drive it by a precise 2.5 PPM, low-power temperature compensated crystal oscillator (TCXO). The stable and accurate TCXO along with the fine adjustable synthesizer are expected to give sufficiently precise control over the local oscillator frequency for applications, where formerly this was found to be an issue [68].

The direct conversion, zero-intermediate frequency RF-to-baseband receiver and baseband-to-RF transmitter paths are also part of the RF chip, along with the programmable 7.5–18 MHz low-pass baseband filters. The analog receive and transmit baseband signals are complex I/Q pairs digitized and processed by the ADCs/DACs and the FPGA, respectively, on the middle-layer mixed-signal processing module. Thus, while the current 2.4 GHz radio front-end hosts a single RF transceiver, future MarmotE SDR front-ends may utilize a second transceiver for multiple-input and multiple-output (MIMO) and multi-band RF applications.

4.1.2 Mixed-signal processing module

The middle-layer of the MarmotE SDR platform is a mixed-signal processing module, which is the main building block of every MarmotE SDR application. In the current setup, this module controls the top-layer radio front-end and provides computational resources for a complete vertical network stack, including PHY layer baseband signal processing.

The basis of the mixed-signal processing module is a flash FPGA-based SoPC and two external analog front-ends (AFE) that make the module capable of simultaneously processing two sets of analog baseband I/Q signal pairs. Each set of the analog baseband I/Q receive and transmit pairs is connected to the 10-bit ADCs and DACs of a Maxim MAX19706 type low-power AFE, respectively. While interfacing with two sets of baseband signals renders the mixed-signal module suitable for MIMO application development, the current 2.4 GHz radio front-end contains only a single transceiver and provides no support for MIMO operation. The AFE sample clock is driven by the SoPC, and it is also used to synchronize the ADC and DAC sample transfers through a 10-bit parallel double data rate (DDR) digital bus at sampling rates up to 22 MSPS. Parallel DDR interface was preferred to high-speed serial interfaces as it matches the SmartFusion FPGA fabric characteristics and allows the FPGA to transfer the I/Q samples in a single clock cycle and, therefore, to operate the entire fabric in a single, low-frequency clock domain.

The Microsemi A2F500 SmartFusion SoPC comprises flash FPGA fabric and a 32-bit microcontroller subsystem interconnected with an ARM Advanced Microcontroller Bus Architecture (AMBA) bus. The SmartFusion chip was primarily selected for its FPGA fabric, built with 130 nm flash-based CMOS process, with sufficient configurable logic elements to implement reasonable PHY layers along with the upper layers of the stack. As discussed in Section 2.1.2, flash FPGAs retain their configuration during power-off, wake up orders of magnitude faster and draw lower static current than SRAM FPGAs [70], which makes them a preferable choice for low-power WSN applications. The actual power consumption of the MarmotE SDR platform is evaluated in Section 4.3.

The SmartFusion SoPC also contains a microcontroller subsystem (MSS) comprising an ARM Cortex-M3-based 32-bit microprocessor with a rich set of communication peripherals and a high-speed, low-latency AMBA bus to interface with the FPGA fabric. This tight connection between the processor and the FPGA fabric provides flexibility to move the border between hardware and software in a network stack implementation. Furthermore, it allows one to accelerate application software components with FPGA cores, which was practically infeasible on former platforms [71].

Besides the two AFEs and the SmartFusion SoPC, the mixed-signal processing module is equipped with Ethernet and USB controllers. The Ethernet connection is primarily for instrumentation, in-application reprogramming and debugging in large-scale WSN deployments, where USB topologies scale poorly. The USB interface, on the other hand, offers a data path to stream raw or partially processed 16-bit I/Q baseband samples to a desktop computer at rates up to 5 MSPS.

4.1.3 Power management module

The bottom-layer interchangeable module is a battery based power management system designed to regulate and monitor the power rails of the MarmotE SDR platform. Its main purpose is to power the entire MarmotE SDR stack, and to measure and log current draw along with battery status.

The power management module has three possible sources of power, a 5 V wall adapter, a USB connector and a Li-Ion battery. The former two are used both to power the voltage regulators and to charge the battery using a CC/CV circuitry with a charging profile tailored to the attached 6000 mAh Li-Ion battery. A BUCK step-down regulator controls the 1.5 V rail, while a low-dropout regulator is used on the 3.3 V rail, primarily supplying the core and the I/O blocks of the SmartFusion SoPC on the mixed-signal module, respectively. Both power rails are available for the upper layer modules, along with the unregulated external 3.6–5 V rail if further voltage levels are needed. The power management module also monitors the current of both the analog and digital 1.5 V and 3.3 V power rails via current sense circuitry, and counts the battery charge using a battery gauge. Both the analog current sense outputs and the digital battery gauge output are connected to a low-power microcontroller that measures and optionally logs these data through USB or to a memory card.

4.2 Development Framework

The hybrid WSN–SDR architecture of the MarmotE SDR platform benefits from both worlds during the development and experimental evaluation of custom communication protocols. The former, however, also imposes challenges compared to traditional software WSN workflows due to the heavy reliance on the FPGA fabric of the SoPC, consequently, the associated HDL implementation effort. This section explains the possible operating modes of MarmotE SDR, the provided support and the steps required to implement communication protocols on the platform.

4.2.1 Operating Modes

Desktop SDR. Although designed for self-contained operation in the first place, the MarmotE SDR is equipped with the necessary connectivity features to operate in the desktop SDR configuration shown in Figure 8(a). The USB interface provides a convenient means to stream raw or preprocessed complex baseband signal to a desktop computer at rates limited by the high-speed USB standard. Direct processing of the USB stream results in the traditional SDR setup, where the MarmotE SDR acts as the analog radio front-end, the flash FPGA fabric on the mixed-signal module optionally implements the DDC and DUC functions of the digital front-end, but the communication protocol is essentially realized in all-software, using GNU Radio, Simulink or similar tools.

Of greater benefit is, however, that real-world baseband data can be recorded for off-line analysis through the very same radio front-end that would be used in a deployed scenario. Therefore, the recorded raw stream can be used to aid the development of software and HDL signal processing blocks, as well as to fine tune the analog radio front-end parameters.

Embedded SDR. The MarmotE SDR is primarily designed for stand-alone operation, as shown in Figure 8(b), to foster experimentation in deployed WSN scenarios. In this mode, the entire communication protocol stack is implemented in the SoPC, with the functions divided between the flash FPGA fabric and the microcontroller. The Ethernet and USB interfaces can then be used for instrumentation, to adjust protocol parameters and collect performance metrics, such as the number of successfully received packets or the power consumption registered by the power management module. However, note that the wired Ethernet network is not intended to be used to stream the raw baseband signals in general.

4.2.2 Framework Components

The MarmotE SDR framework incorporates the SoPC vendor provided development toolchain and a collection of our platform specific HDL and software infrastructure components, see Figure 10. The latter are created with convenient interfaces to hide low-level details and reduce the implementation effort of the protocol specific components. Therefore, the design of a custom communication protocol requires one to partition its functions between the FPGA and the microcontroller, and to develop the corresponding HDL and software code.

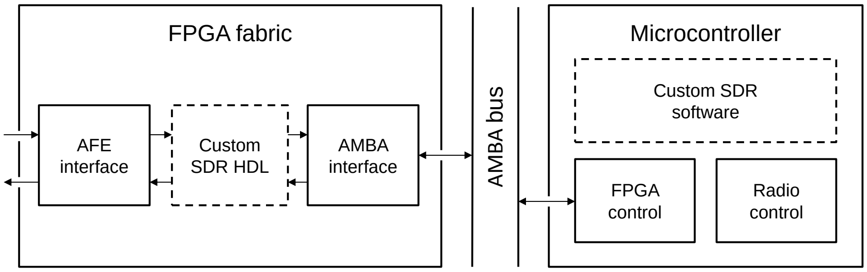


Figure 10: HDL (FPGA) and software (microcontroller) components of the MarmotE SDR development framework.

Infrastructure components. The MarmotE SDR provided infrastructure components are the basic building blocks of every SDR application. They are specifically tailored to interface the FPGA fabric with the high-speed external AFEs and the SoPC microcontroller, and the microcontroller with both the FPGA and the analog radio front-end.

The HDL infrastructure components include the *AFE interface*, the *AMBA interface* and a placeholder for the protocol-specific custom HDL code, as shown in Figure 10. The AFE interface utilizes the DDR capable I/O blocks of the FPGA to communicate with the AFE at sample rates up to 22 MSPS, and strobes the 10-bit I/Q samples to and from the custom HDL component in receive and transmit modes, respectively. The latter component serves as a wrapper for the FPGA partition of the communication protocol, which usually embodies most of the PHY layer signal processing. On the other end, the AMBA interface employs the AMBA bus to pass data between the protocol-specific custom HDL and software components. Since the FIFO and register requirements of the AMBA interface generally vary by the actual protocol implementation and the chosen hardware–software boundary, templates are provided for its customization.

The two software infrastructure components abstract the control of the analog radio front-end and the interface with the FPGA fabric, respectively. The *radio control* component provides functions to initialize the MAX2830 transceiver chip, switch between receive and transmit modes, tune the carrier frequency, and adjust the analog gains and baseband filter bandwidths. The *FPGA control* is the software counterpart of the AMBA interface. It defines the memory map for the FIFOs and registers, and assigns interrupt handlers to their corresponding events. Observe that such scheme allows for low-latency, high-throughput data transfers over the AMBA bus, especially with the use of DMA. The software partition of the communication protocol may then rely on the functions and register definitions of the infrastructure components to implement the upper layers of the protocol stack.

Protocol specific components. The custom HDL and embedded software components together define the actual communication protocol, which one has to partition and implement in accord with the infrastructure component interfaces. As a given processing task can generally be performed by both the FPGA fabric and the microcontroller, the partitioning between the two is primarily determined by the associated computational capabilities and implementation effort.

Since the FPGA fabric is more suitable to process the baseband signals, it usually implements the entire PHY and parts of the MAC layer. However, the development of FPGA applications in VHDL and Verilog languages is generally associated with steep learning curve and long development time, which are often addressed by using algorithmic or model-based high-level synthesis (HLS) tools. The HLS tools simplify HDL entry and the available high-level Simulink stimulus and analysis blocks allow for fast round-trip validations and extensive model-based simulations, see Figure 47 for a demonstrative simulation setup. Therefore, to minimize the HDL implementation effort and improve the simulation fidelity of PHY layer components, the MarmotE SDR development workflow promotes the use of model-based HLS tool through conveniently interfaced infrastructure components that operate in a single-clock domain. Such model-based HLS approach is used for the MarmotE SDR implementation of the communication protocols in Sections 4.3 and 4, and the localization algorithm in Section 5.

The microcontroller performance is limited compared to that of the FPGA fabric, however, the implementation of higher-level functions usually takes less effort in software. Observe that from this point, the design flow is essentially the same as for traditional WSN nodes but with a custom made, memory-mapped radio peripheral. Therefore, the software partition, the custom SDR software component in Figure 7, is expected to implement the non-timing-critical portion of the MAC and the higher layers in a typical WSN communication protocol.

4.3 Evaluation

A prototype communication protocol PHY layer was developed in order to evaluate the MarmotE SDR platform. The proof-of-concept protocol is meant to serve as a case study in order to illustrate the MarmotE SDR design flow and to provide valuable insight into the resource usage of the infrastructure components. Furthermore, it is used as a reference to compare the MarmotE SDR power consumption with an existing integrated radio chip and a desktop SDR.

4.3.1 Reference communication protocol

A Gaussian minimum-shift keying (GMSK) modulation was selected for the reference PHY layer partially because it forms the basis of the Bluetooth and GSM communication standards, but mainly for it is a slightly more complex variation of the binary frequency-shift keying (FSK) scheme employed by the representative CC1000 [72] commodity WSN radio chip. The PHY layer was implemented in the flash FPGA fabric with software-driven control functions running on the microcontroller.

Similarly to binary FSK, a GMSK transmitter switches between two alternative frequencies according to the data symbols, however, with a controlled transition. In the reference design, each binary symbol generates an impulse that drives the Gaussian pulse shaping filter, as depicted in Figure 11. The filter oversamples the data symbols by a factor of 8 and continuously updates the phase accumulator, which designates the phase of the generated baseband complex sinusoid. Observe that the sinusoid phase is continuous due to the integrating effect of the accumulator, hence GMSK is also called a continuous phase modulation scheme.

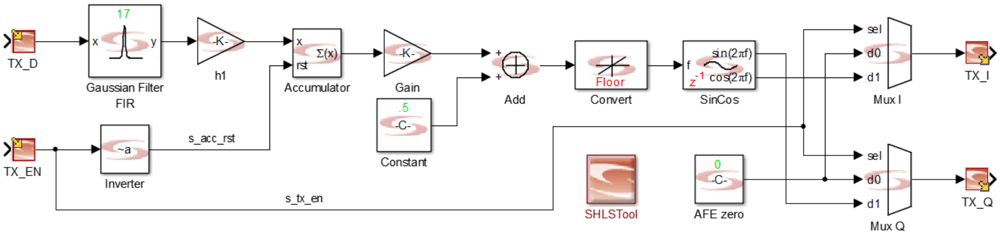
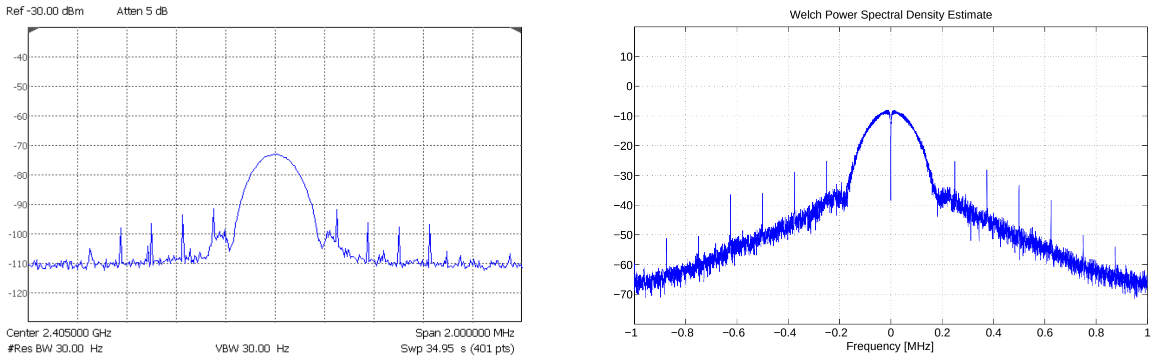


Figure 11: Simulink model of the HDL synthesizable GMSK modulator.

The most fundamental parameter of GMSK modulation is the bandwidth–time (BT) product, which is generally used to parametrize the width of the Gaussian pulse. The pulse width, in turn, can be used to control the sideband power in the GMSK signal spectrum at the expense of increased inter signal interference (ISI). The reference design employs BT = 0.5. The corresponding spectrum of the MarmotE SDR transmitted 250 kbit/s data-rate GMSK signal is shown in Figures 12(a) and (b), as seen by a spectrum analyzer and another MarmotE SDR, respectively.

The GMSK receiver design comprises a demodulator and a synchronizer, shown in Figures 13 and 48 in Appendix A. The demodulator receives the signal with the spectrum shown in Figure 12(b), and smooths it with a lowpass filter. Then it multiplies the actual sample with the conjugate of the previous one to approximate the instantaneous frequency, and estimates the potential symbol values with a limiter–discriminator scheme. The synchronizer implicitly downsamples the incoming signal by consecutively loading the binary samples into 8 different shift registers, and simultaneously correlating the content of each with a synchronization bit pattern. Upon match, symbol and frame synchronism is declared and the output of the corresponding register is regarded as the sequence of the received binary data symbols.



(a) Spectrum analyzer

(b) MarmotE SDR receiver

Figure 12: Power spectral density estimate of the MarmotE SDR transmitted GMSK signal (250 kbps, BT = 0.5), measured by a spectrum analyzer (a) and a MarmotE SDR receiver (b).

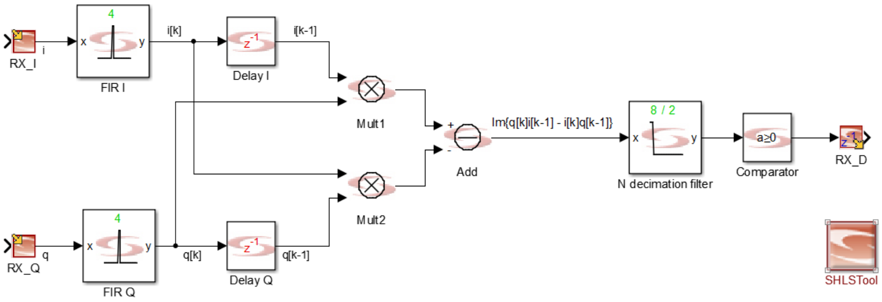


Figure 13: Simulink model of the HDL synthesizable GMSK demodulator.

4.3.2 Resource utilization

The FPGA logic resource utilization of the GMSK modulation based PHY layer and that of the infrastructure HDL components is summarized in Table 10. The AFE and AMBA interfaces represent the infrastructure part of the FPGA design and take 5% of the available FPGA logic resources. The resource requirement of the AFE interface is the same 0.6% in every application, but that of the AMBA interface, which in this case amounts to 4.4%, varies based on the amount of the defined memory mapped registers. The latter also employs two block RAMs for transmit and receive data FIFOs, therefore, approximately 90-95% of the FPGA fabric resources are available for the custom SDR HDL design.

The reference GMSK modulation-based PHY layer design takes 40.5% of the logic resources in total, of which the transmit path is responsible for 14.2% and the receive path for 21.3%. The vast majority, over three-quarters, of the transmit path consumed resources is associated with the Gaussian pulse shaping filter of the modulator, implemented as a full-parallel structure 17-tap FIR filter, constructed from general logic cells due to the lack of hardware multipliers. Observe that the requirements of the Gaussian FIR filter may be relaxed without significantly distorting the transmitted waveforms. In contrast, the FIR and decimation filters in the demodulator take up approximately a quarter of the total resources allocated to the receiver path. The rest of the logic

Component	**Logic cells**	**Block RAM**
AFE interface	68 (0.6%)	0 (0%)
GMSK modulator	1638 (14.2%)	1 (4%)
GMSK demodulator	1824 (15.8%)	1 (4%)
GMSK synchronizer	631 (5.5%)	0 (0%)
AMBA interface	508 (4.4%)	0 (0%)
Total	**4669 (40.5%)**	**2 (8%)**

Table 10: FPGA logic resource utilization of the GMSK transceiver reference design.

cells is used to realize the delay registers and the various operators, such as multipliers, comparators and XOR logic in the demodulator and the synchronizer.

The presented reference communication protocol design realized the entire PHY layer in the FPGA fabric. The custom HDL components that define the modulator, demodulator and synchronizer functions were synthesized from Simulink models without any significant attempt for optimization. Therefore, the analysis of the prodigally used logic cells indicates that the total amount of FPGA logic resources allows for experimentation with PHY layers of complexity typical in WSNs.

4.3.3 Power consumption

To fully characterize the power consumption of the MarmotE SDR platform, its current draw was compared to two fundamentally different radio transceiver solutions. The CC1000 is a highly-integrated, low-cost, low-power commodity RF transceiver chip using FSK modulation in the 433 MHz band. The power consumption for the CC1000 was calculated based on the datasheet specifications assuming 3.3 V supply voltage and the crystal oscillator turned-on. The other reference for comparison was the USRP N210 [50] mid-price SDR, which offers full-stack design flexibility at a higher price and power consumption. The latter was calculated based on the measured total current draw of the USRP at 6 V with an SBX daughterboard attached to it. Thus, this value does not include the consumption of desktop computer additionally required for the operation. The MarmotE SDR power consumption was measured by the bottom-layer power supply monitor, and it included the consumption of the 2.4 GHz radio front-end and the mixed-signal processing board, with the SmartFusion MSS, FPGA fabric and the AFE running at 10 MHz. Note that unlike desktop SDR platforms, the MarmotE SDR operation does not require a desktop computer.

The power consumption of the three approaches is compared in Figure 14 in three common scenarios in WSN duty cycle operation: sleep, transmit and receive mode. As WSN nodes usually spend most of their time dormant, sleep mode is expected to reduce current draw to the fraction of that of active modes. The CC1000 offers true sleep mode with power consumption less than 1 mW, the MarmotE SDR consumes 70.8 mW, while the USRP N210 does not provide similar low-power feature. In-depth analysis of the MarmotE SDR sleep mode showed that the mixed-signal module is responsible for 72% of the 70.8 mW dissipated, while the radio front-end for the remaining 28%. During sleep mode all external peripherals were disabled, making the SmartFusion SoPC the main contributer to the 50 mW consumed. Unfortunately, the current SmartFusion MSS lacks advanced low-power modes that achieve sub-1 mW sleep power without turning the power rails off. Switching the power rails off has the adverse effect of the eSRAM losing its content and,

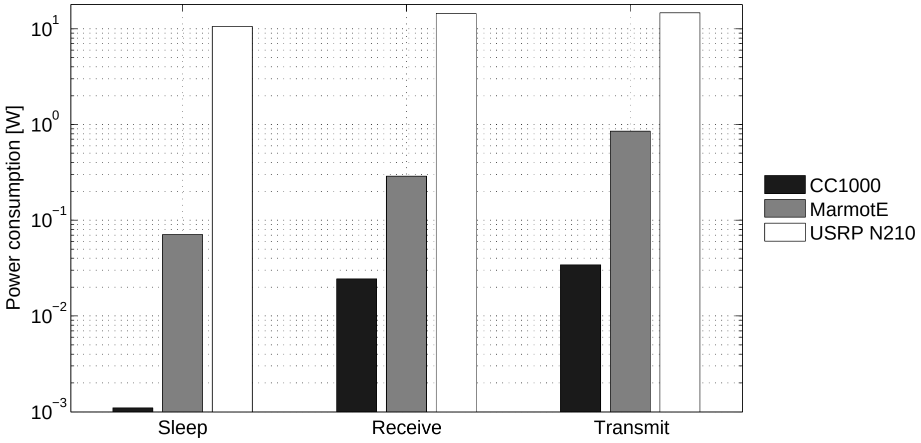


Figure 14: Power consumption comparison of the CC1000 integrated racio chip, the MarmotE SDR platform and the USRP N210 desktop SDR in typical WSN operating modes.

therefore, the application losing its state. As reinitializing the application, saving and restoring its state variables results in significant wake up time penalties, the SmartFusion sleep mode was defined with the integrated AFE and ACE powered off, the FPGA put in reset mode, the required MSS peripherals running on 32 kHz and the Cortex-M3 halted, waiting for interrupt. The latter SmartFusion configuration was found to yield the lowest-power mode from which the system can wake up in only a few clock cycles. While the 50 mW power draw is significantly higher than that of our previous microcontroller plus IGLOO flash FPGA approach [17], this was consider a trade-off for the high-bandwidth, on-chip AMBA bus interface between the MSS and the FPGA fabric. As microcontrollers with ultra-low-power sleep mode already exist and IGLOO flash FPGAs consume less than 60 μW in Flash*Freeze mode [32], the manufacturing technology is expected to lower the static power consumption of next generation flash FPGA based SoPCs significantly. The 2.4 GHz radio front-end had two main components enabled during sleep mode measurements, the MAX2830 transceiver chip and the TCXO. As the RF transceiver chip consumes less than 1 mA in shutdown mode, the main contributor to the 20 mW power was the TCXO. Even though the current radio front-end module keeps the TCXO always-on, this was a design decision and it could be turned off in future versions.

In receive mode, the MarmotE SDR consumed 287.4 mW, approximately 12 times more power than the CC1000 (24.4 mW) but 50 times less than the USRP N210 (14400 mW). Out of the 287.4 mW, the SoPC and AFE on the mixed-signal module dissipated approximately 80 mW and 15 mW, while the MAX2830 and the TCXO on the radio front-end board around 172 mW and 20 mW, respectively. Comparing the transmit mode power consumption at 0 dBm nominal transmit power, the MarmotE SDR dissipated 851.7 mW, the CC1000 24 times less (34.3 mW), while the USRP N210 51 times more (14700 mW). The SoPC, AFE and TCXO consumed the same as in receive mode, while the MAX2830 transmit section and the integrated power amplifier drew roughly 280 mW and 450 mW, respectively.

The MarmotE SDR is estimated to continuously operate for over 24 hours in transmit mode (0 dBm) and to run for over 10 days in sleep mode on the fully charged 6000 mAh 3.7 V Li-Ion battery. Therefore, considering its portability and duty cycling capability, the platform is expected to enable the short-term and multi-day evaluation of custom PHY layer communication protocols in both indoor and outdoor deployed scenarios. The development of such PHY layers, in turn, is further promoted by the support of the accompanying HLS tool-based framework.

5 Conclusion

Long-term deployed WSN nodes face low-power requirements that have been combated on several fronts in the past decade. Power consumption of computational resources have been rapidly reduced through advances in semiconductor process technology, but the same does not hold for the radio communication interface. Significant energy may be saved related to wireless communication in WSNs through the design of full-vertical communication protocol stacks tailored to the specific WSN application. Traditional WSN nodes utilize highly integrated COTS radio transceiver chips that implement the lower layers in ASIC, and therefore, reduce design flexibility of those layers. SDRs, on the other hand, give full access to the entire stack and allow for rapid prototyping, but their power consumption is prohibitively large for practical battery-based operation.

The MarmotE SDR intends to strike a balance between the two requirements, and to provide a deployable WSN platform with SDR capabilities for communication stack research. Through the flash FPGA based SoPC architecture and the support of a HLS-based development workflow, the platform allows for rapid and flexible design of complete network stacks from baseband processing in the PHY layer and up, which was generally not possible with the existing WSN platforms. The computational resources offered by MarmotE SDR are sufficient for prototyping simple to moderate complexity PHY layers, albeit they are less abundant than in a typical desktop connected SDR. In return, the MarmotE SDR consumes an order of magnitude less power than a desktop SDR. Its power consumption with the 2.4 GHz radio front-end is approximately 0.07 W in sleep, 0.29 W in receive and 0.8–1.5 W in transmit mode, depending on the transmit power. Although still significantly higher than for a COTS integrated transceiver, these values are in the targeted range and readily allow for day-long continuous battery based operation, which can further be extended by duty cycling. This confirms that flash FPGA-based SoPCs represent a promising architectural approach for low-power SDR platforms. Furthermore, the recently introduced generation of SmartFusion SoPC devices incorporate embedded multipliers and the Flash*Freeze mode, increasing the effective computational power and reducing both the wakeup-time and the sleep mode power consumption, respectively.

Deployment of MarmotE SDR nodes configured with a prototype network stack allows to collect real world feedback on the communication protocol performance, rather than to rely solely on simulation results. An experimentally verified protocol stack can then later be implemented as a highly integrated transceiver, resulting in much smaller size and significantly smaller power budget. Therefore, the MarmotE SDR platform is expected to serve as a springboard for several future low-power transceiver solutions.

The hardware, HDL and software design files of the MarmotE SDR platform are open-source and freely available for download at http://marmote.googlecode.com.

CHAPTER III

WIRELESS COMMUNICATION PROTOCOLS

1 Introduction

Wireless sensor nodes communicate through a shared medium to coordinate collaborative sensing and to gather sensor data. While the underlying physical medium is essentially the same as for traditional wireless systems, the requirements for the sensor node communication differ in many aspects. The sensor nodes are usually deployed densely in remote areas and in an ad hoc fashion [11][12], where the topology is expected to change due to node failures [73]. Furthermore, the nodes typically employ low-cost radio chips that support only low data rate modulations schemes and limited communication range [3][2], nevertheless, they are required to operate over a long period of time, handle traffic fluctuations and cover large areas [13][16].

Despite the common characteristics, the actual communication bandwidth, coverage and network lifetime expectations are generally set by the particular WSN application. The differences between these expectations, in turn, make convergence to a single communication protocol stack unlikely. Instead, a number of wireless protocols with different and adjustable lower stack layers is expected to surface, a few of which accommodate the need of a given WSN application.

In this chapter, Section 2 discusses the common communication protocol design criteria and gives an overview of the existing WSN radio stacks. Then, Section 3 reviews the fundamentals of spread spectrum communications and elaborates on the proposed WSN protocol *physical* layer. Leveraging the flexibility of the MarmotE SDR platform, a prototype design is presented in Section 4, and its performance is evaluated in Section 5. Lastly, the conclusions are drawn in Section 6.

2 Background

2.1 Protocol Design Considerations

Traditional wireless communications protocols are generally optimized for throughput, latency, bandwidth efficiency and fairness. In contrast, the design of WSN communication protocols is primarily driven by *energy efficiency*, *scalability* and specific *communication patterns*.

2.1.1 Scalability and adaptivity

The deployment of large-scale infrastructureless wireless sensor networks calls for communication protocols that exhibit minimal performance degradation as the number of nodes increases. Such WSN protocols are also often required to self-configure a reliable multi-hop network and adapt to changes in the network topology [74]. Furthermore, topology changes are expected throughout the network lifetime due to nodes failing, moving, joining the network, or temporarily disconnecting because of varying interference conditions.

The above scalability and adaptivity requirements have generally been addressed through the medium access control (MAC) sublayer, where the rigid schedule-based approaches turned out to be ill suited for several WSN applications. Thus, while several modified time division multiple access (TDMA) schemes have been proposed for their power efficiency [75][74][76], the prominent adaptive and scalable protocols extensively rely on the asynchronism of carrier sense multiple access (CSMA) schemes instead [77][78][79][80][81], see Section 2.2.

2.1.2 Communication patterns

The ad hoc nature of the network topology and the characteristics of the applications make certain communication patterns distinct to WSNs. First, typical WSN applications require low communication data rate, being in the order of tens of bytes per second [7], but the traffic shape is often bursty due to external events triggering similar responses from nearby sensor nodes.

Second, the characteristic sensor network traffic patterns may be partitioned into three distinct classes: broadcast, convergecast and local gossip [82]. The broadcast pattern is generally initiated by the base station to disseminate data, that is to send control commands or issue query requests to all sensor nodes. In the convergecast pattern a set of nodes communicates to one specific node and data flows only towards the sink node. This pattern is most often used to report sensor measurements to the base station after a trigger event. Finally, local gossip is defined as communication between a set of neighboring nodes to set up a local measurement or evaluate a commonly sensed event.

2.1.3 Energy efficiency

The power source of traditional wireless devices is typically either rechargeable battery or mains power. On the other hand, WSN nodes are generally designed to be disposably cheap and deployed in unattended areas, which makes replacing or recharging their batteries impractical. Therefore, energy scarcity is of utmost concern in WSNs and improving network lifetime becomes the primary goal in almost every application.

In typical WSN applications, aggressive duty-cycling is used, since the active mode power consumption is dominated by the communication subsystem and the radio chip draws similarly high power in both transmit and receive mode, see Table 6. Therefore, the use of energy efficient communication protocols offers a straightforward way to conserve energy. The first step towards the design of such protocols is to identify the main sources of energy waste. Focusing on the MAC layer, these sources are commonly associated with collision, overhearing, protocol overhead and idle listening.

Collision. Two or more packets sent by different transmitters collide when their reception overlaps at the receiver in time, and their destructive interference prevents proper recovery of the content. In such case, the energy used for transmission and reception is wasted as the collided packets have to be discarded and retransmitted.

In TDMA-based wireless protocols, collision avoidance is inherently achieved by assigning non-overlapping timeslots to the transmitters. However, this schedule is difficult to adapt to topology changes common in WSNs. Contention-based CSMA/CA protocols usually rely on short ready-to-send (RTS) and clear-to-send (CTS) handshake packets to prevent the collision of long data packets. On the other hand, WSN data packets are usually short and the RTS/CTS packets may also collide, therefore, the RTS/CTS packets may generate a significant amount of protocol overhead. Moreover, excessively high transmit power levels without adaptivity are not just inherently wasteful, but also increase the effective area of collisions.

Protocol overhead. The transmission and reception of data unrelated to the application constitutes protocol overhead. The excessive use of control packets, and the improperly chosen frame formats create extra traffic and consequently waste energy.

The protocol overhead is often combated through the use of short packet frame structures and reduced number of control messages. Several CSMA protocols achieve this by omitting the use of RTS/CTS handshake packets and explicit acknowledgments.

Overhearing. The reception of unicast packets, not addressed to the receiver, or redundant broadcast packets causes overhearing. The content of such packets is irrelevant to the receiver and is discarded, therefore, the energy used for their reception is wasted.

Data and control packets may incorporate the destination, source and length fields in the beginning of the frame to allow unintended receivers to go to sleep sooner and reduce overhearing. Similarly, certain transmitter initiated protocols rely on short wake-up packets [79][78] instead of elongated preambles [77] to achieve a similar effect.

Idle listening. Waiting for potential incoming packets in receive mode while there is no communication activity in the channel is generally referred to as idle listening. In typical low-traffic communication patterns, the channel is expected to be inactive for long periods of time. Therefore, spending unreasonably long time with idle listening often becomes the major source of energy waste.

The problem of idle listening is commonly addressed through long sleep cycles accompanied by accurately scheduled rendezvous between the transmitters and the receivers. Similarly to collision avoidance, this is inherently achieved in pure TDMA-based protocols where the schedule is explicitly known at every node. However, clock drift and temperature variations might severely affect the precision of synchronization. On the other hand, the effective and efficient scheduling of rendezvous is the cardinal question in contention-based protocols, for which several transmitter initiated protocols based on preamble sampling [77] and cyclic transmission [78], as well as multiple receiver initiated protocols [80][81], have been proposed.

2.2 Existing Radio Stacks

The low-level interface between the MCU and the radio chip provides reasonable flexibility for radio stack design, which gave way to experimentation primarily with MAC protocols. Aiming for power efficiency, while keeping the resource constraints of the simple MCUs in mind, more than 80 different low-power MAC protocols have been proposed [7][83][84]. These MAC approaches are generally categorized as *synchronous* or *asynchronous* protocols [79], see Table 11.

Synchronous protocols essentially use duty-cycled TDMA with a local [74][76] or global [75] schedule. However, the rigidity of the schedules make the performance of pure TDMA-based protocols drop significantly in the presence of dynamic network topology changes. *Hybrid* protocols [85][73] add asynchronous elements to a synchronous TDMA-based scheme to increase its flexibility. Finally, *asynchronous* low-power protocols improve scalability and adaptivity by decoupling the transmitter and receiver sleep schedules. The asynchronous protocols are generally divided into *transmitter initiated* [77][78][79] and *receiver initiated* [80][81] approaches, based on whether the data exchange starts with the transmitter or the receiver accessing the channel first.

Synchronous	Global schedule	D-MAC [75]
	Local schedule	S-MAC [74], T-MAC [76]
Asynchronous	Transmitter initiated	B-MAC [77], X-MAC [78], BoX-MAC-1, BoX-MAC-2 [79]
	Receiver initiated	RI-MAC [80], A-MAC [81]
Hybrid		WiseMAC [85], Z-MAC [73]

Table 11: Categorized list of low-power MAC protocols proposed for WSNs.

Despite the apparent divergence of research MAC protocols, several standards have emerged to solidify the various layers of communication stacks designed for low-cost, short range wireless devices. The rest of this section presents the major research and standardized WSN radio stacks of the past two decades with a special focus on the MAC layer.

2.2.1 Research stacks

S-MAC. The Sensor MAC [74] is a slotted protocol inspired by the PAMAS [86], in which the neighboring nodes form a virtual cluster to synchronize themselves to a slot structure with common active period. The S-MAC slot comprises fixed-length active and sleep periods, which implicitly define the duty cycling of the WSN. The active period is further divided into a SYNC phase and a contention based data exchange phase. Nodes synchronize their clocks and join the network in the SYNC phase and transfer data with RTS/CTS handshake scheme and optional message passing support for long messages in the data exchange phase. In general, S-MAC offers a simple and efficient way to handle network topology changes and perform duty cycling. The adjustable ratio of the active and sleep periods allows to trade-off energy efficiency for latency, however, its predefined value severely limits the adaptiveness to varying traffic conditions.

T-MAC. The Timeout MAC [76] is a slotted protocol that extends S-MAC by introducing an adaptive duty cycle scheme. T-MAC holds on to the virtual cluster concept and also uses RTS/CTS handshake for data exchange. However, it relies on shorter listening period and a time-out mechanism to adaptively change the length of the active period within the fixed-length slot. Due to the adaptive duty cycling, T-MAC achieves better energy efficiency than S-MAC under fluctuating network traffic. On the other hand, the varying length active period and the aggressive power-down policy leads to listen-period synchronization issues (e.g. early sleeping).

B-MAC. The Berkeley MAC [77] is an asynchronous CSMA protocol that supports on-demand reconfiguration to optimize for power efficiency, latency or throughput. B-MAC features adaptive preamble sampling to reduce duty cycling and idle listening, provides an improved channel arbitration algorithm and relies solely on PHY layer carrier sensing to wake-up a receiver node. B-MAC was implemented as a TinyOS component with a convenient interface towards the higher-level services to dynamically configure the MAC parameters, which allows to adapt to changing traffic conditions. The drawback of B-MAC is, however, that after wake-up, a receiver must remain active for the entire duration of the preamble before it can acquire the packet.

D-MAC. The Data-gathering MAC [75] is a slotted protocol built upon S-MAC and developed for efficient convergecast communication. It achieves low latency in the uplink by aligning the slot structure based on the node position in the data gathering (routing) tree. The main disadvantage of D-MAC is that it heavily relies on the routing tree, therefore, it handles topology changes inefficiently.

WiseMAC. WiseMAC [85] is a contention-based protocol proposed for downlink communication of infrastructure WSNs, where the CSMA scheme is combined with preamble sampling to reduce the time spent on idle listening. In WiseMAC the nodes maintain a schedule of the neighboring receivers and start to transmit only when the intended receiver is about to sample the channel. The preamble length is dynamically adjusted based on the elapsed time since the last transmission to account for clock drifts, save energy and implicitly adapt to traffic variations. The weakness of WiseMAC is broadcast communication, which calls for stretched preambles to accommodate the schedule of all receivers and results in significant amount of redundant retransmissions.

X-MAC. The X-MAC [78] contention based CSMA protocol is essentially the successor of B-MAC, specifically tailored for the 802.15.4 compliant CC2420 radio. The main difference between the two is that X-MAC uses link layer wake-up packets, whereas B-MAC relies solely on PHY layer preamble sampling. The advantage of X-MAC is that can end wake-up transmission twice as fast as B-MAC, however, receive checks usually last at least an order of magnitude longer.

BoX-MAC. The BoX-MAC [79] CSMA protocols are the direct descendants of B-MAC and X-MAC, and represent the recent generation of on-the-fly configurable MACs. While B-MAC exploits only the PHY layer and X-MAC only the link layer, BoX-MAC-1 and BoX-MAC-2 are cross-layer protocols that incorporate information from both, but with different emphasis. BoX-MAC-1 relies predominantly on the PHY layer by sampling longer preambles, while BoX-MAC-2 primarily on the link layer by checking for shorter wake-up packets. The BoX-MAC protocols consume up to 30% and 50% less energy than B-MAC and X-MAC, respectively, and yield up to 46% higher throughput than X-MAC. Both BoX-MAC protocols became part of TinyOS and together and allow for adaptive solutions for a wide range of traffic conditions.

RI-MAC. The RI-MAC [80] is an asynchronous CSMA duty cycling protocol that uses receiver-initiated data transmission. The goal of RI-MAC is to find rendezvous time for data exchange, while minimizing channel occupancy and idle listening, without maintaining synchronized schedules between the transmitters and receivers. In RI-MAC, the receiver nodes wake up periodically and transmit a beacon. A node that intends to exchange data waits for this beacon and initiates data transmission in response. Upon successful reception, the receiver then sends another beacon messages with the same structure both to acknowledge the received data and to initiate the immediate transmission of possible queued packet. If no further data transmission takes place for some extra time, called *dwell time*, the receiver enters sleep mode. Compared to X-MAC, RI-MAC achieves higher performance in terms of packet delivery and latency at comparable energy consumption, even at low-traffic conditions, for which X-MAC is optimized for. RI-MAC handles bursty and other type of traffic fluctuations well, but offers only unicast service. However, a broadcast service support is proposed for RI-MAC in [87] and [81].

A-MAC. The A-MAC [81] is a receiver-initiated asynchronous protocol that incorporates several low-power services, such as wake-up, unicast, broadcast, pollcast and discovery, into a unified component. Similarly to RI-MAC, data exchange starts with the transmitter listening until the receiver sends a beacon frame. However, instead of responding to the beacon with the data packet immediately, the transmitter sends a precisely timed acknowledgment frame before the data content. Finally, the receiver sends an additional beacon to initiate possible further transmissions. The basic idea behind the beacon-acknowledgment exchange based *backcast* synchronization primitive is that in case of collision, the (auto) acknowledgments from the transmitters are precisely timed and collide non-destructively at the receiver. Therefore, the receiver is able to decode the superposition of the acknowledgments to infer that more incoming traffic is pending and to retransmit the beacon with an explicit contention window. A-MAC offers its service abstractions based on the backcast synchronization primitive and allows to optionally use secondary channels for data exchange, which improves scalability.

2.2.2 Standardized stacks

IEEE 802.15.1. The IEEE 802.15.1 standard, or Bluetooth technology, was originally designed to replace the wired communication between cellular phones and other devices. The protocol stack defined the entire physical and data link layers with synchronous medium access, furthermore, it proposed a middleware layer for higher level entities (e.g. profiles). The physical layer supported only GFSK modulation with its spectrum spread by frequency-hopping (FHSS) in the 2.4 GHz band, see Section 3.1. Subsequent revisions of the protocol added support for other modulation schemes and primarily focused on improving speed, but the latest improvements in Bluetooth Low Energy also address low-power operation. Although the number of supported network topologies is limited, several WSN platforms experimented with Bluetooth-based radios [43][88].

ANT. ANT [89] is a proprietary technology for wireless communication between low-power sensors. The ANT protocol stack specification spans the physical, data link, network and transport layers. Similarly to Bluetooth, the ANT radio operates in the 2.4 GHz ISM band and uses GMSK modulation, however, without FHSS. The upper layers of the ANT protocol support a wide range of scalable network topologies, including certain mesh topologies, that mostly rely on synchronous medium access. Possibly due to their late introduction, however, ANT radio chips have not become widely used in the WSN research community.

IEEE 802.15.4. The IEEE 802.15.4 is a communication standard for low-rate wireless personal area networks. In contrast to Bluetooth and ANT, the 802.15.4 specifies only the physical layer and parts of the media access sublayer, leaving the upper portion of the protocol stack undefined. Thus, it provides basis for several custom protocol stacks and for other standards, such as WirelessHART and ZigBee. The basic 802.15.4 standard defines multiple physical layers with O-QPSK, BPSK, ASK, GFSK and UWB modulation schemes, however, typical 802.15.4 compliant commodity radio chips operate in the 2.4 GHz ISM band and use O-QPSK modulation with direct-sequence spectrum spreading (DSSS) to tolerate in-band interference, see Section 3.1. The relatively low-level interface of these radio transceivers inspired several sensor node platforms [2][3][90], as it allows for experimentation with custom MAC protocols [74][76][85].

ZigBee. ZigBee is a high-level communication protocol specification for low data rate, low-power wireless networks. The ZigBee standard specifies the data link and network layers on top of the IEEE 802.15.4 PHY and MAC to add routing and networking functionality with support for star, cluster tree and mesh topologies. The standard also defines parts of the application layer with high-level entity definitions and services for device discovery and secure communication.

WirelessHART. WirelessHART is a wireless communication protocol primarily designed for industrial applications, such as process monitoring and control. Similarly to ZigBee, WirelessHART adds further layers to the IEEE 802.15.4 protocol stack to support networking with an emphasis on reliability, security and power-aware operation. Reliability is mainly advocated through the built-in redundancy of the mesh networking topology and the interference tolerance provided by frequency hopping, while security is ensured through authentication and encryption services.

In the past decades, WSN communication protocols have been extensively researched with an increased attention to the MAC layer. Despite the wide variety of proposed MAC protocols, several surveys [7][83][84] agree that there is little convergence, and the particular protocol choice is expected to be largely application dependent. Recent low-power protocols [79][81] address this issue by offering on-the-fly configurable cross-layer solutions. However, the design of these protocols increasingly relies on precise timing and exploits certain properties of PHY-layer waveforms. Therefore, the imprecision of software controlled timers and the inflexibility posed by the rigid silicon implementation of COTS radio transceivers clearly became the limiting factors for experimentation with the lower layers of the radio stack.

In search for alternative access schemes for WSNs, concepts are borrowed from cellular communications to trade-off collision avoidance for additional computational complexity, which is then distributed asymmetrically between the low-complexity transmitter nodes and a resourceful receiver basestation. In the following, Section 3 reviews the basics of spread spectrum communications, and its potentials in the context of WSNs. After describing the proposed communication protocol tailored to WSNs, Section 4 presents its prototype implementation using the MarmotE SDR platform. The performance of the protocol is then evaluated based on real-world measurements in Section 5 and the final conclusions are drawn in Section 6.

3 Simultaneous Access Communication

The most fundamental task of wireless communication in a WSN is to transfer the sensory data from the deployed sensors to a central repository for processing. This is generally achieved using *sensor nodes* equipped with highly integrated radio chips of Section 2.2, along with a *basestation node* of essentially the same radio transceiver architecture but attached to a laptop or desktop computer. One disadvantage of this approach is that it prescribes the use of a CSMA/CA or TDMA access scheme throughout the network, including the vicinity of the basestation. Thus, it relies on the same symmetric PHY layer communication link across the entire network and fails to take advantage of the increased resources usually available at the basestation.

However, in practical WSN deployments the basestation node is generally connected to a computer, therefore, it is reasonable to assume that the basestation is not subject to the resource and power constraints of the battery operated sensor nodes. Given the necessary computational power at the basestation, in the form of FPGAs or high-end processors, and the ability to define custom PHY layer waveforms between the basestation and the directly connected sensor nodes then allows to experiment with multiple access schemes borrowed from cellular communications. Observe that the MarmotE SDR nodes and a desktop SDR provide the necessary flexibility and resources for defining arbitrary such PHY layers.

Since the vicinity of the basestation is a generally congested area, and often the bottleneck of WSN communication, avoiding collisions and increasing hop-distances there holds the promise to reduce and balance energy consumption, and consequently to extend the network lifetime. Therefore, based on the above assumptions of *customizable baseband waveforms* and *resourceful basestation*, the following section concentrates on the PHY layer design of a WSN communications protocol. First, it reviews the principles of spread-spectrum (SS) communication techniques. Then, it proposes a PHY layer for wireless sensor networkings stacks, which allows sensor nodes to report their sensory data to the basestation asynchronously and simultaneously.

3.1 Overview

The origin of spectrum spreading dates back almost a century, yet various forms of the technique are still widely used in current communication systems. Spread spectrum (SS) techniques have several advantages that led to their popularity. The most prominent ones are *jamming avoidance* and the *low-probability of intercept* [91], both of which were heavily sought by military communication systems. The bandwidth increase that accompanies the SS strategy makes the jamming and detection of the signal more difficult because it requires a larger frequency band to be jammed or monitored, respectively. Furthermore, the power spectral density of the signal is implicitly lowered proportionally to the *spreading factor*, allowing the communication system to operate well below the noise floor. Another benefit of spectrum spreading is that the increased bandwidth allows for *high-resolution Time of Arrival (TOA)* measurements. Precise TOA estimates, in turn, enable accurate ranging in radar or GPS [92], and discrimination against the multi-path delayed versions of the transmitted signal in the receiver of a communications system. An equally important advantage of SS communications is, however, the ability to reject independent in-band interference. Interference rejection, proportional to the spreading factor, applies both to adversary jamming and to other communication systems operating in the same band. This latter asynchronous form of spectrum sharing, in turn, provides the basis of the *multiple-access capability* of SS systems and is commonly referred to as code-division multiple access (CDMA).

Two widely employed SS schemes are *frequency hopping* (FHSS) and *direct-sequence* (DSSS) modulation. Denoting the basic pulse waveform with $p(t)$, the former FHSS generated waveform can be written as

$$c(t) = \sum_n e^{j(2\pi f_n t + \phi_n)}\, p(t - nT_h), \tag{3.1}$$

where the pulse duration equals to the T_h *hop time* and $\{f_n\}$ is a pseudorandomly generated sequence of frequency shifts [91, p. 10]. The FHSS modulation is used in the Bluetooth wireless communication protocol standard and for radio-controlled aircrafts.

Similarly, the DSSS modulated waveform can be expressed as

$$c(t) = \sum_n c_n\, p(t - nT_c), \tag{3.2}$$

where T_c denotes the *chip time*, and the $\{c_n\}$ pseudorandom sequence linearly modulates the series of pulse waveforms of T_c duration each. The DSSS technique is utilized in GPS and in several communications standards, including the 802.15.4, IS-95 and W-CDMA.

Note, however that the Bluetooth and 802.15.4 standard PHY layers employ spectrum spreading in the form of FHSS and moderate DSSS only to combat interference originating from *outside* the network. In contrast, third-generation cellular communications technologies utilize aggressive DSSS to also suppress *in-network* interference, thus, provide multiple-access capability.

Based on the above properties of DSSS modulation technique, its use in the PHY layer for direct communication from the wireless sensor nodes to the basestation holds the following promises:

- Rejection of independent in-band interference and jamming, analogously to the tolerance offered by the PHY layer of the 802.15.4 standard.
- Contention-free, asynchronous multiple-access capability that allows for simultaneous packet transmissions, and requires no synchronization between the sensor nodes. The multiple-access capability as a function of node density and spreading factor is investigated in Section 5.1.
- Asymmetric communication link that shifts the processing burden from the sensor nodes to the basestation. This allows to keep the transmitter simple at the expense of increased complexity in the receiver architecture, as described in Section 4.
- Ability to increase the hop-distance for a given fixed transmission power level by changing only the spreading factor. The attainable reliable hop-distance as a function of spreading is examined in details in Section 5.2.
- Improved communications security through inherent resistance to eavesdropping. The detection and demodulation of the DSSS messages require an exact replica of the spreading pseudo-noise sequence, see Section 3.2.1. Therefore, proper construction and allocation of the spreading codes can provide an additional level of security.
- Relative phase offset measurement method for time-differences of arrival (TDOA) based node localization, a viable alternative to the single- and multi-carrier solutions described in Appendix B-1 and Section 3.2, respectively.

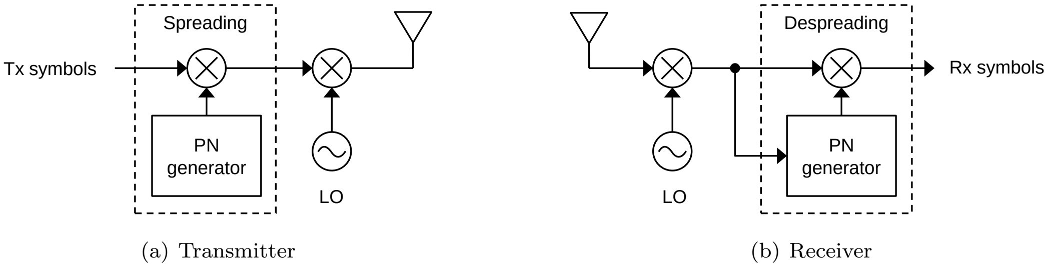
3.2 Spread-Spectrum System Model

The fundamental tasks of DSSS are to spread the data symbols at the transmitter with an appropriate pseudo-noise (PN) sequence, and to detect, synchronize and despread the received signal at the receiver. This section reviews the principles of the DS-CDMA communication scheme, primarily based on [91] and [93], and describes the proposed communication protocol.

3.2.1 Pseudo-noise sequences

With the DSSS modulation technique, the transmitted data symbols are linearly modulated by the independent $\{c_n\}$ pseudo-noise code sequence of *chips*, as illustrated in Figure 15(a). The spreading PN code is generated in a deterministic way, however, it acts as a random sequence of ± 1 valued chips with statistical properties similar to noise if the corresponding generator algorithm is unknown.

Furthermore, the spreading sequence has significantly higher chip rate than the original data symbols, therefore, it expands the bandwidth of the original signal proportionally to the spreading factor. Knowing the exact same PN sequence at the receiver, its precisely timed correlation with the received noise-like signal allows for the reconstruction of the original data symbols, see Figure 15(b). When properly constructed unique PN-codes are assigned to the sensor nodes, the communication links in the vicinity of the basestation effectively allow for an asynchronous DS-CDMA scheme.



(a) Transmitter

(b) Receiver

Figure 15: Simplified direct-sequence spread spectrum (DSSS) transmitter (a) and receiver (b) block diagrams.

Two fundamental performance measures of the PN sequences are their auto-correlation and cross-correlation functions. The auto-correlation function is defined as

$$R_i(\tau) = \int_{-N_cT_c/2}^{N_cT_c/2} c_i(t)\, c_i(t+\tau)\, dt, \tag{3.3}$$

where N_c is the length of the $\{c_i\}$ PN sequence and T_c is the chip period. The auto-correlation function is expected to have a high correlation peak at $\tau = 0$ and close to zero value at $\tau \neq 0$ to enable accurate detection of the spread signals. Meanwhile, the cross-correlation function is defined as

$$R_{i,j}(\tau) = \int_{-N_cT_c/2}^{N_cT_c/2} c_i(t)\, c_j(t+\tau)\, dt, \tag{3.4}$$

where $\{c_i\}$ and $\{c_j\}$ are PN sequences of length N_c and T_c is the chip duration. $R_{i,j}(\tau)$ is expected to be close to zero for *any* value of τ, as it measures the agreement between sequences $\{c_i\}$ and $\{c_j\}$, and consequently characterizes the possible interference between the two transmitters using these spreading sequences.

PN sequences with outstanding auto-correlation properties can be generated using a single linear-feedback shift register. The class of exactly $2^L - 1$ long sequences, which can be produced by an L-stage shift register, is called maximal-length sequences or simply *m-sequences* [91, p. 283–284]. Unfortunately, the cross-correlation properties of m-sequences are less desirable, and the number of cyclically distinct m-sequences is finite for a given shift register length L. Therefore, in a multiple-access scenario with a large number of wireless nodes, either the m-sequences have to be carefully chosen or other sequences have to be considered. One possible alternative is the use of Gold sequences that are constructed by concatenating two linear-feedback shift registers of identical length [94]. Gold sequences exhibit good cross-correlation properties and are the basis of GPS and several CDMA communications standards.

The DSSS communication protocols may further be classified based on the ratio of the spreading PN sequence length and the symbol time. A *short-code* system spreads each symbol with the same short PN pattern, thus $N_c T_c = T_s$. In contrast, a *long-code* system assigns a different chip pattern to each upcoming symbol, therefore, the associated total sequence length is significantly longer than the symbol duration, $N_c T_c \gg T_s$. Observe that a modified short-code solution is employed in the 802.15.4 standard, where each data symbol, comprising of four bits, maps to one of the sixteen predefined 32-long PN sequences. Such mapping, in turn, corresponds to spreading the original data sequence by an effective factor of 8.

3.2.2 Transmitter side

In the proposed DSSS protocol, the packet payload is first prepended a PHY header, as shown in Figure 16. Then the low-rate binary data symbols are differentially encoded and spread with the unique PN sequence of the sensor node. The PN generator is operated at a significantly higher chip-rate to spread the data symbols. Moreover, it is reset only at the start of the packet transmission, as opposed to after each data symbol, therefore, the transmitter follows the long-code approach for PN sequence generation. Finally, the spread chips are filtered and directly modulated onto the carrier. The block diagram of the corresponding DSSS differential binary phase-shift (D-BPSK) keying modulator is shown in Figure 16.

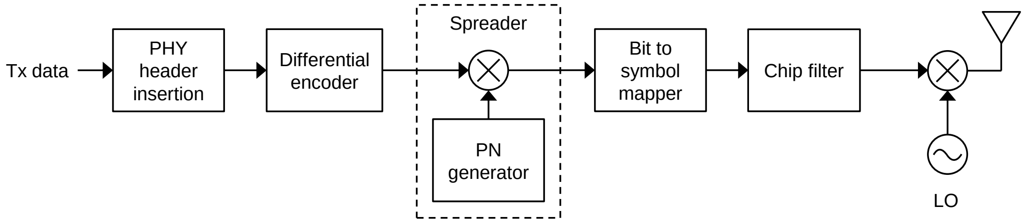


Figure 16: Block diagram of the proposed DS-CDMA transmitter for the wireless sensor nodes.

The modulated spread spectrum D-BPSK baseband signal of the k^{th} node can be written as

$$u_T(t) = \sum_i c_i^q d_{\lfloor i/\mathrm{SF} \rfloor}\, p(t - iT_c), \tag{3.5}$$

where $\{c_i^q\}$ is the PN sequence assigned to the q^{th} node with $c_i^q \in$ {-1, 1}, SF is the spreading factor, $\lfloor \cdot \rfloor$ denotes the integer part and $\{d_n\}$ is the differentially encoded data sequence with the $d_n \in$ {-1, 1} data symbols allowed to change every SF chip times. Finally, $p(t)$ is the pulse shape defined by the chip filter. Then, the transmitted direct-spread D-BPSK passband signal becomes

$$s_T(t) = \mathrm{Re}\left[u_T(t)\, e^{j(2\pi f_c t + \phi_T)}\right]. \tag{3.6}$$

3.2.3 Receiver side

The fundamental functions of a DSSS receiver are to *despread* the incoming signal prior to demodulation, and to perform the accompanying *synchronization* tasks. Despreading requires a local replica of the $\{c_i^q\}$ PN sequence at the receiver, and its perfect alignment to the received waveform. Synchronization of the local PN sequence to the one superimposed on the received signal is generally accomplished in two stages. Initially, the two sequences are brought into coarse alignment during the *PN acquisition* stage. Then, a finer synchronization is sought and maintained continuously throughout the *PN tracking* stage. Once proper synchronism is attained, the received waveform is despread by correlating it to the local PN reference waveform, see Figure 17.

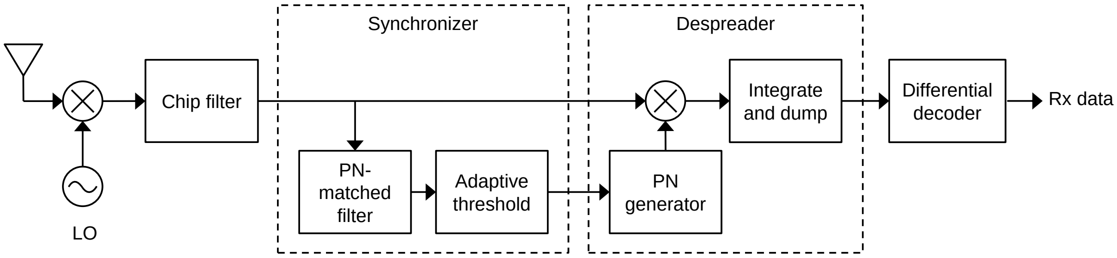


Figure 17: Block diagram of the proposed DS-CDMA receiver for the WSN basestation.

PN acquisition. The primary goal of the PN acquisition process is to detect the incoming PN-spread signal, and coarsely align its PN code with the local replica, usually with a time synchronization error less than a fraction of a chip time. While several acquisition schemes with widely varying speed and and complexity exist, observe that in WSNs, where the network traffic tends to be bursty and the typical packet lengths are short, rapid and robust acquisition becomes especially crucial.

One way to categorize the PN acquisition schemes is based on their detector, either as coherent or non-coherent. Coherent detection is generally more accurate, but requires compensation for frequency and phase offsets between the received carrier signal and the local oscillator, while non-coherent detection does not. Since PN acquisition takes place before carrier synchronization, when the carrier phase is still unknown, the vast majority of DSSS acquisition schemes rely on non-coherent detection.

Another classification viewpoint is the rate at with decisions are made on each PN sequence alignment under test. Low-decision rate detectors generally perform a serial search over a large time region with slow adjustment of the alignment offset between the received waveform and the local PN reference. The acquisition time of such *active* correlators is, therefore, usually prohibitively long for detecting the short packets typical in WSNs. High decision rate detectors, in contrast, perform the same search in parallel, and produce the correlation output at the chip rate or above. Such *passive* correlators, or *PN-matched filters* typically demand substantial amount of computational power, which, according to the assumptions of the proposed protocol, is available at the basestation.

In the proposed passive correlator based acquisition schemes, the matched filter essentially mimics correlation with the PN waveform segment corresponding to the first M spreading chips,

$$\mathrm{MF^q(t)} = \int_{0}^{\mathrm{MT_c}} \mathrm{r}(\tau)\,\mathrm{c^q}(\mathrm{t}+\tau)\mathrm{d}\tau = \int_{-\infty}^{+\infty} \mathrm{r}(\tau)\,\mathrm{h^q}(\mathrm{t}-\tau)\mathrm{d}\tau. \tag{3.7}$$

Thus, the matched filter impulse response is directly associated with the time-reverse of the first M elements of the c_i^q PN sequence,

$$h^q(t) = \begin{cases} c^q(MT_c - t) & 0 \le t \le MT_c \\ 0 & \text{otherwise,} \end{cases} \tag{3.8}$$

where $c^q(t)$ is the PN waveform assigned to the q^{th} sensor node and MT_c is the length of the M-chip segment. As suggested by Figure 18, the magnitude-squared filter output contains the necessary information for coarse alignment. Therefore, the initial PN code and correlation interval synchronization may be established using a threshold logic that triggers the PN generator, see the block diagram in Figure 17.

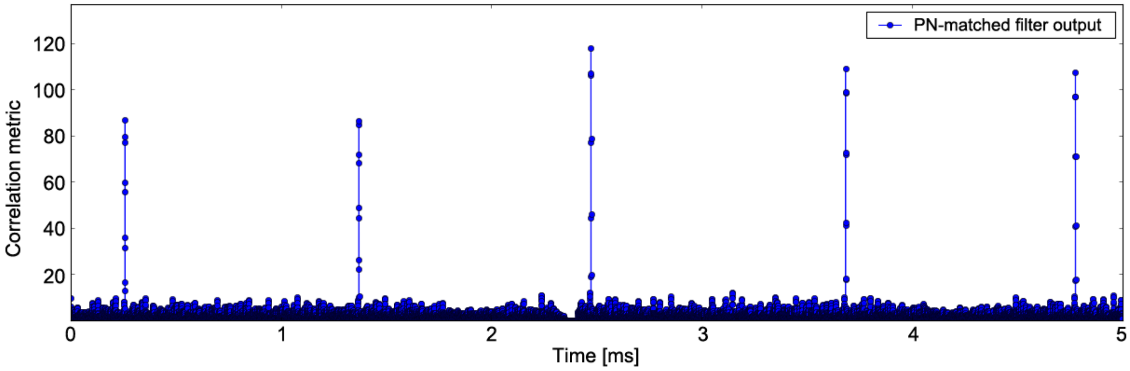


Figure 18: The output of the PN-matched filter in the synchronizer based on measured data and a sampling rate five times the chip rate. Observe the distinct pulses at the start of the spread packet frames and the insensitivity to in-band noise and other network traffic.

PN tracking. The primary function of PN tracking is to maintain fine grained synchronism between the received signal and the PN generator of the receiver. This is ordinarily achieved using active correlators in a delay-lock loop scheme to compensate for alignment errors by slightly advancing or retarding the local PN generator. Such PN tracking is particularly important for communications systems that organize their transmission into long streams of data, where the transmitter and receiver PN generators tend to drift away. In contrast, the short messages in WSNs make drifting negligible. Furthermore, as the packet duration becomes comparable to the integration interval of the active correlator, the tracking loop has limited time to operate. Consequently, robust PN acquisition remains overwhelmingly more important than PN tracking.

Despreading and demodulation. Assuming perfect PN sequence and chip-time synchronization, and given the received $r(t)$ baseband signal, the demodulation of the k^{th} symbol is performed through the

$$I_k(\tau) = \int\limits_{kT_s}^{(k+1)T_s} r(t-\tau)\, c_i^q(t)\, dt \qquad 0 \leq k < K \tag{3.9}$$

correlation over the k^{th} symbol interval, where $I_k(\tau)$ is the integrator output for the k^{th} payload symbol, T_s is the data symbol period, $r(t)$ is the complex chip-filtered signal, $c^q(t)$ is the despreading PN sequence of the q^{th} receiver, τ is the assumed onset of the first payload symbol and K is the number of payload symbols.

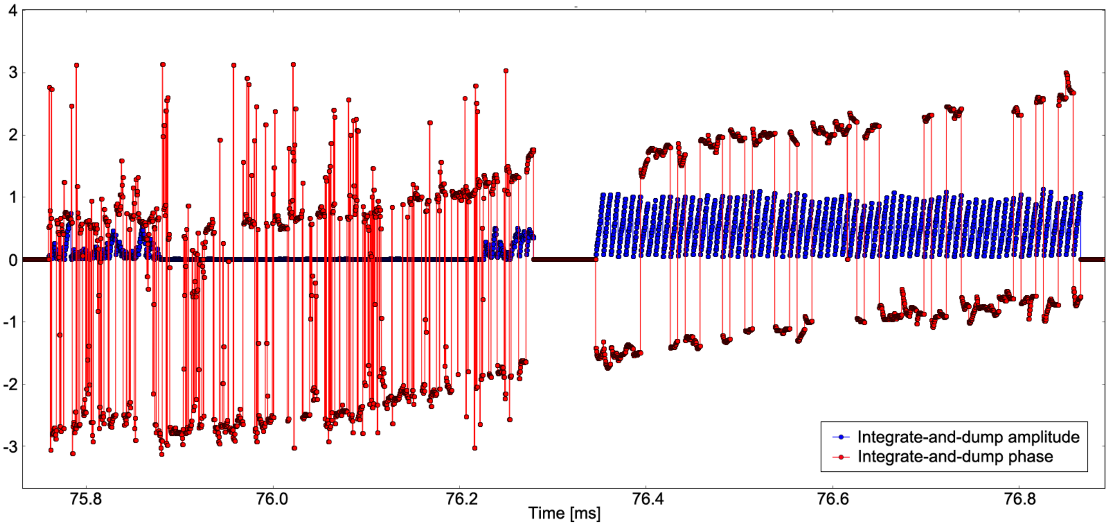


Figure 19: The amplitude (blue) and phase (red) of the integrate and dump block output for an inaccurately (left) and an accurately (right) synchronized packet.

In case of accurate synchronization, the amplitude of $I_k(\tau)$ steadily builds up and its phase remains stable for each data symbol, allowing for proper demodulation, see Figure 19 (right). However, the integrator output becomes disordered and noise-like when the despreading PN sequence is inaccurately synchronized, see Figure 19 (left).

Receiver architectures are generally more sophisticated than those of transmitters due to the synchronization tasks involved. The DSSS approach accompanied with the desire for robust acquisition of short radio packets make this asymmetry even more pronounced. Indeed, PN sequence generation and direct-sequence spectrum spreading are computationally inexpensive, allowing to keep the transmitter simple. In contrast, the receiver has to perform rapid PN acquisition prior to despreading the received signal, which calls for *passive* correlator based acquisition schemes. The matched-filter realization makes the receiver architecture complex by itself. Moreover, in the proposed DS-CDMA scheme the synchronization and despreading functions need to be replicated for each participating WSN node, further increasing the complexity of the basestation.

4 MarmotE SDR Implementation

The proposed DS-CDMA scheme prescribes an asymmetric communication link, with a simple transmitter at the sensor node and a resourceful receiver at the basestation. The required DSSS transmitter complexity is approximately identical to that of the 802.15.4 protocol, for which several COTS radio chips exist. Their rigid silicon implementation, however, generally prohibits the alteration of the transmitted waveforms, hence the customization of the physical layer.

The MarmotE SDR platform provides a means to experiment with novel, full-custom physical layer designs by exploiting the direct access to the baseband signals at the sensor nodes. It has the necessary flexibility and resources to host the proposed spread-spectrum D-BPSK transmitter. The basestation functions require substantially more computing power, therefore, they are implemented in GNU Radio [53] and run on a high-end computer attached to a USRP N210 desktop SDR [50]. The rest of this section elaborates the MarmotE SDR transmitter and GNU Radio receiver designs of the proposed DS-CDMA communication protocol.

4.1 Spreading Code Assignment

The pseudo-noise spreading codes are generated by $L = 11$ long linear-feedback shift registers, which produce m-sequences with a corresponding period of $2^{11} - 1$. There exist 176 distinct m-sequences of this length [91, p. 284], and each node is assigned one of them to spread its transmitted signal with. Certain m-sequence pairs exhibit poor cross-correlation properties, therefore, the actual set of m-sequences are selected such that they minimize the maximum of the integral in (3.4) across all participating PN sequence pairs. A possible set of $N = 4$ distinct m-sequences is shown in Table 12. Note that in case the cross-correlation properties are still unsatisfactory, the PN generator may employ Gold or Kasami codes without significantly increasing the complexity.

Node ID	Generator polynomial	Feedback taps
4	0x416	[11 5 3 2]
5	0x606	[11 10 3 2]
6	0x431	[11 6 5 1]
7	0x415	[11 5 3 1]

Table 12: Assignment of the m-sequence spreading codes to $N = 4$ sensor nodes generated by an $L = 11$ long linear-feedback shift register.

4.2 Frame Format

The packet frame format used by the protocol consists of a PHY header and a PHY payload part, as shown in Table 13. The PHY header contains a fixed 2-byte synchronization pattern used for obtaining symbol, frame and frequency synchronization. The PHY payload comprises four fields, where the source address identifies the sensor node, the sequence number provides a means to ensure frame sequence integrity, payload data carries the actual payload and the 16-bit CRC field allows to check the integrity of the individual frame. Observe that the PN-sequence assigned to the sensor node may also be used for its identification.

PHY header	PHY payload			
Sync pattern	Source address	Sequence number	Payload data	CRC 16
2	1	2	15	2

Table 13: Frame format and field lengths (bytes) used in the experimental protocol.

4.3 Transmitter Design

The DS-CDMA transmitter is associated with a low-complexity architecture that naturally lends itself to the flash FPGA-based SoPC found on the MarmotE SDR platform.

The MAC-level operations, such as generating the PHY payload and scheduling the transmissions, are handled by the microcontroller and the assembled PHY payload is transferred to the FPGA fabric through the AMBA bus interface. The PHY-level functions are implemented in the FPGA fabric, the corresponding Simulink model is shown in Figure 20. Transmission of a PHY frame starts with prepending the header to the payload (not shown), and passing the serialized binary data through the differential encoder. The encoded binary data is then spread by the locally generated PN sequence. As both the encoded data and the generated PN sequence consist of binary symbols, the multiplication reduces to the inexpensive binary XOR operation, which is favorable as the current flash FPGA fabric lacks hardware multipliers. The PN generator is a linear feedback shift register with a configurable generator polynomial that produces a length $2^{11} - 1$ m-sequence at a fixed 2 MHz chip rate. Therefore, since the FPGA fabric operates in a single 20 MHz clock region, the PN generator is enabled only on every 10^{th} clock cycle, and the desired spreading factor is attained by slowing down the binary data stream in a similar fashion. The spread binary symbols are then mapped to BPSK symbols and filtered by a root-raised-cosine (RRC) pulse shaping filter before phase modulating the carrier.

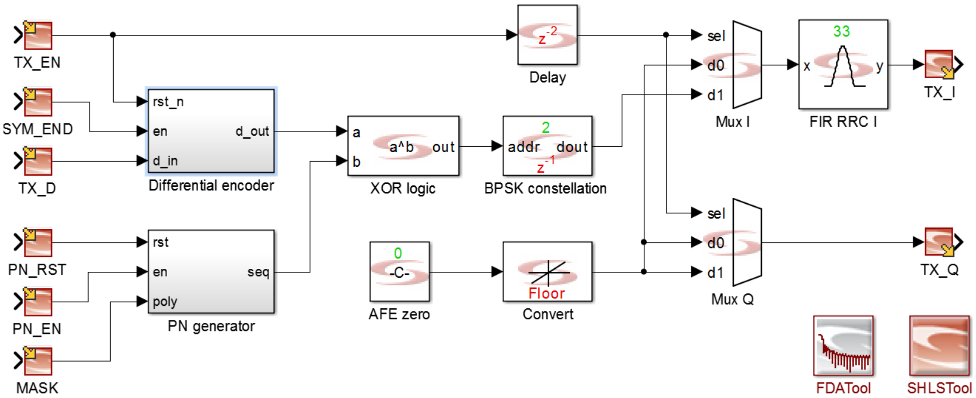


Figure 20: Simulink model of the HDL synthesizable DS-CDMA transmitter.

The corresponding FPGA resource utilization is coarsely summarized in Table 14. The entire DSSS transmitter path, not including the chip filter, takes approximately 14% of the available resources. The PN generator is responsible for 2%, the differential encoder, XOR multiplier and symbol mapper for less than 2% in total. Thus, the AMBA bus interface and the configuration registers contribute most to the 14.1%. The RRC chip filter is a 31-tap FIR filter with 16-bit coefficients.

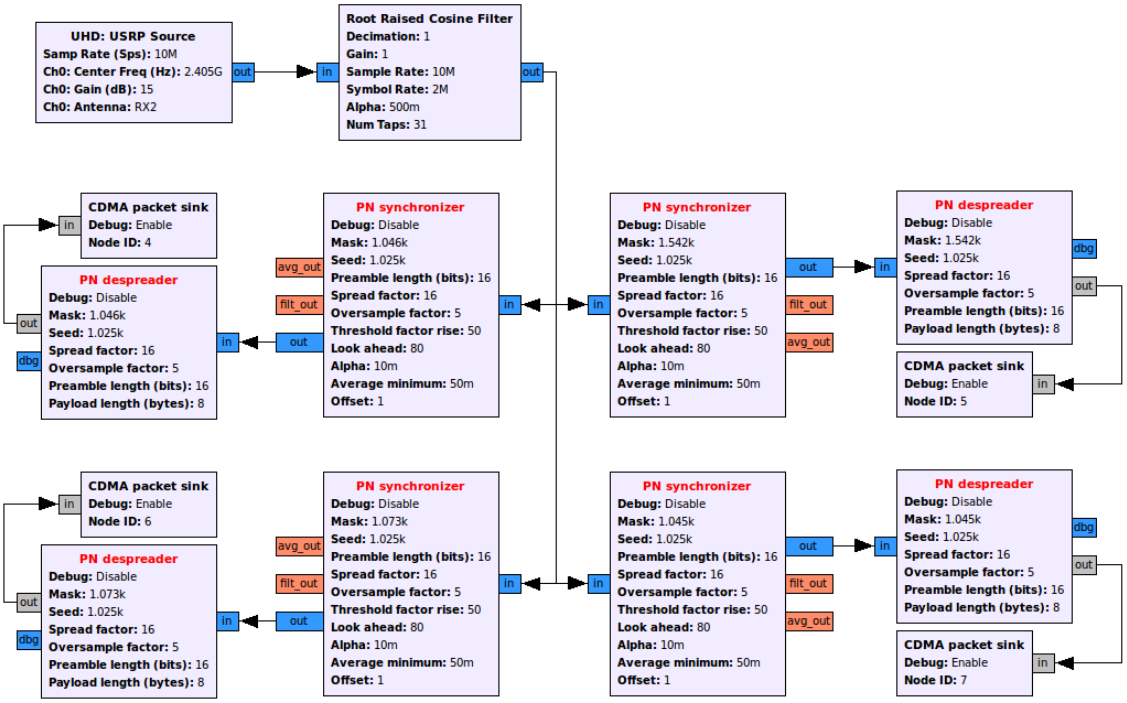
While it consumes 28.4% of the available total FPGA logic resources, the filter complexity may be greatly reduced without significant performance degradation.

Component	Logic cells
DSSS transmitter	1624 (14.1%)
Chip filter	3272 (28.4%)
Total	**4896 (42.5%)**

Table 14: FPGA logic resource utilization of the DSSS transmit path.

4.4 Receiver Design

As the DS-CDMA receiver is of significantly higher complexity than the transmitter, the basestation is implemented in GNU Radio and run on a desktop computer connected to a USRP N210 radio front-end. The block diagram of the receiver associated with one particular PN code sequence is shown in Figure 17. Observe that an N-node WSN requires a unique PN code sequence for each node, therefore, the synchronizer, despreader and differential encoder blocks are replicated N times, as illustrated in Figure 21.



UHD: USRP Source
Samp Rate (Sps): 10M
Ch0: Center Freq (Hz): 2.405G
Ch0: Gain (dB): 15
Ch0: Antenna: RX2

Root Raised Cosine Filter
Decimation: 1
Gain: 1
Sample Rate: 10M
Symbol Rate: 2M
Alpha: 500m
Num Taps: 31

CDMA packet sink
Debug: Enable
Node ID: 4

PN despreader
Debug: Disable
Mask: 1.046k
Seed: 1.025k
Spread factor: 16
Oversample factor: 5
Preamble length (bits): 16
Payload length (bytes): 8

PN synchronizer
Debug: Disable
Mask: 1.046k
Seed: 1.025k
Preamble length (bits): 16
Spread factor: 16
Oversample factor: 5
Threshold factor rise: 50
Look ahead: 80
Alpha: 10m
Average minimum: 50m
Offset: 1

PN synchronizer
Debug: Disable
Mask: 1.542k
Seed: 1.025k
Preamble length (bits): 16
Spread factor: 16
Oversample factor: 5
Threshold factor rise: 50
Look ahead: 80
Alpha: 10m
Average minimum: 50m
Offset: 1

PN despreader
Debug: Disable
Mask: 1.542k
Seed: 1.025k
Spread factor: 16
Oversample factor: 5
Preamble length (bits): 16
Payload length (bytes): 8

CDMA packet sink
Debug: Enable
Node ID: 5

CDMA packet sink
Debug: Enable
Node ID: 6

PN despreader
Debug: Disable
Mask: 1.073k
Seed: 1.025k
Spread factor: 16
Oversample factor: 5
Preamble length (bits): 16
Payload length (bytes): 8

PN synchronizer
Debug: Disable
Mask: 1.073k
Seed: 1.025k
Preamble length (bits): 16
Spread factor: 16
Oversample factor: 5
Threshold factor rise: 50
Look ahead: 80
Alpha: 10m
Average minimum: 50m
Offset: 1

PN synchronizer
Debug: Disable
Mask: 1.045k
Seed: 1.025k
Preamble length (bits): 16
Spread factor: 16
Oversample factor: 5
Threshold factor rise: 50
Look ahead: 80
Alpha: 10m
Average minimum: 50m
Offset: 1

PN despreader
Debug: Disable
Mask: 1.045k
Seed: 1.025k
Spread factor: 16
Oversample factor: 5
Preamble length (bits): 16
Payload length (bytes): 8

CDMA packet sink
Debug: Enable
Node ID: 7

Figure 21: GNU Radio Companion block diagram of the DS-CDMA receiver implemented using the USRP N210 desktop SDR.

The GNU Radio Companion model is shown in Figure 21, where the chip filter is a RRC filter with the same parameters as the pulse shaping filter in the transmitter. Together, the transmit and receive chip filters perform matched filtering to minimize the inter-symbol interference.

The filtered samples are fed to the synchronizer block to detect the packet and establish symbol and frame synchronization. The synchronizer block comprises a PN-matched filter with coefficients corresponding to the first segment of the spreading sequence and a noise-adaptive peak detector. The PN-matched filter calculates the correlation value at each sample, which gives the shortest acquisition time at the expense of significant computational requirement. Observe that the PN-matched filter output exhibits peaks when the corresponding PN spread synchronization pattern is found, but remains insensitive to patterns spread by other PN sequences, see Figure 18. The adaptive threshold logic determines the peak locations and sets the time base for the despreader block to reconstruct the D-BPSK symbols from the spread packet.

The despreader block is based on an active correlator scheme, where a PN generator realizes an m-sequence of $2^{11} - 1$ length through a linear-feedback shift register with parameters shown in Table 13. The PN generator is reset and triggered by the synchronizer block to properly time the onset of the integrate-and-dump block, recovering the D-BPSK symbols according to the integral in (3.9). After differential decoding, the PHY payload is reconstructed and its integrity is checked. Note that while the despreader block contains a PN tracking loop as well, it is disabled because it showed no performance improvement for short packets.

5 Performance Evaluation

The primary goal of the following protocol evaluation is to characterize the PHY layer performance through real-world experiments. For that, the PHY layer protocol implemented using MarmotE SDR nodes and a USRP N210 based basestation, as described in Section 4, is assumed, and two scenarios are considered. First, robustness against interference caused by simultaneous node access is investigated under varying traffic load and number of nodes. Then, the hop-distance dependency is considered with the transmit power fixed. In both cases, the average packet reception ratio (PRR) serves as the ultimate performance metric. Furthermore, control measurements conducted with commonly used 802.15.4 radio based WSN nodes in the same arrangement serve as a reference.

5.1 Simultaneous Access Measurements

5.1.1 Measurement Setup

For the simultaneous access experiment, the MarmotE SDR nodes were configured as DSSS transmitters and the basestation was represented by a USRP N210 connected to a desktop computer. The setup was deployed in an office environment and the basestation registered the packet reception ratios (PRR) using different number of nodes, varying traffic load and different spreading factors. There were three measurement scenarios involving 1, 2 and 4 nodes and one basestation, respectively, with the four-node setup shown in Figure 22.

In each scenario, the basestation calculated the *average packet reception ratio* based on the number of correctly received ones of the 1000 transmitted packets in total. To control the traffic load, the messages were transmitted with a τ_{avg} average interval, predefined for each experiment, and having ±20% jitter. That is, the time delay between two consecutive packet transmissions was calculated as $\tau_{\text{avg}}(1 + \nu)$, where the $\nu \sim \mathcal{U}\,[\text{-}0.2, 0.2]$ jitter was distributed uniformly. With these, the *average transmission duty cycle* was defined as the ratio of the message duration and τ_{avg}.

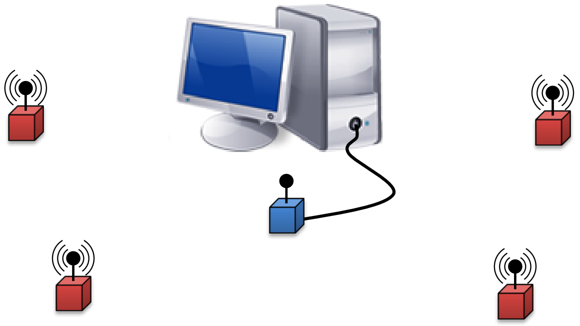


Figure 22: Simultaneous access measurement setup with four MarmotE SDR nodes (red) acting as transmitters, and a USRP N210 (blue) connected to a desktop computer as the basestation.

The packets were constructed according to the frame format described in Section 4.2, where each comprised 22 bytes of PHY header and payload in total. The radios were tuned to 2.405 GHz carrier frequency and the chip rate was fixed at 2 MHz for all cases. Therefore, the spreading factor values 8, 16 and 32 reduced the data rate to 250 kbps, 125 kbps and 62.5 kbps, respectively. In all cases, the transmit power of the nodes was adjusted manually based on the perceived packet error rates, until the received power levels were approximately equal at the basestation. Given a communication link from the basestation to the sensor nodes, a similar closed-loop power-control scheme may be employed in an automated fashion.

The control measurements were executed in the same indoor environment using up to four CC2420 equipped TelosB WSN nodes as transmitters, and a CC2531 evaluation module as basestation. The TelosB nodes hosted a TinyOS application, with the clear channel assessment (CCA) option turned on or off, for periodically transmitting packets consisting of 22 bytes of synchronization header, PHY header and payload in total. Meanwhile, the basestation relied on a vendor provided packet sniffer software. Observe that both the CC2420 and the CC2531 radio chips are built upon the 802.15.4 communications protocol standard.

5.1.2 Evaluation

The first experiment relied on the basestation and a single transmitter to create a baseline measurement with interference coming from the office environment only. The results plotted in Figure 23(a) show that a reliable communication link was established for both the control and the DSSS measurements, independently of the CCA option or spreading factor used, respectively. The packet delivery ratio was 100% regardless of the traffic load, as ordinarily expected for a contention-free and clear channel. Observe that the control measurements were preformed starting at an average transmission duty cycle below 25%. This is due to the long acquisition time, over 3 ms, of the radio interface with TinyOS on the TelosB platform when scheduling individual packets. In comparison, the DSSS physical layer on the MarmotE SDR platform allows for precise and fine grained timing of packet transmissions, which highlights the flexibility of having direct access to the baseband signals.

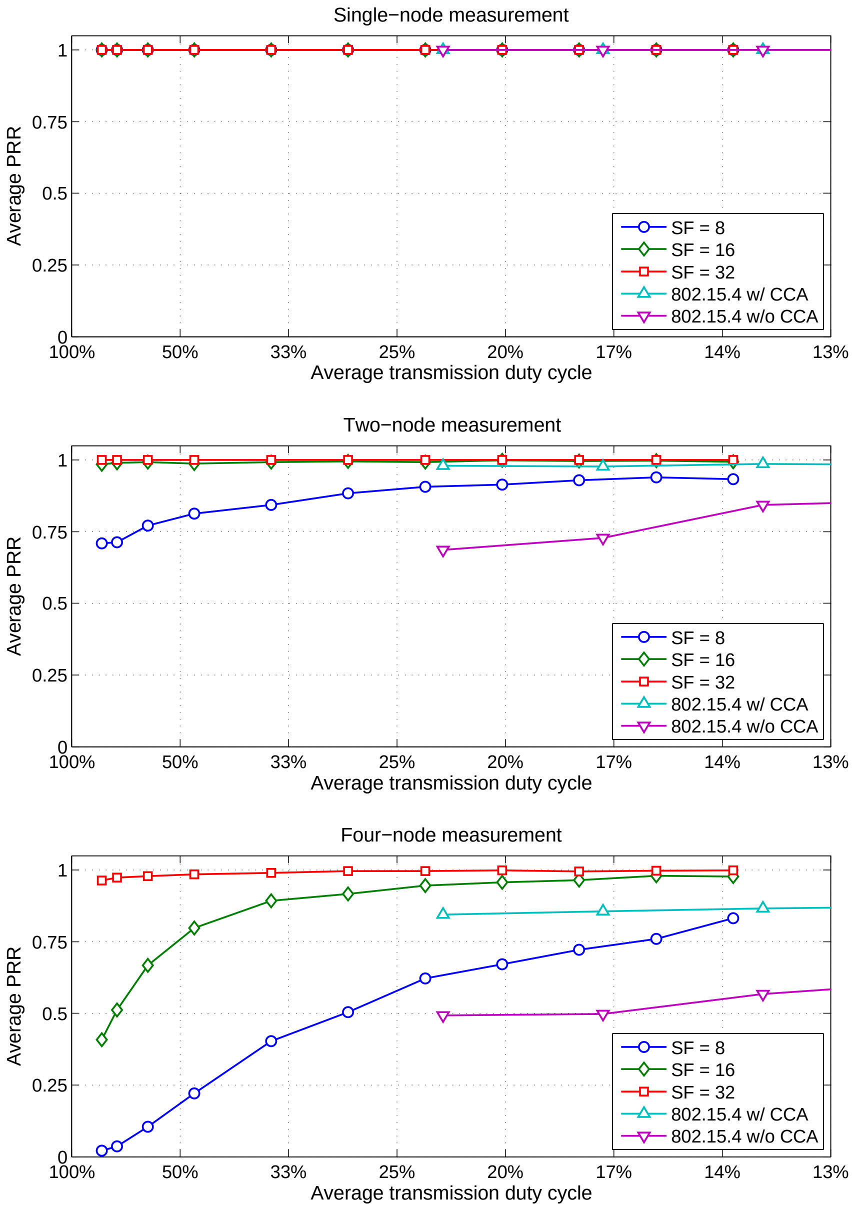


Figure 23: Measured packet reception ratio for the single-node (top), two-node (center) and four-node (bottom) simultaneous access experiments under various traffic loads for the proposed DSSS protocol on the MarmotE SDR with spreading factors of 8, 16 and 32, and the 802.15.4 protocol used on TelosB with CCA on and off.

Adding a second transmitter node to the experiment introduced multiple access interference to the channel, which clearly impacted the packet reception ratio for the control measurement without CCA and for the low spreading factor case, as suggested by Figure 23(b). Under heavy traffic conditions, the average packet delivery ratio dropped below 75% for the control measurement with CCA turned off and for the DS-CDMA protocol with spreading factor of 8. However, it remained above 97% for all the other cases. Examination of the received data showed that packet losses were equally probable due to missed detection of the synchronization header and to the corruption of the payload data.

Incrementing the number of nodes to four, further increased the channel interference, creating a more realistic multi-node scenario. The average PRR curves depicted in Figure 23(c) show that approximately 85% and 50% of the packets got delivered in the control measurements with CCA on and off, respectively. Relying on the DS-CDMA protocol with a spreading factor of 8, only 2% of the packets were delivered on average under high traffic, approximately 50% were received under moderate traffic, and more than 80% arrived in the low-traffic channel. Increasing the spreading factor to 16 improved the average packet reception ratio to over 90% in moderate network traffic, but it fell down to 41% under heavy loads. Meanwhile, setting the spreading factor to 32, the protocol delivered over 96% of the packets, even in a highly loaded channel.

In summary, the experiments showed that both the proposed DS-CDMA and the 802.15.4 communication protocols performed reliably when there was no contention in the channel. A spreading factor of 8, the same as effectively used by 802.15.4, already offered reasonable protection against external interferences, such as the ongoing WLAN communication. However, as the number of nodes increased, such a low degree of spreading offered little protection against simultaneous transmission even under low traffic loads. Therefore, a truly collision-free DS-CDMA communication calls for reasonable spreading factors, which still remains a design parameter primarily determined by the size of the network, the expected network traffic, the power budget and the hop-distance discussed in Section 5.2. In the presented experimental setup the desired factor of spectrum spreading was attained by using a fixed chip rate and reducing the effective data rate, hence implicitly increasing the unit energy per bit. Alternatively, the data rate could be fixed and the chip rate (bandwidth) increased to achieve the same goal, without increasing the transmission duration.

5.2 Hop-Distance Measurements

5.2.1 Measurement Setup

The purpose of the hop-distance experiment was to characterize the DSSS communication link performance as a function of the transmitter-receiver distance and spreading factor, while keeping the transmit power fixed. Throughout the evaluation, a MarmotE SDR node was configured as the DSSS transmitter node and a USRP N210 with a laptop computer served as the receiver basestation, both employing the PHY layer described in Section 3. The control measurements relied on IRIS motes [95], equipped with 802.15.4 based CC2420 radio chips, for both the transmitter node and the basestation roles. For each measurement, the basestation was elevated to 55 cm height, while the transmitter node was placed directly onto the ground in an outdoor environment, see Figure 24. This arrangement effectively represents a near-ground scenario with a propagation environment of similar characteristics as described in [96]. The average packet reception ratio was then measured based on 1000 transmitted messages for transmitter-receiver distances ranging from 15 m to 85 m.

The radios were tuned to the 2405 MHz carrier frequency and operated at a nominal transmit power of 0 dBm. The chip rate was fixed at a constant 2 MHz, and the spreading factor was varied between 8, 16 and 32, implicitly reducing the transmission rate to 250 kbps, 125 kbps and 62.5 kbps, respectively. The DSSS messages used the frame format described in Table 13, and the 802.15.4 packets were trimmed to the same length.

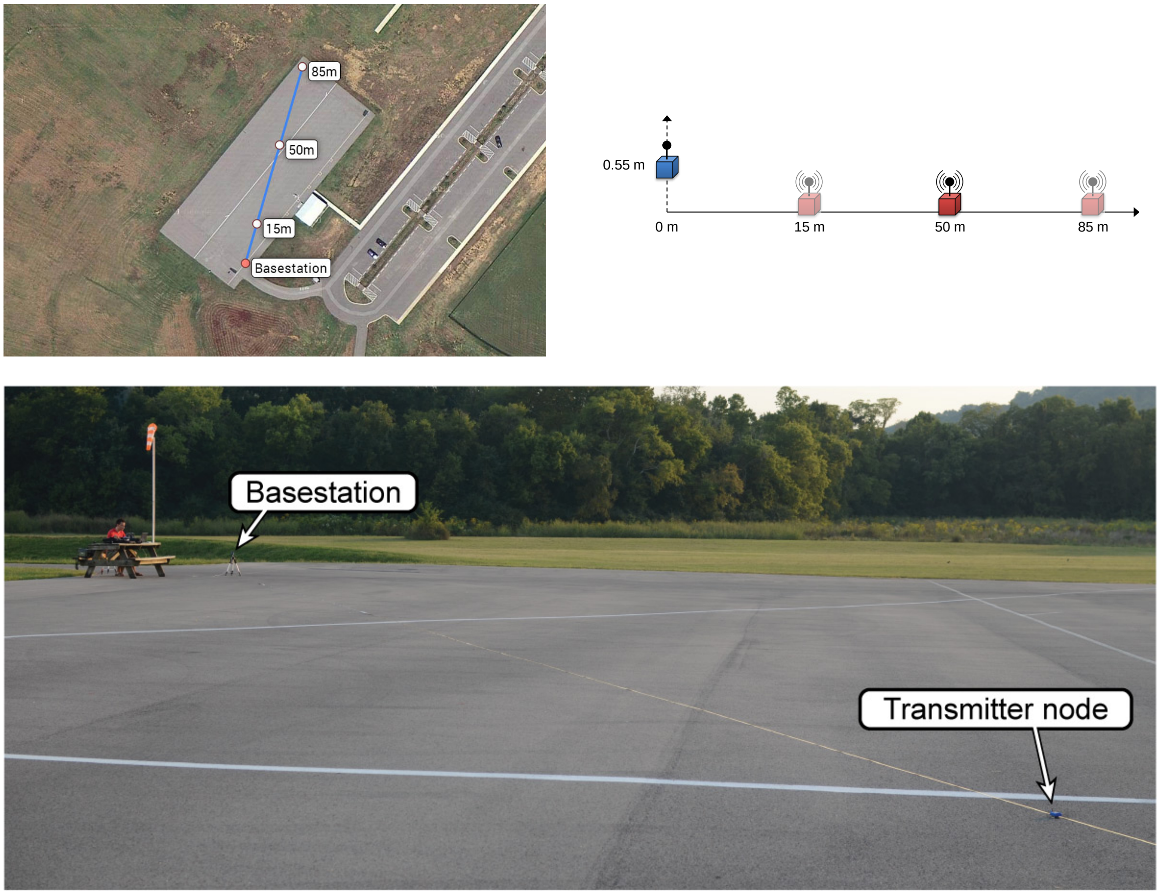


Figure 24: Map (top-left), schematic arrangement (top-right) and photo (bottom) of the hop-distance measurement setup. Map data © 2014 Google.

5.2.2 Evaluation

The observed average packet reception ratios plotted against the hop-distance in Figure 25 show that the communication link was reliable for all measurement setups with a transmitter-receiver distance less than 23 m. The performance of the 802.15.4 communication, used as the control measurement, started to degrade rapidly at approximately 25 m, and cut off completely above 35 m. In contrast, the proposed MarmotE SDR implemented DSSS protocol performed reliably up to 47 m and kept the average PRR above 90% even at 57 m for all spreading factors used. Above 55 m, the PRR started to drop quickly for the spreading factor of only 8, albeit at a slower rate than the control measurement, and remained below 25% for distances over 75 m. When spread by a factor of 16, the threshold of the performance breakdown was observed at approximately 67 m. Above that, the PRR started to show a gradually falling trend, though exhibited high fluctuations around 74.7 m and 80.7 m. Increasing the spreading factor to 32, the DSSS protocol kept the PRR above 90% for up to 82 m hop-distance.

The above experiments showed that the maximum hop-distance, over which reliable communication could be established, varies widely despite the fixed nominal radio transmit power. The estimated breakdown-thresholds for the outdoor near-ground scenario were 48 m, 67 m and 82 m

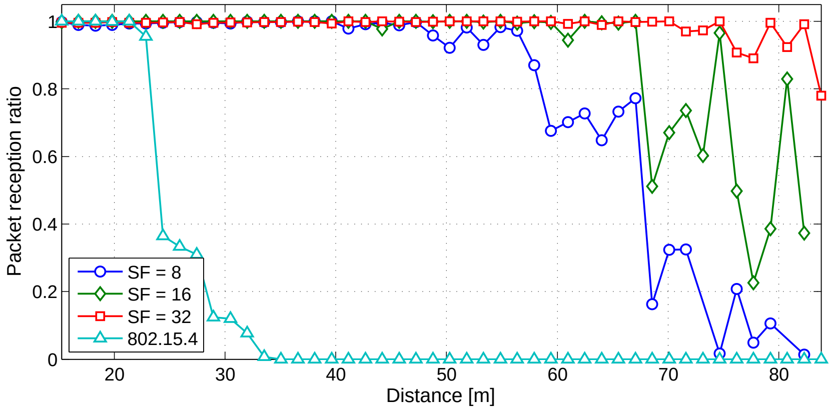


Figure 25: Measured packet reception ratios as the function of hop-distance for the proposed DSSS protocol on MarmotE SDR with spreading factors of 8, 16 and 32, and the 802.15.4 protocol used on the IRIS node.

for the proposed DSSS protocol with spreading factors of 8, 16 and 32, respectively, while 25 m for the control setup. On one hand, this confirms that the effective communication range of transmit power-limited WSN nodes may be extended through spectrum spreading. In the proposed DSSS protocol, it was achieved by keeping the chip rate constant and implicitly increasing the message duration, consequently, at the price of a proportionally increased energy per transmitted bit ratio.

Comparison with the control measurement further suggests that the actually attained performance is heavily dependent on both the propagation environment and several implementation details. Indeed, while the 802.15.4 protocol also employs DSSS, and the transmit power of the IRIS mote and the MarmotE SDR were tuned to the same level, significant performance mismatch was observed due to possible differences in the observed ground reflection, antenna gain and overall receiver sensitivity.

6 Conclusion

The closed architecture of the integrated COTS radio chips used in WSNs generally restricts experimentation with communication protocols to the MAC layer and above. The MarmotE SDR platform intends to overcome such limitations through its flexibly configurable architecture and by granting direct access to the baseband radio signals, in order to foster the development of novel PHY layers for WSN communication protocols. Thus, the purpose of the design and evaluation of the spread-spectrum communication protocol in this chapter was twofold.

First, it emphasized that the ability to define custom PHY layer waveforms has indeed great potential in WSN communications research. The proposed DS-CDMA scheme was specifically tailored to data gathering WSNs that are characterized by short packet lengths and bursty traffic. The protocol enabled the sensor nodes to simultaneously transmit their PN-spread messages to the basestation with arbitrarily reduced collision rates. It also allowed to extend the maximum attainable hop-distance for a given transmit power level, therefore, to increase the number of nodes that are able to reach the basestation in a single hop. In both cases, the trade-off was the increased energy per transmitted bits. Furthermore, the protocol leveraged the asymmetry in the PHY level waveform processing requirements to keep the transmitter complexity of the sensor nodes low, and shift the computational burden to the resourceful basestation. Note that for the same reason, the proposed PHY layer is not directly applicable to inter-node communication in a multi-hop scheme.

Second, it demonstrated that the MarmotE SDR platform is an excellent vehicle to implement such protocols and to experimentally evaluate their performance. Its reconfigurable logic resources were abundant for the task, and the accompanied high-level synthesis based workflow allowed both to improve the simulation fidelity of the PHY layer transmitter components, and to transfer their functions to the FPGA fabric with minimal implementation effort. Furthermore, the battery-enabled operation of MarmotE SDR then simplified deployment for both the indoor measurements, evaluating the multiple access capability of the protocol, and especially for the outdoor scenario, investigating the attainable maximum hop-distance for nodes with limited transmit power. Such evaluation of the proposed DS-CDMA communication protocol would have been practically infeasible using existing COTS radio chip based WSN nodes.

CHAPTER IV

SENSOR NODE LOCALIZATION

1 Introduction

Wireless sensor networks collect information about the physical world, in which the sensor locations give context to the measured data. Accurate characterization of the spatial relationship between the sensor nodes is, therefore, crucial for most applications including environmental [13] and habitat monitoring [11][12], agriculture [14], asset tracking, industrial monitoring [15], shooter localization [16] and structural health monitoring [17]. In many cases, the sensor node locations can be determined as part of the deployment process by mounting them at predefined positions or by performing their one-time location estimation using external tools, such as a measure tape, a laser rangefinder or GPS. However, the deployment of large scale infrastructureless ad-hoc sensor networks often makes this approach either impractical or even prohibitive, and call for automated node localization algorithms that meet the particular WSN application requirements. The wide variation of the requirements for node cost, size and energy consumption, localization accuracy and the availability of infrastructure, makes one particular solution unlikely to fit all applications. Rather, a set of solutions with common characteristics, but of different complexity is expected to emerge for the applications to choose from.

In this chapter, Section 2 describes the typical stages of the sensor node localization process and reviews the existing solutions. Section 3 discusses the underlying principles of the radio phase-based approaches proposed for ad-hoc WSNs. Most importantly, it presents a generalized yet simple description for phase measurement in order to point out the differences and similarities between the already available and the proposed methods while avoiding the unnecessary technical details. Still focusing on the measurement stage of localization, Sections 3 and 4 describe two novel methods for phase and distance estimation, respectively. The details of the corresponding MarmotE SDR implementation are covered in Section 5, while the experimental results are evaluated in Section 6. Finally, Section 7 concludes this chapter.

2 Background

2.1 Localization in Sensor Networks

Despite the wide variety of approaches, the process of sensor node localization can naturally be divided into three distinct stages. The *calibration* stage sets up the sensor nodes to compensate for distortions due to manufacturing variation of the individual devices and gradual changes in the environmental conditions. Following, the *measurement* stage performs measurements and parameter estimation to obtain information on the spatial arrangement of the sensor nodes, such as relative location or angular separation [97]. Finally, the *localization* stage fuses the measurement results to calculate the location of the individual sensor nodes.

2.1.1 Calibration

The calibration stage ensures that the output of a given sensor matches a reference output within a specified accuracy. Sensors with different modalities call for different calibration strategies to compensate for manufacturing variations or environmental conditions. To account for the significant manufacturing differences of low-cost devices, traditional sensors are usually *calibrated individually* in a controlled environment before deployment. Changes in the environmental conditions, such as temperature and humidity, affect the propagation speed of acoustic waves and frequency of the local oscillator and deteriorate the measurement results if not compensated for. To reduce localization-related measurement distortions after deployment, *micro-* and *macro calibration* have been proposed for sensor networks.

Individual calibration. The general approach to improve the accuracy of actuators and sensors is to calibrate them in a controlled environment on an individual basis. While this is also applicable to sensor node components used for localization measurements, their low-cost requirement implies significant residual inaccuracy. Furthermore, calibration has to be performed regularly to accommodate changes in the operating environment, therefore, individual calibration by itself is often inadequate for sensor network localization measurements.

Micro calibration. Measurements for sensor network localization are generally performed between an actuator (transmitter) and sensor (receiver) pair, and their accuracy is affected by the distortions present in both sides. The aim of micro calibration is to separate and estimate the error contributions of the transmitter and the receiver to improve the accuracy of pairwise ranging measurements. The SpotON [98] system operates on the received signal strength (RSS) output of the radio to estimate the distance between node pairs. During calibration SpotON dedicates one transmitter as reference and calibrates all receivers at known distances. Then, it selects one of the receivers and sequentially calibrates the rest of transmitters relative to it. A similar approach is proposed in [99], where each receiver is calibrated to the mean of all transmitters without calibrating the latter at all. The Cricket system [100] also performs micro calibration based on known distances between a set of nodes to compensate for receiver and transmitter introduced errors, as well as for variations in the propagation speed of acoustic signals. Therefore, the above micro calibration techniques rely on infrastructure information in the form of a priori known distances between a set of nodes.

Macro calibration. Several sensor network deployments make pairwise micro calibration either impractical or unattainable due to the lack of infrastructural support. An alternative approach for ad-hoc sensor networks is to perform the calibration at the system level. Calamari [99] treats the calibration of RSS and acoustic sound based measurments as a joint parameter estimation problem and assign correction parameters to sensor nodes based on a system-level optimization criteria, rather than based on a pairwise calibration.

2.1.2 Measurement

The measurement stage of sensor node self-locaization estimates signal parameters that are associated with the spatial relationship between a set of nodes. Depending on the particular signal modality and processing technique used, the measurement output conveys information on the proximity, the relative distance or the relative direction of a node with respect to other nodes or the infrastructure.

Range-free. A simple approach for node localization is to detect the presence of a node by the infrastructure and associate its location with the corresponding region. The Active Badge system [101] uses infrared transmitter tags and deployed infrared detectors in different rooms of a building, while LANDMARC [102] relies on the detection of RFID tags. An alternative approach for multi-hop sensor networks with reduced infrastructure support is to detect proximity of neighbor nodes based on the communication range [103][104] and exploit network connectivity at a later stage. In general, range-free localization measurements are simple, but provide a very coarse grained distance estimate.

Distance-based. A straightforward way to characterize the spatial arrangement of a set of nodes is to estimate their pairwise relative distances. Therefore, several distance-based measurements have been proposed mainly using the received signal strength, time-of-flight and time-difference-of-arrival of radio or acoustic signals.

Signal Strength. RSS-based measurements estimate the distance between a transmitter-receiver pair based on an assumed attenuation rate of the propagating radio signal. Most RSS approaches utilize a noisy signal model with exponential power decay over distance, where the additive zero mean Gaussian noise and the value of the path loss exponent characterize the propagation environment [97]. The SpotON [98] and LANDMARC [102] localization systems rely solely on RSS-derived distances, while CALAMARI [99] and AHLoS [105] use them as a secondary measurement. RSS-profiling is an alternative approach, which constructs an RSS map with respect to the infrastructure nodes either offline, during the calibration stage [106], or online, using deployed reference devices. In either case, the RSS readings are obtained in active mode in a similar fashion for comparison with the map. Utilizing the RSS is an attractive way to estimate the distances between neighboring nodes as it is simple, requires no additional hardware and is more accurate than proximity-based techniques. However, its performance is usually inferior compared to other distance and directionality based methods [97].

Time of Flight. The time-of-flight (ToF) measurements leverage the finite propagation speed of radio or acoustic signals to deduce distance information. The speed of radio waves is approximately 300×10^6 m/s, while that of the sound in air is only around 330 m/s. Therefore, the ToF of radio signals is negligible compared to that of acoustic signals, making radio message coordinated one-way ultrasound ToF measurements the one of the most compelling approaches in sensor networks. In Active Bats [107], the infrastructure polls the nodes via radio messages to request an ultrasound pulse and measure its ToF and, inherently, the distance to the node. Similarly, the Cricket [100] and CALAMARI [99] systems perform radio synchronized acoustic ToF measurements between the sensor nodes to calculate their distance. One-way ToF measurements require tightly synchronized transmitter and receiver clocks with respect to the signal propagation time, however, such a precise time synchronization is usually unattainable in sensor networks for radio signals. Therefore, ultra wideband (UWB) radio signal based distance measurements [108] primarily rely on *roundtrip* ToF calculation where the same clock is used to register the time when the signal was transmitted and when it was returned. While this inherently eliminates the need for precise clock synchronization between the two nodes, the stable high-frequency clock source required to process the several MHz bandwidth UWB signals is usually still prohibitive on the sensor nodes.

Time (Difference) of Arrival. Time-of-arrival (ToA) and time-difference-of-arrival (TDoA) techniques estimate distances from multiple transmitters or to multiple receivers simultaneously based on the finite propagation speed of a signal. The traditional GPS [92] is probably the most well known localization system, which estimates ranges from multiple transmitters based on the ToA of radio signals. During the ranging measurement, the precisely time synchronized satellites continuously transmit messages that contains ephemeris data and the start time of the transmission. The receiver measures the precise ToA of the radio messages from at least four satellite to estimate the pseudo-ranges and compensate for its clock error. In contrast, the distance measurement in the Ubisense [109] location system involves multiple receivers and a single transmitter. The infrastructure comprises a set of precisely time synchronized, high-performance receiver nodes deployed at known locations in a building. Upon the request from the infrastructure, the node to be localized emits an UWB radio pulse and the receivers register the ToA, and inherently the TDoA. The RIPS [9][8] obtains distance-related information through implicit TDoA measurements of radio signals. In a basic RIPS measurement, two nodes transmit unmodulated sinusoid carriers at slightly different frequencies while two receivers measure the location dependent phase offset of the two signals. The difference of the two phase offsets corresponds to an ambiguous difference of two TDoAs, where the ambiguity is resolved by repeating the measurement at substantially different carrier frequencies.

Directionality-based. A third technique to determine the spatial relationship of sensor nodes is to measure direction or angle of arrival (AoA) of a signal with respect to a set of infrastructure nodes. The SpinLoc [110] is analogous to GPS, as the infrastructure nodes generate the reference signal and multiple receiver nodes can determine their location simultaneously. The infrastructure nodes physically rotate a radio transmitter to generate a Doppler-shifted signal. The periodic Doppler-shift is measured at a reference node and the node at unknown location to determine the angular separation between the two with respect to the spinning transmitter. The quasi-Doppler approach [111] is based on the same scheme, but with the transmitter periodically switching between a set of circularly-arranged antennas to imitate the spinning of a single one. Therefore, it requires multiple antennas at the infrastructure nodes, but not their mechanical movement. Similarly, the directionality-based measurement in the Ubisense [109] system uses multiple antenna array equipped receivers to determine the AoA of the node emitted UWB pulse. Finally, TripLoc [112] measures the direction from infrastructure nodes to sensor nodes based on radio interferometric phase measurements. The basic TipLoc AoA measurement is similar to the RIPS measurement with a special arrangement of the node-quartet: one receiver and two transmitters are placed within half-wavelength distance from each other to from an infrastructure node. The measured relative phase offset in this setup is then directly related to the direction towards the second receiver, the node with unknown position.

2.1.3 Localization

The localization stage uses the range-free, distance-based or directionality-based measurement results to build a coherent map of the sensor node positions. As the measurement outputs are generally estimates corrupted by measurement noise, optimization methods are used to accurately determine the node locations. Based on the nature of the measurement and the particular optimization algorithm used, the computational complexity of the localization may vary significantly. Such computations, however, may be performed in a centralized [101][107][9] or distributed [92][112] manner.

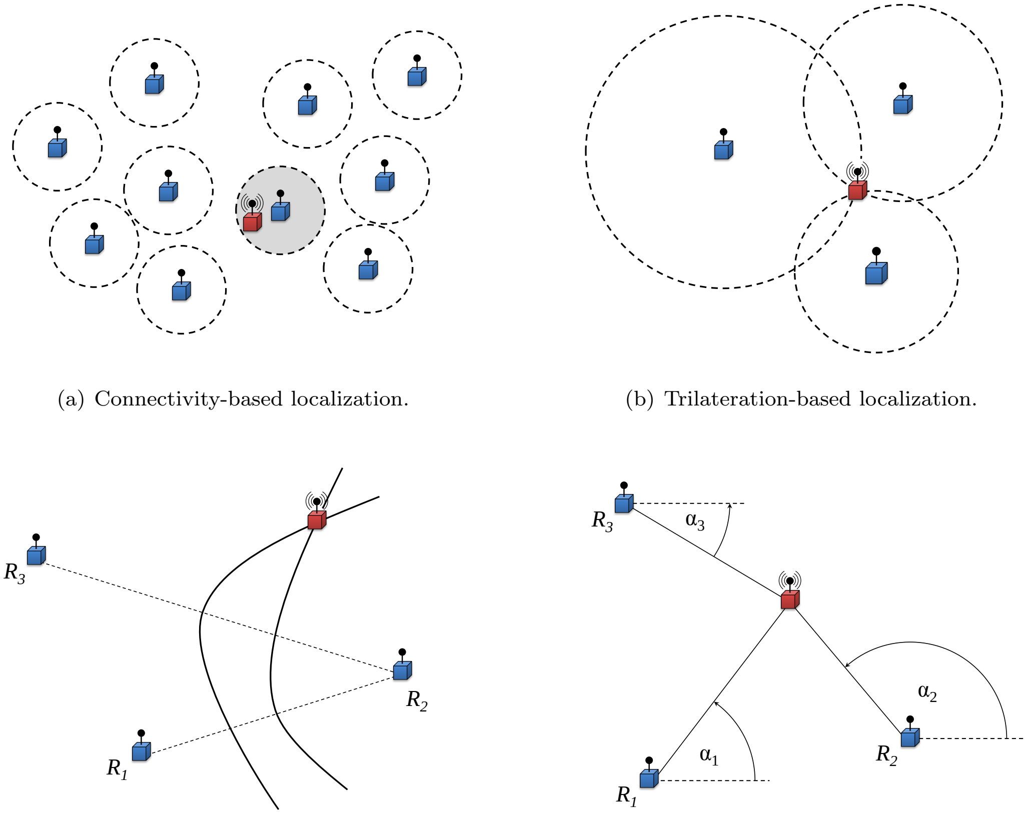
Connectivity. One of the simplest approaches to node localization is to use range-free measurements to determine the proximity of a sensor node to either infrastructure nodes [101][102] or other sensor nodes [104][103] and associate its location with the corresponding region, see Figure 26(a). The greatest advantage of connectivity-based localization algorithms is their simplicity, which comes at the cost of coarse grained location estimates, heavily affected by the infrastructure and sensor node density. Therefore, connectivity-based localization is especially attractive for large node count, dense sensor networks.

Lateration. Lateration is a localization technique that operates on distance-based measurements to resolve the relative or absolute node positions. In sensor networks, *trilateration* is used most commonly [100][99][105][107], however, *multilateration* is also often employed [109][9][8].

Trilateration. Trilateration determines the position of a sensor node based on its pairwise *distance* to reference nodes with known locations. If the distance estimates are perfectly accurate, then the two-dimensional trilateration requires three reference nodes to unambiguously locate a sensor node by finding the intersection of three circles, as illustrated in Figure 26(b). However, in sensor networks, the distances are usually obtained from noisy ToF measurements performed using limited accuracy sensors and time synchronization. As a consequence of imperfect distance measurements, the circles no longer intersect in a single point, and estimating the node location becomes an optimization problem [97]. Trilateration is used in the Cricket [100], CALAMARI [99], AHLoS [105] location systems, as well as its three dimensional version in GPS [92].

Multilateration. Multilateration is an alternative localization scheme based on the measurement of *distance difference* between multiple nodes. The accurately measured pairwise distance difference defines a hyperbolic curve in the two-dimensional case, as shown in Figure 26(c). Assuming perfect measurements with at least three reference nodes, the intersection of the hyberbolas pinpoint the node location. In sensor networks, the distance difference estimates are usually based on limited accuracy TDoA measurements, therefore, the hyperbolas rarely intersect at a single point, and finding the node location transforms into an optimization problem. Multilateration is often employed in UWB radio location systems [108][109] and implicitly in the RIPS [9][8].

Angulation. Angulation is a directionality based localization technique based on AoA measurements from multiple reference locations. In case of ideal measurements, the AoA from two infrastructure nodes is sufficient to determine the node location in two dimensions, as depicted in Figure 26(d). However, the AoA measurements are typically corrupted by measurement noise and the location estimate becomes inaccurate, especially at large distances. Furthermore, the AoA lines from more than two reference nodes cease to intersect at a single point, and similarly to the lateration case, determining the node location turns into an optimization problem. Angulation based localization is performed in the Ubisense [109], TripLoc [112] and SpinLoc [110] systems.



(a) Connectivity-based localization.

(b) Trilateration-based localization.

(c) Multilateration-based localization.

(d) Angulation-based localization.

Figure 26: Comparison of the different sensor node localization techniques.

2.2 Existing Localization Systems

Sensor node localization faces substantially different challenges indoors and outdoors. The small, closed nature of indoor environments are subject to reverberation and multipath propagation, which make ranging measurements challenging. However, they also offer tremendous opportunities for infrastructure-dependent techniques. Outdoor sensor localization systems, on the other hand, usually cover larger areas and the ranging measurements have to accommodate to the larger distances. Localization in these two environments requires different approaches, therefore, existing indoor and outdoor localization systems are discussed separately.

2.2.1 Indoor localization systems

Active Badge. The Active Badge [101] location system is designed to locate people in an office environment. The tracked person wears a badge that periodically emits a unique identifier using an infrared transmitter. Sensors deployed at known positions in the building detect the transmitted signal and notify a central computer about the presence of the person in the area. Among the major drawbacks of this approach are the heavy reliance on the centralized infrastructure and the line-of-sight signal detection.

Active Bats. The Active Bats [107] is often considered as the successor of Active Badge in tracking people indoors. Similarly, the tracked person carries a transmitter tag with him, however, it is polled by a radio message to emit an ultrasonic pulse. The location of the tag is then estimated based on the pulse time-of-flight from the tag to the ceiling-mounted ultrasound receivers. The Active Bats system works without line-of-sight signals and achieves more precise localization at the cost of a tightly controlled and centralized infrastructure.

Cricket. The Cricket [100] also uses a combination of radio messages and ultrasound pulses to locate nodes attached to people or objects. Similarly to Active Bats, it measures the one-way time-of-flight of ultrasonic pulses, however, in the opposite direction. The fixed location infrastructure nodes transmit radio messages along with ultrasonic pulses to allow listener nodes to determine their own location. Therefore, the Cricket also relies on infrastructure, but in a decentralized manner.

RADAR. The RADAR [106] is a radio frequency indoor localization system that combats the complex multipath environment of office areas by relying on RSS-profiling in a 802.11 wireless network. In the setup phase, the RSS is measured between multiple infrastructure nodes at fixed positions and a transmitter moved to multiple known locations to build an empirical offline map of the radio environment. During normal operation, the system takes similar RSS measurements from the transmitters and fits them on the offline stored map to estimate the current transmitter location. The main advantage of RADAR is its low cost as it uses an already existing infrastructure. However, it is reported to achieve lower overall accuracy than either Cricket or Active Bats [102].

LANDMARC. The LANDMARC [102] location sensing system explores another RSS-profiling approach based on active radio frequency identification (RFID) tags. The active RFID tags periodically transmit their unique identifier in a building with deployed RFID readers at known locations. An RFID reader reports the presence of a tag only if the RSS from that tag is above a threshold. Thus, the RFID readers scan at 8 different threshold levels to obtain coarse RSS measurements and locate the tag. The LANDMARC has several drawbacks compared to RADAR, including the low-resoultion (binary) output of RFID readers obtained through a long scanning time and the strong variation in the behavior of the tags.

Ubisense. The Ubisense [109] is a commercial location system that utilizes UWB radio signals to estimate the position of tags in multipath-rich indoor environments. Similarly to Active Bats, a tag is polled through a radio message to emit an UWB pulse on its separate radio interface. The UWB pulse is then processed by tightly synchronized infrastructure sensors to first obtain TDoA and AoA estimates and then to locate the tag. Ubisense achieves sub-foot accuracy and 10 ms order update rate through high sampling rates and precise time synchronization.

SpinLoc. The SpinLoc [110] system exploits the Doppler effect in radio signals to determine the position of sensor nodes indoors. Infrastructure nodes at known locations physically rotate a radio transmitter to generate the Doppler-shifted signal. The periodic Doppler-shift is measured at a reference node and the target node to determine their angular separation with respect to the spinning transmitter. SpinLoc repeats the same process for several spinning infrastructure nodes and uses triangulation to estimate the target node position. Based on the reports SpinLoc achieved 70 cm or better accuracy in 90% of cases in a parking garage.

2.2.2 Outdoor localization systems

GPS. The Global Positioning System (GPS) [92] determines the location of a receiver by measuring the time-of-flight (ToF) of radio signals from satellites orbiting around the globe. The satellites constitute the infrastructure and continuously transmit messages indicating the time of the transmission and their position at that time instant. The receiver uses these information from at least four satellites to establish a precise time reference and calculate its three dimensional location using trilateration. The nominal accuracy of GPS is 15 meters, while differential GPS is orders of magnitude better.

CALAMARI. The CALAMARI [99] is an ad-hoc localization system for sensor networks that uses both RSS readings and ultrasound ToF measurements for ranging. The ranging measurements lack infrastructure support and the device parameters are assumed to have high variance. Rather than calibrating each device individually, CALAMARI treats calibration as a joint parameter estimation problem, which assigns parameters to the individual devices based on a system-level optimization criteria. Acoustic ToF based experiments with 32 nodes placed on a 30 cm x 30 cm grid showed that calibration approach of CALAMARI can reduce the the mean ranging error from 75% to 10%.

AHLoS. The AHLoS [105] is a distributed localization system that relies on RSS or ultrasound ToF measurements for ranging. It assumes an initial set of nodes with known positions and uses iterative multilateration to resolve the location of further nodes. The evaluation of AHLoS started with the experimental characterization of the ranging error for both RSS and ultrasound ranging, yielding mean errors of 2-4 m and 2 cm, respectively. These errors then served as a parameter for scalability simulations of the iterative multilateration algorithm.

RIPS. The RIPS [9][8] indirectly measures the *phase* difference of radio signals to obtain information on the spatial relationship between nodes in an ad-hoc network. A basic ranging measurement relies on four nodes, where two transmitters emit a constant amplitude unmodulated carrier at slightly different frequencies, $f_A = f + \delta$ and $f_B = f - \delta$, and two receivers individually estimate the *absolute* phase offsets, ϑ_C and ϑ_D, from the interference signal, see Figure 27. The $\vartheta_D - \vartheta_C$ *relative* phase offset, the difference between the *absolute* phase offsets, then corresponds to the modulo 2π

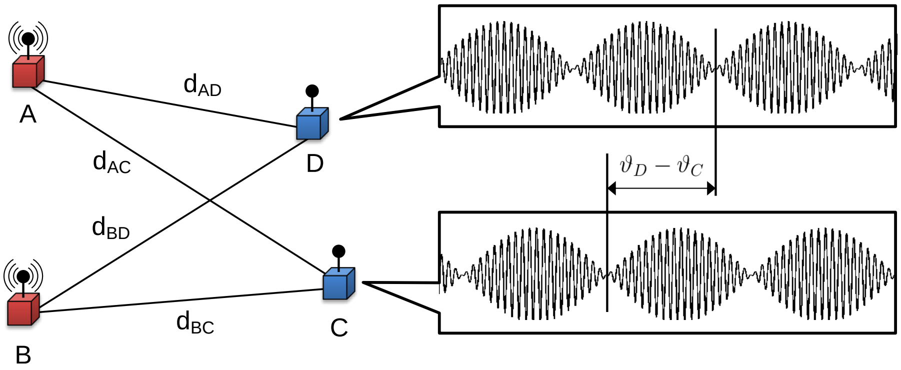


Figure 27: The basic four-node RIPS phase measurement setup.

linear combination of the node distances according to

$$\begin{aligned}\vartheta_D(f) - \vartheta_C(f) = 2\pi f\, &\frac{-d_{BD} + d_{AD} + d_{BC} - d_{AC}}{c} \\ + 2\pi\delta\, &\frac{d_{BD} + d_{AD} - d_{BC} - d_{AC}}{c} - 4\pi\delta t_e \pmod{2\pi},\end{aligned} \tag{4.1}$$

where c is the speed of light, δ is the low-frequency beat signal, t_e is the timing error, and the terms of the second line are generally disregarded based on distance and time synchronization related assumptions. The basic phase measurement is repeated multiple times to reduce the phase estimation error, and over multiple f frequencies to resolve the modulo 2π ambiguity. The measured *relative* phase offsets are used in RIPS [9] to first estimate the linear combination of the pairwise node distances and then to fuse the ranges from multiple node combinations to determine the node positions. In contrast, the method in SRIPS [8] fuses the *relative* offsets from different node-quartet directly. The two methods provide an average location accuracy of 3 cm and 50 cm, respectively. This considerably outperforms the existing approaches based on direct RSS ranging at the cost of significant post-processing effort at the basestation.

TripLoc. The TripLoc [112] is an infrastructure-based localization system that relies on radio interferometric phase measurements to estimate the bearing to a node with respect to anchors. Similarly to RIPS, a phase measurement requires four cooperating nodes, however, in TripLoc one receiver and two transmitters are placed close to each other at a known position to form an anchor. In this setup the measured relative phase offset bears information regarding the direction towards the other receiver, which can be exploited to determine the position of the node by triangulation. TripLoc calculates the direction of arrival estimates on the nodes in 0.5 s, and achieves an average accuracy of approximately 3°.

Numerous localization techniques have been proposed for WSNs that rely on different signal modalities and achieved widely varying accuracy. The original radio signal phase-based localization system, RIPS [9], gained significant attention primarily because it used only the readily available radio chip for distance measurement and because it provided more accurate location estimates in low-multipath environments than other radio-based approaches. The RIPS inspired further real-world experimentation [8][110][112], as well as theoretical analysis [113][114][8][115] of phase-based location estimation. Commonly, each of these approaches either relied on the interferometric phase measurement or assumed its *relative* phase offset output at hand, then attempted to improve a later stage of the localization process.

The following sections focus on the *measurement* stage of radio signal phase-based wireless sensor node localization, and discuss its underlying phase and distance estimation problems separately. First, Section 3 points out that interferometry is one possible implementation for *relative* phase offset measurement and proposes a completely different phase measurement approach, the analysis of which implicitly covers the *calibration* stage. Second, Section 4 introduces novel method for distance estimation, to arrive at the quad-range metric defined in (4.2). However, the non-linear optimization problem of the *localization* stage is considered to be solved [9][8][112], and is out of the scope of the upcoming discussion.

3 Multi-Carrier Phase Estimation

3.1 Overview

Existing radio signal phase-based WSN localization systems [9][8][110][112] primarily rely on interferometry for the first part of the *measurement* stage, to measure the *relative* phase offsets. Indeed, the interferometric phase measurement requires only the transmission of unmodulated sinusoidal carriers and minimal receiver-side signal processing, which is well suited to the resource constraint WSN nodes equipped with inflexible radio architectures.

Assuming moderate signal processing capability on both the transmitter and receivers sides, this section presents a more general description of the *relative* phase offset measurement to emphasize that the underlying principle of RIPS is essentially *differential TDOA* (dTDOA) estimation. The resulting phase measurement model and the overview of a multi-carrier communication scheme then provide clear reference points to compare the original interferometric approach, see [9] and Appendix B, with the proposed multi-carrier phase estimation method elaborated in Sections 3.2 and 3.3. The models assume perfect frequency synchronization and the effects of carrier frequency offset are discussed separately in Section 3.4.

3.1.1 General System Model

Consider the basic four-node setup shown in Figure 27, where A and B denote two radio transmitters and C and D two receivers. Rather than transmitting unmodulated sinusoids, let $u_A(t)$ and $u_B(t)$ be two *arbitrary* baseband complex waveforms assigned to nodes A and B, respectively. Then, the transmitted radio signals can be written as

$$s_A(t,f) = \mathrm{Re}\left[u_A(t-t_A)\, e^{j(2\pi f t+\varphi_A)}\right] \tag{4.2}$$

$$s_B(t,f) = \mathrm{Re}\left[u_B(t-t_B)\, e^{j(2\pi f t+\varphi_B)}\right], \tag{4.3}$$

where t_A and t_B signify the onset of the baseband waveforms plus any delay in the transmit path, φ_A and φ_B are the initial phase of the transmitter local oscillators and f is the carrier frequency.

Let Y denote either receiver C or D. Then the transmitted signals from A and B arrive at node Y with delays τ_{AY} and τ_{BY}, respectively. The two passband signal components are

$$s_{AY}(t,f) = \text{Re}\left[u_A(t - t_A - \tau_{AY})e^{j[2\pi f(t-\tau_{AY})+\varphi_A]}\right] \tag{4.4}$$
$$s_{BY}(t,f) = \text{Re}\left[u_B(t - t_B - \tau_{BY})e^{j[2\pi f(t-\tau_{BY})+\varphi_B]}\right], \tag{4.5}$$

where $\tau_{AY} = d_{AY}/c$ and $\tau_{BY} = d_{BY}/c$ are the path delays introduced by the transmitter-receiver distances with c denoting the speed of light. The down-mixed and low-pass filtered baseband complex signal is the superposition of the components

$$r_{AY}(t,f) = u_A(t - t_A - \tau_{AY})e^{j[2\pi f(t-\tau_{AY})+\varphi_A]}\ e^{-j[2\pi ft+\varphi_Y]} \tag{4.6}$$
$$r_{BY}(t,f) = u_B(t - t_B - \tau_{BY})e^{j[2\pi f(t-\tau_{BY})+\varphi_B]}\ e^{-j[2\pi ft+\varphi_Y]}, \tag{4.7}$$

where φ_Y is the initial phase of the receiver local oscillator.

Now assume that the above to signal components can be separated by some means and their phase can be precisely measured. Then the *absolute* phase difference, $\vartheta_Y(t,f) = \angle r_{BY}(t,f) - \angle r_{AY}(t,f)$, can be expressed as

$$\begin{aligned}\vartheta_Y(t,f) = {} & \angle u_B(t - t_B - \tau_{BY}) - \angle u_A(t - t_A - \tau_{AY}) \\ & + 2\pi f\,(-\tau_{BY} + \tau_{AY}) + \varphi_B - \varphi_A,\end{aligned} \tag{4.8}$$

where $\angle$ denotes the angle of a complex value, taken modulo 2π for convenience. The important aspect of (4.8) is that the term related to the receiver local oscillator phase, φ_Y, dropped out.

Now consider $\gamma_{ABCD}(t,f) = \vartheta_D(t,f) - \vartheta_C(t,f)$, the *relative* phase difference between receivers C and D,

$$\begin{aligned}\gamma_{ABCD}(t,f) = {} & \angle u_B(t - t_B - \tau_{BD}) - \angle u_A(t - t_A - \tau_{AD}) \\ & - \angle u_B(t - t_B - \tau_{BC}) + \angle u_A(t - t_A - \tau_{AC}) \\ & + 2\pi f\,(\tau_{BC} - \tau_{AC} - \tau_{BD} + \tau_{AD})\end{aligned} \tag{4.9}$$

and observe that the transmitter initial phases φ_A and φ_B also disappeared. Moreover, the only time-dependent terms are the ones associated with the baseband waveforms.

The above expression of $\gamma_{ABCD}(t,f)$ clearly reflects that the goal of the *relative* phase offset estimation is to precisely measure the difference between two TDOAs, $\tau_{BC} - \tau_{AC}$ and $\tau_{BD} - \tau_{AD}$, which can be used to obtain some distance metric using $d = c\tau$, as described in Section 4. Therefore, *the main challenge of the phase measurement is to find appropriate $u_X(t)$ baseband waveforms that enable the precise measurement of the* relative *phase offset without corrupting its value.* To illustrate the problem, consider the following simplified model.

Simplified model. A straightforward and simple choice is to let $u(t) \triangleq 1$ and to assume again that the two components $r_{AY}(t,f)$ and $r_{BY}(t,f)$ can be separated at receiver Y by some means. Thus, if the corresponding phases can be accurately measured, then the *absolute* phase offset reduces to

$$\vartheta_Y(t,f) = 2\pi f\,(-\tau_{BY} + \tau_{AY}) + \varphi_B - \varphi_A, \tag{4.10}$$

which is now independent of time. Moreover, since the carrier frequency is a controlled parameter the only undesirable terms in (4.10) are the transmitter phases φ_A and φ_B.

Consequently, the *relative* phase offset between the two receivers also becomes time-independent, and the transmitter phases cancel out

$$\gamma_{ABCD}(f) = 2\pi f\left(\tau_{BC} - \tau_{AC} - \tau_{BD} + \tau_{AD}\right). \tag{4.11}$$

While (4.11) contains only the terms of interest, the fundamental flaw is that with $u(t) \triangleq 1$ the two signals $r_{AY}(t, f)$ and $r_{BY}(t, f)$ become unseparable at the receivers, which make the measurement of the *absolute* phase offsets infeasible. The RIPS [9] addressed this problem using a single-carrier interferometric approach, described with the current notation in Appendix B. Meanwhile, an alternative multi-carrier approach is presented in Sections 3.2 and 3.3.

3.1.2 Orthogonal Frequency Division Multiplexing

Multi-carrier modulation schemes have gained increasing popularity in digital wireless communication systems during the past decades, primarily due to their ability to handle intersymbol interference (ISI), thus combat frequency selective fading. One such multi-carrier scheme is the spectrally efficient Orthogonal Frequency Division Multiplexing (OFDM) that forms the basis of numerous recent communication standards, including 802.15.4g [116], 802.11a/g/n [117], Digital Video Terrestrial Broadcasting (DVB) and 3GPP Long Term Evolution (LTE) [118].

The underlying principle of OFDM is to divide the available channel bandwidth into several narrowband subchannels and modulate each allocated subcarrier simultaneously [93]. In OFDM each subchannel is associated with a sinusodial carrier of the form

$$u_{c,k}(t) = \mathrm{Re}\left[e^{j2\pi k\Delta f t}\right] \qquad 0 \le k < N, \tag{4.12}$$

where $\Delta f \triangleq 1/T$ is the frequency separation and T is the OFDM symbol duration. The subcarriers are mutually orthogonal over the symbol time T regardless of their phase relationship,

$$\int_0^T \mathrm{Re}\left[e^{j(2\pi k\Delta f t+\phi_k)}\right] \mathrm{Re}\left[e^{j(2\pi l\Delta f t+\phi_l)}\right] dt = 0 \qquad k \ne l, \tag{4.13}$$

where $0 \le k, l < N$, and $\phi_k, \phi_l \in$ [-π, π] are arbitrary phases.

Assuming that the k^{th} subcarrier is modulated with the complex constellation M_k, the complex baseband waveform can be expressed as

$$u(t) = \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} M_k\, e^{j2\pi k\Delta f t}, \tag{4.14}$$

where $M_k = m_k\, e^{j\phi_k}$ with $m_k \in [0, 1]$ and $\phi_k \in$ [-π, π], and the transmitted passband signal becomes

$$s(t) = \mathrm{Re}\left[\sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} M_k\, e^{j[2\pi(f+k\Delta f)t+\varphi_X]}\right] \tag{4.15}$$

Now consider the propagation model. When N is sufficiently large, the narrow subchannels can be individually characterized with the complex-valued frequency response

$$C(f_k) = C_k = g_k e^{j\theta_k} \qquad 0 \leq k < N, \tag{4.16}$$

where g_k is the attenuation and θ_k is the phase shift of the channel between the transmitter and the receiver, plus additive noise. Therefore, the received passband signal on the k^{th} subchannel can be written as

$$s_k(t) = \mathrm{Re}\left[\frac{1}{\sqrt{N}} C_k M_k e^{j[2\pi(f+k\Delta f)t+\varphi_X]}\right] + \nu_k(t) \qquad 0 \leq k < N, \tag{4.17}$$

with $\nu_k(t)$ representing the additive noise on the corresponding subchannel. After complex down-mixing the baseband signal waveform on the k^{th} subchannel is of the form

$$r_k(t) = \frac{1}{\sqrt{N}} C_k M_k e^{j[2\pi k\Delta f t+\varphi_X-\varphi_Y]} + n_k(t) \qquad 0 \leq k < N, \tag{4.18}$$

where φ_Y is the receiver local oscillator phase and the $n_k(t)$ additive noise component are generally modeled as mutually statistically independent white Gaussian random processes.

A coherent detection OFDM receiver estimates the C_k subchannel frequency responses first, and compensates with the $\hat{C}_k$ estimates while computing the correlation metrics [93, p. 738]

$$\mathrm{CM}_k = \mathrm{Re}\left[\hat{C}_k^* \int_0^T r_k(t) e^{-j2\pi k\Delta f t} dt\right] \qquad 0 \leq k < N. \tag{4.19}$$

The receiver then estimates the transmitted M_k constellations for each subcarrier based on the calculated CM_k correlation metrics.

In search for a novel method for phase-based localization, the fundamental idea of OFDM serves as a starting point. Observe that the actual M_k constellations carried by the OFDM symbols are not important, only the estimation of the C_k channel parameters are of interest. In OFDM terms, the multi-carrier phase estimation problem of Section 3.2 may be phrased as the *simultaneous estimation of the $\theta_k = \angle C_k$ phase shifts between the receiver and multiple transmitters.* Consequently, the OFDM-based system model underpins the proposed multi-carrier phase estimation approach of the next section. Following the same notation, the details of the *relative* phase offset estimation are described in Section 3.3. Then, several aspects of the proposed *relative* phase offset estimation method are analyzed in Section 3.4.

3.2 Multi-Carrier System Model

The OFDM subcarriers are by definition orthogonal to each other and naturally occupy the allocated frequency band with uniform spacing. The multi-carrier phase offset estimation method presented in this section exploits these properties to perform phase measurements on multiple subchannels simultaneously and to derive the *relative* phase offsets from them.

First, the transmitter side of the usual four-node measurement setup is considered in order to define the basic constraints for the construction of OFDM symbols. In short, each transmitter is required to occupy at least two subcarriers, but no subcarrier should be allocated to both transmitters. The occupied subcarriers are allowed to be arbitrarily rotated by predefined phases to shape

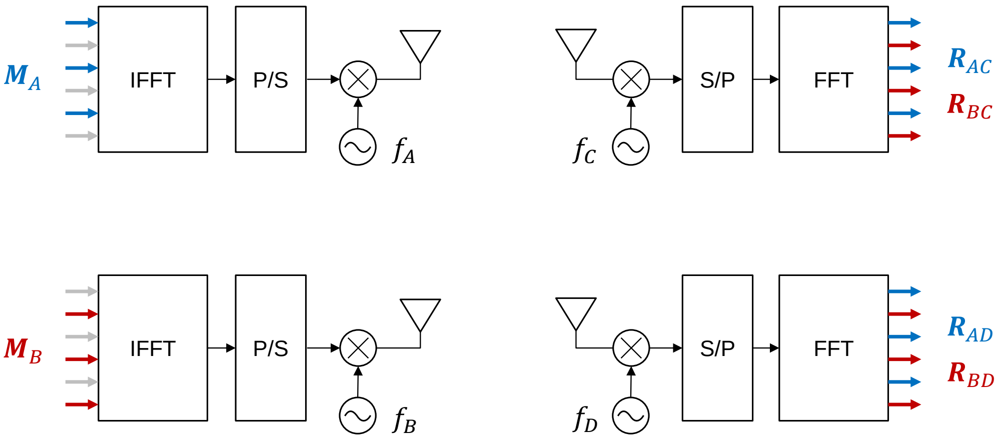


Figure 28: Block diagram of the basic OFDM-based phase measurement.

the transmitted signal, as long as their values are known at the receivers. Second, the receiver signal model and the raw phase estimation of the received subcarriers is discussed.

3.2.1 Transmitter side

Let A and B denote the transmitter pair, while C and D be the receiver pair in the four-node measurement setup pictured in Figure 27 and with the architectures shown in and Figure 28. Furthermore, let the transmitters utilize a set of mutually orthogonal signals as the baseband complex waveform according to

$$u_A(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \underbrace{m_{A,k}\, e^{j\phi_{A,k}}}_{M_{A,k}} \; e^{j2\pi k\Delta f t} \tag{4.20a}$$

$$u_B(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \underbrace{m_{B,k}\, e^{j\phi_{B,k}}}_{M_{B,k}} \; e^{j2\pi k\Delta f t}, \tag{4.20b}$$

where $k\Delta f$ is the frequency of the k^{th} subcarrier and the range of the symbols $M_{A,k}$ and $M_{B,k}$ is restricted to

$$m_{A,k}, m_{B,k} \in \mathcal{M} = \{0, 1\} \quad \text{and} \quad \phi_{A,k}, \phi_{B,k} \in \mathcal{P} = [\text{-}\pi, \pi)\,. \tag{4.21}$$

Let the symbols further be constrained by

$$m_{A,k} \cdot m_{B,k} = 0 \qquad 0 \le k < N \tag{4.22}$$

$$\sum_{k=0}^{N-1} m_{A,k} \ge 2 \quad \text{and} \quad \sum_{k=0}^{N-1} m_{B,k} \ge 2 \tag{4.23}$$

That is, each transmitter uses at least two subcarriers and a given subcarrier is assigned to at most one transmitter. The subcarrier phases may take arbitrary pre-defined values that are fixed for the duration of the phase measurement.

Note that the above definition of the baseband complex waveforms is essentially equivalent to the continuous transmission of a constant OFDM symbol without cyclic prefix. This, in turn, means that computationally efficient algorithms, based on Inverse Fast Fourier Transform (IFFT), are available to generate arbitrary waveforms of (4.20). A simple transmitter architecture for this task is depicted in Figure 28, however, existing OFDM transceivers may also be suitable.

After complex modulation the transmitted passband radio signal of the two receivers can be expressed as

$$s_A(t) = \mathrm{Re}\left[\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} m_{A,k}\, e^{j\phi_{A,k}}\, e^{j2\pi k\Delta f(t-t_A)}\, e^{j(2\pi ft+\varphi_A)}\right] \tag{4.24a}$$

$$s_B(t) = \mathrm{Re}\left[\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} m_{B,k}\, e^{j\phi_{B,k}}\, e^{j2\pi k\Delta f(t-t_B)}\, e^{j(2\pi ft+\varphi_B)}\right], \tag{4.24b}$$

where t_A and t_B mark the onset of the baseband waveforms and φ_A and φ_B are the initial phase of the transmitter local oscillators.

3.2.2 Receiver side

The signals transmitted by node A and B arrive to receiver Y with path delays $\tau_{AY} = d_{AY}/c$ and $\tau_{BY} = d_{AY}/c$. The superposition of the impinging passband signals can then be written as

$$\begin{aligned} s_Y(t) = {} & \mathrm{Re}\left[\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} m_{A,k}\, e^{j\phi_{A,k}}\, e^{j[2\pi(f+k\Delta f)(t-\tau_{AY})-2\pi k\Delta f t_A+\varphi_A]}\right] \\ & + \mathrm{Re}\left[\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} m_{B,k}\, e^{j\phi_{B,k}}\, e^{j[2\pi(f+k\Delta f)(t-\tau_{BY})-2\pi k\Delta f t_B+\varphi_B]}\right]. \end{aligned} \tag{4.25}$$

According to assumption (4.22) a particular subchannel is used exclusively by only one transmitter. Thus, the complex down-mixed and low-pass filtered waveform can now be naturally separated into two sets of orthogonal waveforms

$$r_{AY}(t) = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} \underbrace{m_{A,k} e^{j\phi_{A,k}} e^{j[2\pi k\Delta f(t-t_A-\tau_{AY})-2\pi f\tau_{AY}+\varphi_A-\varphi_Y]}}_{R_{AY,k}(t)} \tag{4.26a}$$

$$r_{BY}(t) = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} \underbrace{m_{B,k} e^{j\phi_{B,k}} e^{j[2\pi k\Delta f(t-t_B-\tau_{BY})-2\pi f\tau_{BY}+\varphi_B-\varphi_Y]}}_{R_{BY,k}(t)} \tag{4.26b}$$

that correspond to transmitters A and B, respectively. Then, the received baseband waveform of the k^{th} subcarrier is

$$R_{Y,k}(t) = \begin{cases} R_{AY,k}(t) & \text{if } m_{A,k} \neq 0 \\ R_{BY,k}(t) & \text{if } m_{B,k} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.27}$$

which is illustrated in Figure 28 for both receivers C and D.

A particularly convenient way to estimate the phase of $R_{Y,k}$ at a given time instant is through the complex Fourier coefficient

$$R_{Y,k}(t_Y) = \frac{1}{T} \int\limits_{t_Y}^{t_Y+T} r_Y(t) e^{-j2\pi k \Delta f t} dt, \tag{4.28}$$

where t_Y is the receiver onset time and $T = 1/\Delta f$.

Now consider the rearranged arguments of $R_{AY,k}(t)$ and $R_{BY,k}(t)$

$$\angle R_{AY,k}(t) = 2\pi k \Delta f (t - t_A) - 2\pi (f + k\Delta f) \tau_{AY} + \phi_{A,k} + \varphi_A - \varphi_Y \tag{4.29a}$$

$$\angle R_{BY,k}(t) = 2\pi k \Delta f (t - t_B) - 2\pi (f + k\Delta f) \tau_{BY} + \phi_{B,k} + \varphi_B - \varphi_Y, \tag{4.29b}$$

taken modulo 2π for convenience. Remember, that since $\phi_{A,k}$ and $\phi_{B,k}$ are design parameters, they can be compensated for at the receiver by a simple complex multiplication.

When addressed to multiple subcarriers simultaneously, this task is equivalent to the reception of an OFDM symbol with the sampling started at t_Y. Therefore, the Fast Fourier Transform (FFT) provides a computationally efficient solution, similarly as IFFT on the transmitter side. Again, Figure 28 shows one such simple receiver architecture suitable for the phase measurement.

3.3 Relative Phase Offset Estimation

In the single-carrier RIPS [9] the *absolute* phase offset estimation is inherent in the interferometric phase measurement, which is performed at the receiver as described in Appendix B-1. Therefore, the straightforward way to calculate the *relative* phase difference is to perform an explicit subtraction at some central point where the *absolute* phase offsets from multiple receivers are available.

In contrast, the multi-carrier approach of Section 3.2 measures the subcarrier phases directly, without implicitly calculating their offset. In fact, the fundamental challenge becomes the estimation of the phase offset at frequencies where the subcarrier from only one transmitter is available due to (4.22). Having undistorted phase measurements at multiple frequencies, however, gives means to restore the missing data points and to obtain the *relative* phase offsets in new ways. This section presents two such solutions for *relative* phase offset estimation.

Simplified model. Before discussing algorithms for obtaining the *relative* phase offset from the aligned phases (4.57), consider the following simplified case, similar to the one in Section 3.1.1. Assume temporarily that the subcarriers of the two transmitters occupy the *same* subchannels and that their phase can be estimated individually. Then the *absolute* phase offset on the k^{th} subchannel would be

$$\vartheta_{Y,k}(t,f) = 2\pi k \Delta f \left(-t_B + t_A\right) + 2\pi (f + k\Delta f) \left(-\tau_{BY} + \tau_{AY}\right) + \varphi_B - \varphi_A \pmod{2\pi}, \tag{4.30}$$

which is now time-independent as $\vartheta_{Y,k}(t,f) = \vartheta_{Y,k}(f)$. Similarly as in (4.10), the undesirable terms are not a function of the receiver parameters and they drop out from the *relative* phase offset. Therefore, the $\vartheta_{D,k}(f) - \vartheta_{C,k}(f)$ phase difference on the k^{th} subcarrier would become

$$\gamma_{ABCD,k}(f) = 2\pi (f + k\Delta f) \left(-\tau_{BD} + \tau_{AD} + \tau_{BC} - \tau_{AC}\right) \pmod{2\pi}, \tag{4.31}$$

which is essentially equivalent to (4.11) tuned to carrier frequency $f + k\Delta f$. Furthermore, since the subcarriers are spaced uniformly in frequency, $\gamma_{ABCD,k}(f)$ is linear in both f and k with modulo 2π ambiguity. Finally, note that since (4.31) is independent of time, the measurement may be repeated at the same f carrier frequency for averaging or at different frequencies, say $N\Delta f$ apart, to increase the range over which the *relative* phase offsets are obtained.

3.3.1 Interpolation

The major challenge is clearly to recover the missing phase measurement points. In search for suitable algorithms, the same linearity of $\angle R_{AY,k}(t)$ and $\angle R_{BY,k}(t)$ in f and k may be exploited in conjunction with constraint (4.23). These two conditions together ensure that both $\angle R_{AY,k}(t)$ and $\angle R_{BY,k}(t)$ have at least two directly measured subcarrier phases that can be *interpolated* due to their linearity in k, both subject to modulo 2π wrapping. A convenient approach directly follows from the following theorem:

Theorem 1. *The problem of missing phase measurement recovery is equivalent to estimating the frequency and phase of a complex sinusoid with additive noise from a discrete set of observations.*

Proof. Consider the definition of $R_{XY,k}(t)$ with $t = t_Y$ fixed and k as the running variable. Then observe that $R_{XY,k}(t_Y)$ can be modeled as a complex sinusoid corrupted by additive noise

$$\begin{aligned} R_{XY,k}(t_Y) &= e^{j[2\pi k\Delta f(t_Y - t_X - \tau_{XY}) - 2\pi f\tau_{XY} + \varphi_X - \varphi_Y]} + W \\ &= e^{j2\pi k\tau + \theta} + W, \end{aligned} \tag{4.32}$$

where k denotes the *time*, $\tau = \Delta f(t_Y - t_X - \tau_{XY})$ is now the *frequency*, θ is the phase and $W \sim \mathcal{CN}(0, \sigma_R)$ is an additive complex white Gaussian noise (CWGN). □

Such interpretation of the problem calls for several remarks. First, having two discrete sample points to estimate the complex single tone of 4.32 is necessary, but not sufficient in general. In the presence of noise, the accurate frequency and phase estimation require significantly more statistics, and the spacing and arrangement of the sample locations also call for detailed planning, see Sections 3.4 and 4.4, respectively.

Second, the linearity in k is due to the linear phase shifts of the channel, which is justified by the assumption of a multipath-free propagation environment between all possible transmitter-receiver pair combinations. Furthermore, the CWGN noise model is a natural consequence of (4.18).

Third, Theorem 1 designates the problem as *frequency* estimation, however, it is actually the *time* measure, $t_Y - t_X - \tau_{XY}$, of particular interest. The simple substitution in notation emphasizes that, in this case, time estimation is equivalent to frequency estimation, which is a well known problem.

Frequency and phase estimation by linear regression on the signal phase is proposed in [119]. The shortcoming of this approach is that the phase is wrapped according to modulo 2π, which has to be resolved prior to regression. A computationally more efficient method is described in [120], which operates on differenced phase data and, therefore, implicitly addresses the unwrapping problem too. Two similar approaches with comparable performance and complexity are proposed in [121] and in [122]. The discrete Fourier transform (DFT) with different weighting functions is considered in [123] for coarse frequency estimation. The classic [124] derives the appropriate maximum-likelihood (ML) estimators and discusses their relationship with the DFT to obtain practical algorithms. And recently, [125] and [126] investigated the relationship between frequency estimation and an algorithmic number theory problem known as the nearest lattice point problem. Furthermore, several other spectral estimation algorithms are discussed in books such as [127] or [128].

The proposed interpolation algorithm for restoring the missing points of $R_{XY,k}$ is based on the ML frequency estimator for a complex sinusoid with unknown phase, that is on the maximization of the periodogram [124]. The details of this MLE are discussed in Section 4.2, while the steps of the interpolation are summarized in Algorithm 1.

```
Data: Direct phase measurements $R_{XY,k} \quad \forall k\colon\ m_{X,k} \neq 0$
Result: Estimated subcarrier phases $\hat{R}_{XY,k} \quad \forall k\colon\ m_{X,k} = 0$
begin
    1. Estimate $\tau = t_Y - t_X - \tau_{XY}$ using:
       $\hat{\tau}_{\mathrm{ML}} = \arg\max_{\tau} \Upsilon(\tau)\Upsilon^*(\tau), \quad \text{where} \quad \Upsilon(\tau) \triangleq \sum_{k=0}^{N-1} R_{XY,k}\, e^{-j2\pi k\Delta f\tau}$
    2. Estimate $\theta = -2\pi f\tau_{AY} + \varphi_X - \varphi_Y$ using $\hat{\tau}_{\mathrm{ML}}$:
       $\hat{\theta}_{\mathrm{ML}} = \angle\Upsilon(\hat{\tau}_{\mathrm{ML}})$
    3. Estimate the missing points using $\hat{\tau}_{\mathrm{ML}}$ and $\hat{\theta}_{\mathrm{ML}}$:
       $\hat{R}_{XY,k} \triangleq e^{j(2\pi k\Delta f\hat{\tau}_{\mathrm{ML}} + \hat{\theta}_{\mathrm{ML}})}$
end
```

Algorithm 1: Interpolation of the direct phase measurements to obtain phase estimates on subchannels where the subcarrier is actually missing.

3.3.2 Phase Offset Estimation Algorithms

Given a means to reclaim the missing phases on all subchannels from only a subset of direct observations allows for different ways to arrive at the $\gamma_{ABCD,k}$ *relative* phase offsets. The description of two possible approaches are as follows.

Method 1: Differences of Time Differences of Arrival. The straightforward multi-carrier adaptation of the original RIPS is to first obtain $R_{XY,k}$ for all combinations of transmitter $X \in \{A, B\}$ and receiver $Y \in \{C, D\}$ either by direct measurement or interpolation. Then explicitly calculate the

$$\vartheta_{Y,k} = \angle R_{BY,k} - \angle R_{AY,k} \tag{4.33}$$

absolute phase differences, which operation is inherent in the single-carrier interferometric measurement. Finally, subtract them to obtain the

$$\gamma_{ABCD,k} = \vartheta_{D,k} - \vartheta_{C,k} = (\angle R_{BD,k} - \angle R_{AD,k}) - (\angle R_{BC,k} - \angle R_{AC,k}) \tag{4.34}$$

relative phase offsets. The process is illustrated for $N = 32$ subcarriers in Figure 29 and the steps are detailed in Algorithm 2.

An interesting aspect of this approach is that the *relative* phase offsets may be calculated either at the receiver or at the basestation. The former allows to distribute parts of the computations among the nodes, while the latter might be preferable if the basestation offers superior computing performance, such as floating point numerical representation. In both cases, however, the communication burden is identical, since either occupied carrier number $R_{Y,k}$ or $\vartheta_{Y,k}$ values need to be transmitted to the basestation.

Obtaining the *relative* phase offsets this way requires the interpolation of four datasets in total, the four variations of $R_{XY,k}$. Figure 29 illustrates the case for $N = 32$ subcarriers equally divided between the two transmitters, and aligned in an alternating fashion. The solid markers represent *direct* observations while hollow markers indicate data derived through *interpolation* at some point. Note that interpolation is performed early in the processing, even before calculating the *absolute* phase offsets.

Method 2: Differences of Time Differences of Departure. The alternative multi-carrier *relative* phase offset estimation method exploits that all combinations of $R_{XY,k}$ are available at some central place. That is, it calculates the

$$\xi_{X,k} = \angle R_{XD,k} - \angle R_{XC,k} \tag{4.35}$$

per-transmitter differences first. Then, it recovers the missing $\xi_{X,k}$ points and obtains the *relative* phase offsets through

$$\gamma_{ABCD,k} = \xi_{D,k} - \xi_{C,k} = (\angle R_{BD,k} - \angle R_{BC,k}) - (\angle R_{AD,k} - \angle R_{AC,k}). \tag{4.36}$$

The steps of this approach are summarized in Algorithm 3.

Note that such sequence of the phase-differencing operations is not possible with the single-carrier approach, as the *absolute* phase offset estimation is an inherent part of the interferometric phase measurement. Furthermore, since all $R_{XY,k}$ are assumed to be present at the same place, all calculations have to be performed at the basestation. This implies no additional communication costs because, similarly to the previous approach, only the $R_{Y,k}$ observations of each occupied subchannels need to be transmitted to the basestation.

Obtaining the *relative* phase offset through this alternative method requires the interpolation of only two datasets in total, $\xi_{A,k}$ and $\xi_{B,k}$. Less interpolation is preferable as it leads to less perturbation of the sample statistics, which is important for the distance estimator proposed in Section 4. Furthermore, as a consequence of calculating the per-transmitter differences, the $\phi_{X,k}$ phase shifts drop out implicitly. Therefore, Algorithm 3 can operate on arbitrary OFDM symbols without their knowledge at the receivers, as opposed to Algorithm 2.

Figure 30 depicts the algorithm steps for $N = 32$ subcarriers assigned alternately to the two transmitters. Again, the solid markers signify *direct* observations or their difference, whereas hollow markers betoken data obtained by interpolation at some point.

Algorithms 2 and 3 propose two alternative solutions to calculate the same *relative* phase offset. Interpreting the *relative* phase offset as a DTDOA estimator of $\tau_{BD} - \tau_{AD} - \tau_{BC} + \tau_{AC}$, the two approaches differ only in the order they calculate the two time differences. Algorithm 2 calculates the path delay difference between a receiver and two transmitters, $\tau_{BY} - \tau_{AY}$, whereas Algorithm 3 determines that between a transmitter and two receivers, $\tau_{XD} - \tau_{XC}$, first.

Detailed examination of the underlying steps, however, reveals that Algorithm 3 has two advantages. First, it can operate independently of the OFDM symbol used, which is convenient when considering the integration into OFDM communication systems. Second, it requires half as many *interpolation* steps as Algorithm 2. Since the *interpolation* method in Algorithm 1 has an averaging side effect, its excessive use may considerably alter the statistics of the final *relative* phase offset estimates. This, in turn, can negatively affect the performance of the distance estimation presented in Section 4.

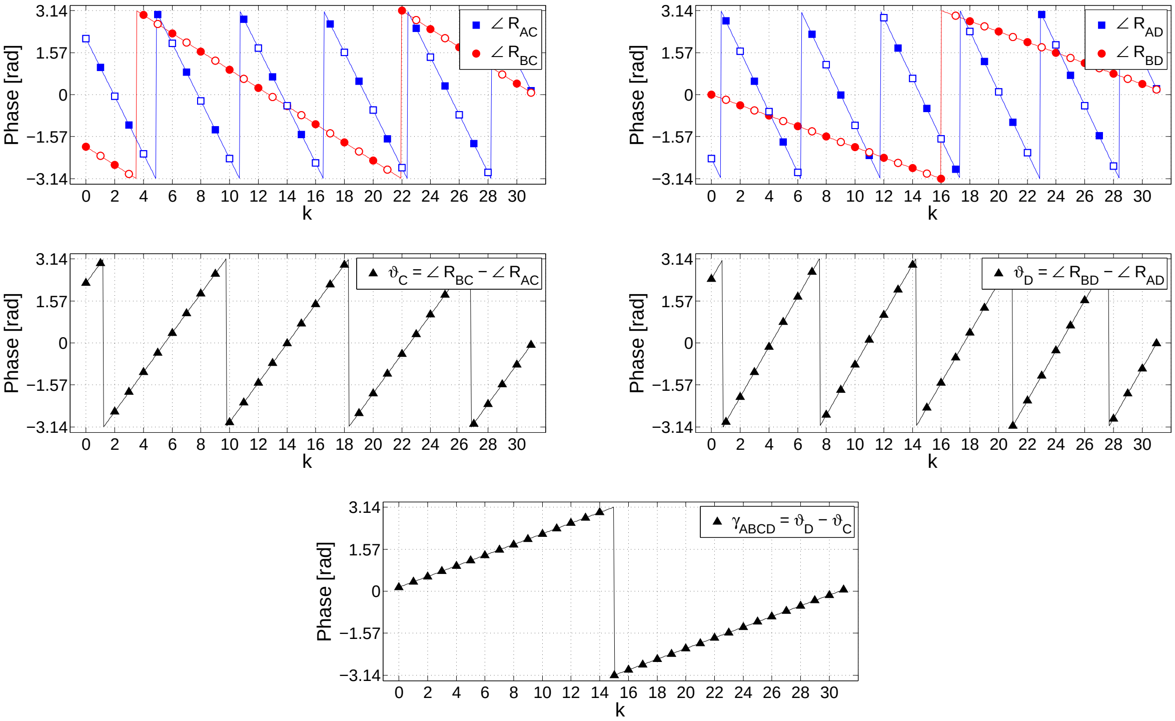


Figure 29: *Relative* phase offset calculation using explicit *absolute* phase offset estimation based on a multi-carrier phase measurement with $N = 32$ subcarriers.

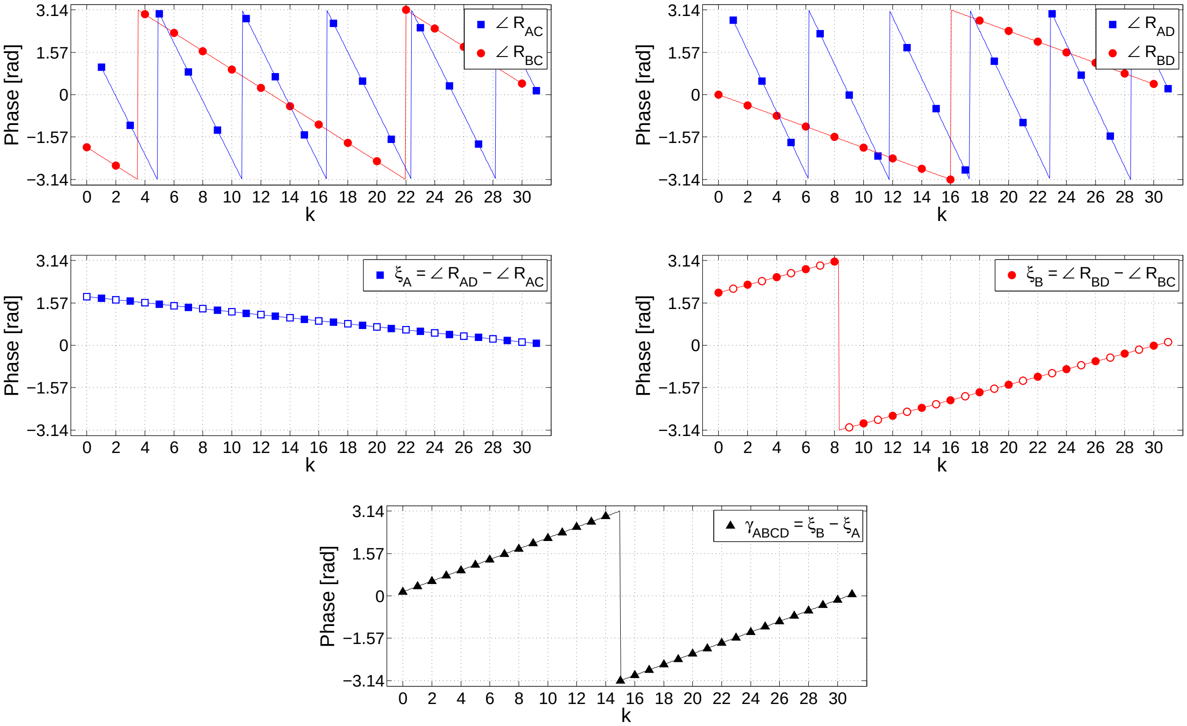


Figure 30: *Relative* phase offset calculation with *no* explicit *absolute* phase offset estimation based on a multi-carrier phase measurement with $N = 32$ subcarriers.

```
Data: Receiver waveforms $r_C(t), r_D(t)$
Result: Relative phase offsets for $\forall k : m_{A,k} \neq 0 \cup m_{B,k} \neq 0$
begin
    for receiver $Y \in \{C, D\}$ do
        Measure the phase on all occupied subcarriers:
            $\angle R_{Y,k} = \frac{1}{T}\int_T r_Y(t)e^{-j2\pi k\Delta f t}dt \quad \forall k : m_{A,k} \neq 0 \cup m_{B,k} \neq 0$
        Rotate phases according to $\phi_k$:
            $\angle R_{Y,k} = \angle R_{Y,k} - \phi_k \bmod 2\pi$
        for transmitter $X \in \{A, B\}$ do
            $\angle R_{XY,k} = \begin{cases} \angle R_{Y,k} & \text{if } m_{X,k} \neq 0 \\ \text{interpolate}(\angle R_{XY,k}) & \text{otherwise} \end{cases}$
        end
        Calculate absolute phase offsets:
            $\vartheta_{Y,k} = \angle R_{BY,k} - \angle R_{AY,k} \bmod 2\pi$
    end
    Calculate $\gamma_{ABCD,k} = \vartheta_{D,k} - \vartheta_{C,k}$
end
```

Algorithm 2: *Relative* phase offset calculation with explicit *absolute* phase offset estimation.

```
Data: Receiver waveforms $r_C(t), r_D(t)$
Result: Relative phase offsets for $\forall k : m_{A,k} \neq 0 \cup m_{B,k} \neq 0$
begin
    for receiver $Y \in \{C, D\}$ do
        Measure the phase on all occupied subcarriers:
            $\angle R_{Y,k} = \frac{1}{T}\int_T r_Y(t)e^{-j2\pi k\Delta f t}dt \quad \forall k : m_{A,k} \neq 0 \cup m_{B,k} \neq 0$
    end
    for transmitter $X \in \{A, B\}$ do
            $\xi_{X,k} = \angle R_{XD,k} - \angle R_{XC,k} \quad \forall k : m_{X,k} \neq 0$
            $\xi_{X,k} = \begin{cases} \xi_{X,k} & \text{if } m_{X,k} \neq 0 \\ \text{interpolate}(\xi_{X,k}) & \text{otherwise} \end{cases}$
    end
    Calculate $\gamma_{ABCD,k} = \xi_{B,k} - \xi_{A,k}$
end
```

Algorithm 3: *Relative* phase offset calculation with *no* explicit *absolute* phase offset estimation.

3.4 Performance Analysis

3.4.1 Timing and Frequency Offset

The discussion of multi-carrier phase estimation disregarded the effects of synchronization errors so far to keep the descriptions tractable. However, carrier frequency offset (CFO) and timing offset (TO) have a substantial impact on the phase estimation performance, therefore, they should be analyzed and compensated for. The synchronization problems are addressed from two directions. First, synchronization inaccuracies between the two transmitters and that between the two receivers are considered, followed by their comparison with the single-carrier case. Second, the impact of CFO between transmitter and receiver is inspected, which is specific to the OFDM-based approach.

DTDOA Perspective. For simplicity and without the loss of generality let $m_{X,k} \triangleq 1$ and $\phi_{X,k} \triangleq 0$ for $\forall k$, which is the simplified model from Section 3.3. Moreover, let ε_A and ε_B denote the carrier frequency offset (CFO) of the transmitter local oscillators with respect to the nominal carrier frequency f. Then the transmitted passband signals can be written as

$$s_A(t) = \mathrm{Re}\left[\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{j2\pi k\Delta f(t-t_A)}\, e^{j[2\pi(f+\varepsilon_A)t+\varphi_A]}\right] \tag{4.37a}$$

$$s_B(t) = \mathrm{Re}\left[\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{j2\pi k\Delta f(t-t_B)}\, e^{j[2\pi(f+\varepsilon_B)t+\varphi_B]}\right], \tag{4.37b}$$

where Δf is the subcarrier frequency separation, t_A and t_B are the onset of the baseband waveforms and φ_A and φ_B are the initial phase of the transmitter local oscillators.

After complex down-mixing the baseband signal on the receiver side becomes

$$r_Y(t) = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} m_{A,k} e^{j\phi_{A,k}} e^{j[2\pi k\Delta f(t-t_A-\tau_{AY})-2\pi f\tau_{AY}+2\pi\varepsilon_A(t-\tau_{AY})+\varphi_A-\varphi_Y]} \tag{4.38a}$$

$$+\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} m_{B,k} e^{j\phi_{B,k}} e^{j[2\pi k\Delta f(t-t_B-\tau_{BY})-2\pi f\tau_{BY}+2\pi\varepsilon_B(t-\tau_{BY})+\varphi_B-\varphi_Y]}. \tag{4.38b}$$

Disregarding the technical details of interpolation, described in Section 3.3.1, and assuming that the direct phase measurements are available from both transmitters on any subchannel, the subcarrier phases on the k^{th} channel are

$$\angle R_{AY,k}(t) = 2\pi k\Delta f(t-\tau_{AY}-t_A) - 2\pi f\tau_{AY} + 2\pi\varepsilon_A(t-\tau_{AY}) + \varphi_A - \varphi_Y \tag{4.39a}$$

$$\angle R_{BY,k}(t) = 2\pi k\Delta f(t-\tau_{BY}-t_B) - 2\pi f\tau_{BY} + 2\pi\varepsilon_B(t-\tau_{BY}) + \varphi_B - \varphi_Y. \tag{4.39b}$$

Then, the *absolute* phase offsets can be written as

$$\vartheta_{Y,k}(t,f) = 2\pi f(\tau_{AY}-\tau_{BY}) + 2\pi k\Delta f(\tau_{AY}-\tau_{BY}+t_A-t_B)$$
$$+ 2\pi\varepsilon_B(t-\tau_{BY}) - 2\pi\varepsilon_A(t-\tau_{AY}) + \varphi_B - \varphi_A. \tag{4.40}$$

while the *relative* phase difference $\gamma_{ABCD,k}(f) = \vartheta_{D,k}(t_D, f) - \vartheta_{C,k}(t_C, f)$ as

$$\begin{aligned}\gamma_{ABCD,k}(f) &= 2\pi(f + k\Delta f)(\tau_{BC} - \tau_{AC} - \tau_{BD} + \tau_{AD}) \\ &\qquad + 2\pi\varepsilon_B(t_D - t_C - \tau_{BD} + \tau_{BC}) - 2\pi\varepsilon_A(t_D - t_C - \tau_{AD} + \tau_{AC}).\end{aligned} \tag{4.41}$$

After some rearrangement

$$\begin{aligned}\gamma_{ABCD,k}(f) &= 2\pi(f + k\Delta f)(\tau_{BC} - \tau_{AC} - \tau_{BD} + \tau_{AD}) \\ &\qquad + 2\pi\varepsilon_B(\tau_{BC} - \tau_{BD}) - 2\pi\varepsilon_A(\tau_{AC} - \tau_{AD}) + 2\pi \underbrace{(\varepsilon_B - \varepsilon_A)}_{\text{relative CFO}} \cdot \underbrace{(t_D - t_C)}_{\text{relative TO}}.\end{aligned} \tag{4.42}$$

A comparison with (4.31) reveals that all the unwanted terms in (4.42) are introduced through ε_A and ε_B, and that their effect can be separated for offsets between transmitter and receiver and that between two transmitters. The transmitter-receiver frequency misalignment brings in the time-independent terms $2\pi\varepsilon_X(\tau_{XD} - \tau_{XC})$ that are proportional to both the CFO itself and the constant path delay difference between the transmitter and the two receivers. The relative CFO between the two transmitters, on the other hand, introduces an error that is also dependent on the difference between the t_Y receiver start times.

Note that (4.42) is a generalization of the single-carrier case in (B.11), and with the notation $\varepsilon_A = +\delta$ and $\varepsilon_B = -\delta$ the two become equivalent. With the single-carrier approach, the presence of the δ frequency offset between the two transmitters is essential to make the interferometric phase measurement possible. In contrast, the multi-carrier approach requires no deliberately introduced frequency offset, hence, the CFO should be completely compensated for. Thus, synchronizing the carrier frequency of one transmitter to that of the other, thereby minimizing $\varepsilon_B - \varepsilon_A$, relaxes the relative time-synchronization requirement set between the two receivers. In terms of timing synchronization, this relative timing offset between the receivers is the only concern, since all the other terms related to transmitter timing dropped out of (4.42).

In the single carrier case, synchronizing the receivers to the transmitters offers no improvement in accuracy because δ is not be eliminated. However, establishing transmitter-receiver frequency synchronism in the multi-carrier approach reduces the undesired time-independent terms and alleviates an important synchronization issue described in the following section.

OFDM Perspective. A different synchronization problem arises with the multi-carrier nature of the phase measurement due to the close spacing of the subcarriers. To show this, consider the transmitted passband signal model defined in (4.37) and assume that the transmitter carrier frequencies are perfectly synchronized to each other, but not to that of the receiver. That is, let $\varepsilon_A = \varepsilon_B \triangleq 0$, and let $\varepsilon_Y \neq 0$ denote the CFO of the receiver relative to the transmitter. Then, after complex down-mixing the received baseband signal can be written as

$$r_Y(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} m_{A,k} e^{j\phi_{A,k}} e^{j[2\pi k\Delta f(t - t_A - \tau_{AY}) - 2\pi f\tau_{AY} + \varphi_A - \varphi_Y]}\, e^{-j[2\pi(f+\varepsilon_Y)t + \varphi_Y]} \tag{4.43a}$$

$$+ \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} m_{B,k} e^{j\phi_{B,k}} e^{j[2\pi k\Delta f(t - t_B - \tau_{BY}) - 2\pi f\tau_{BY} + \varphi_B - \varphi_Y]}\, e^{-j[2\pi(f+\varepsilon_Y)t + \varphi_Y]}, \tag{4.43b}$$

where ε_Y is the receiver CFO and the other parameters are as described in Section 3.2

Rationalized by the assumptions that the transmitters operate at identical carrier frequencies and occupy the subchannels mutually exclusively, and to keep the description tractable, consider the simplified model with a single transmitter-receiver pair. Thus, let X and Y denote the transmitter and the receiver, respectively. The received complex baseband signal waveform can be expressed as

$$r_Y(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} m_{X,k} e^{j\phi_{X,k}} e^{j[2\pi k \Delta f(t - t_X - \tau_{XY}) - 2\pi f \tau_{XY} - 2\pi \varepsilon_Y t + \varphi_X - \varphi_Y]}. \tag{4.44}$$

Now let $r_Y(t)$ be sampled at time instants $t = t_Y + nT/N$ according to $r_Y[n] \triangleq r_Y(t_Y - nT/N)$, where $T = 1/\Delta f$ is the duration of an OFDM symbol and t_Y is the receiver start time from (4.28). Then, the discrete-time samples of the baseband signal are

$$r_Y[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \underbrace{m_{X,k} e^{j\phi_{X,k}} \, e^{j[2\pi k \Delta f(t_Y - t_X - \tau_{XY}) - 2\pi f \tau_{XY} - 2\pi \varepsilon_Y t_Y + \varphi_X - \varphi_Y]}}_{\Lambda_{XY,k}} \, e^{j2\pi(k\Delta f - \varepsilon_Y)nT/N}, \tag{4.45}$$

where the complex constant $\Lambda_{XY,k}$ encapsulates all the discrete-time-independent terms. Adopting the notation of [129] by introducing $\epsilon_Y = \varepsilon_Y T$ for the *normalized* CFO, reduces (4.45) to

$$r_Y[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \Lambda_{XY,k} \, e^{j2\pi n(k - \epsilon_Y)/N}. \tag{4.46}$$

Consider the subcarrier phase estimation of (4.28), which may be approximated by performing a DFT on the discrete time samples $r_Y[n]$. The result of the DFT can be concisely written as

$$R_{Y,k} = \Lambda_{XY,k} C_{X,0} + \sum_{l=0,\, l \neq k}^{N-1} \Lambda_{XY,l} C_{X,l-k}, \tag{4.47}$$

where

$$C_{Y,k} = \frac{1}{N} \sum_{n=0}^{N-1} e^{j2\pi n(k + \epsilon_Y)/N}. \tag{4.48}$$

The first term in (4.47) corresponds to a common phase shift, $C_{X,0}$, which causes the constant rotation of the signal. This rotation is closely associated with the *relative CFO* term in (4.42) and has negligible effect on the final *relative* phase offset estimates if the two transmitters are indeed synchronized. However, as the relative CFO between the transmitters becomes non-zero, the signal components r_{AY} and r_{BY} start to drift away. This drift, in turn, may cause significant phase error, especially if the measurement time is extended for averaging, see Section 3.4.2.

The second term of (4.47) characterizes the inter-carrier interference (ICI), a well-known phenomenon in OFDM systems, that destroys the orthogonality between subcarriers. This ICI significantly degrades the phase estimation performance unless accounted for by proper CFO compensation. Note that even with the two transmitters properly synchronized to each other, ICI is still observed unless the receiver is also synchronized to the transmitters. However, if any CFO persists between the transmitters then it is not possible to completely eliminate the ICI at the receiver.

While the single-carrier approach does not require synchronization between transmitter and receiver, its deliberately introduced CFO between the transmitters prohibits the use of a single common carrier frequency across all nodes. In contrast, with the multi-carrier approach there is no reason not to synchronize both the transmitters and the receivers to the same frequency. Therefore, the carrier of all participating nodes should be tuned to a single common frequency value to eliminate the detrimental effects of CFO on the *relative* phase offset estimation. This approach, in turn, also reduces the TO requirement between the receiver nodes.

A straightforward way to achieve such synchronism is to let one transmitter broadcast its unmodulated subcarriers, and have the other transmitter and all the participating receivers estimate and compensate their CFO. Then an adequately constructed OFDM symbol sequence from the other transmitter, or simply its turn-on time, may serve as the reference point for TO compensation between the receivers.

The literature for TO and CFO compensation in OFDM systems is wide, and preamble-based synchronization methods presented in [130][131][132][133][134] and [135] offer directly applicable solutions for both. Furthermore, since CFO and TO compensation are so fundamental to OFDM communication, existing systems inherently provide some form of these synchronization mechanisms.

3.4.2 Phase Measurement Averaging

When proper carrier frequency synchronization is established, the continuous nature of the transmitted multi-carrier signal of (4.20) allows for further averaging to reduce the effect of the additive noise in $r_Y(t)$. Consider the correlation of the received signal for an extended period of time, say for multiple OFDM symbol periods, as in

$$R_{Y,k}(t_Y) = \frac{1}{MT} \int\limits_{t_Y}^{t_Y+MT} r_Y(t) e^{-j2\pi tk/T} dt \qquad M \in \mathbb{Z}^+, \tag{4.49}$$

where $T = 1/\Delta f$ is the duration of an OFDM symbol, M is the number of symbols over the averaging is performed and $r_Y(t)$ is the received signal corrupted by additive Gaussian noise. Based on (4.28), the direct phase measurement of the i^{th} consecutive symbol can be written as

$$R^i_{Y,k}(t_Y) = \frac{1}{T} \int\limits_{t_Y+iT}^{t_Y+(i+1)T} r_Y(t) e^{-j2\pi tk/T} dt. \tag{4.50}$$

Observe that the average of the M consecutive symbols,

$$R_{Y,k}(t_Y) = \frac{1}{M} \sum_{i=0}^{M-1} R^i_{Y,k}(t_Y), \tag{4.51}$$

essentially equals to (4.49). Therefore, the same low-complexity FFT block can be used on the discrete samples of $r_Y(t)$ to estimate the subcarrier phases of multiple non-overlapping OFDM symbols.

Note that the subcarrier phase difference between the consecutive symbols characterizes the CFO between the receiver and the corresponding transmitter, which may be utilized for synchronization. Moreover, small amounts of CFO, $|\epsilon_Y| \ll 1/(MT)$, are tolerable at the receiver because the subcarriers rotate together, leaving the *absolute* phase offsets unaffected. However, if $MT \cdot 2\pi|\epsilon_Y|$ is close to or greater than π, then (4.51) becomes unsuitable for averaging and either the CFO or the measurement time needs to be reduced, see Figure 31.

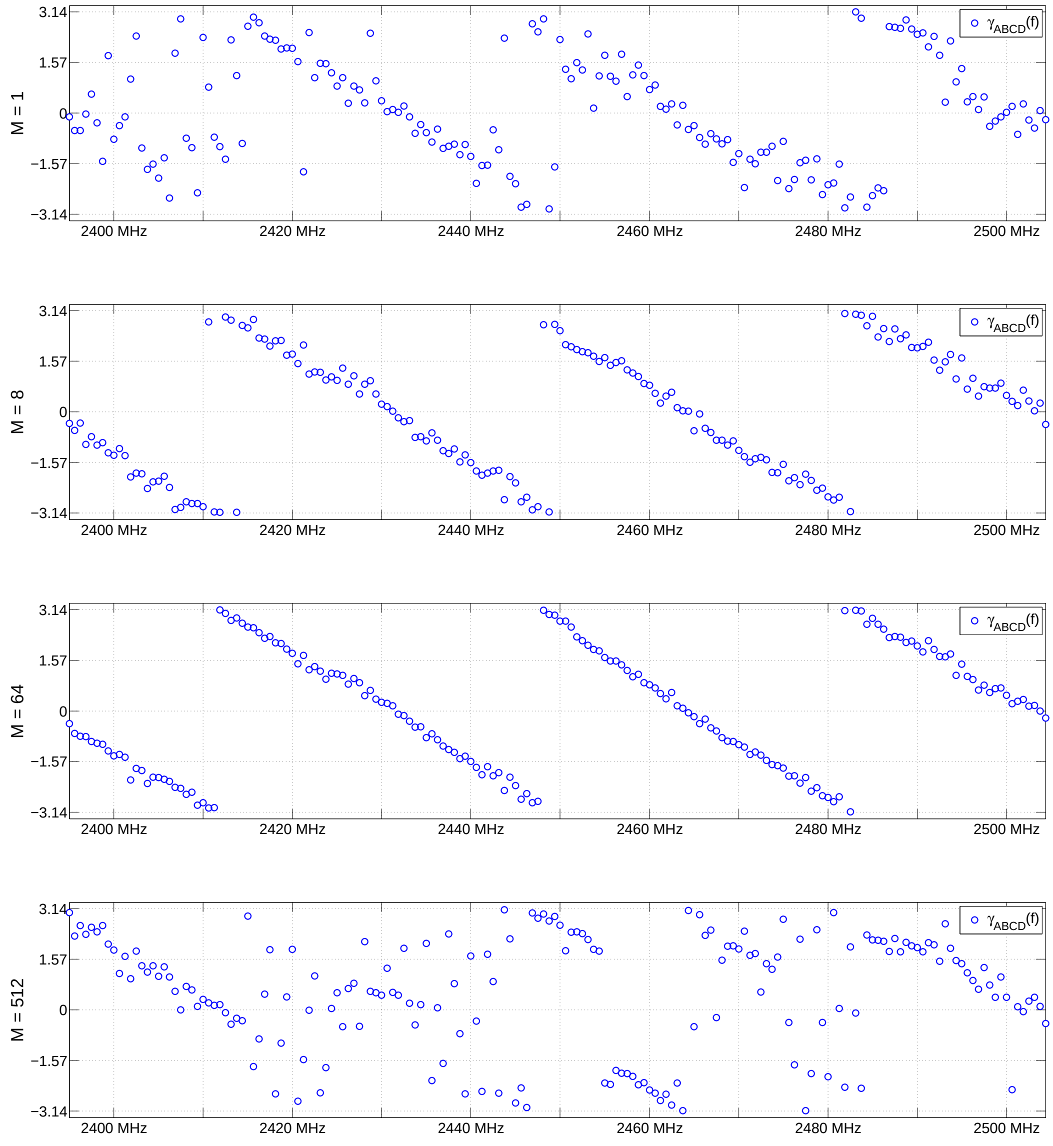


Figure 31: The effect of phase measurement averaging on the final *relative* phase offset estimate for different averaging factors M = 1, 8, 64 and 512. Averaging over multiple symbols improves the estimation performance but breaks down after a certain point due to the lack of synchronization between the transmitters.

3.4.3 Subcarrier Allocation

The way the available subcarriers are assigned to the transmitters has a significant impact on the performance of the phase interpolation algorithm presented in Section 3.3.1. In search for efficient subcarrier allocation maps, the assumptions made so far are reviewed first, followed by further design criteria. Then several possible allocation maps are presented along with their performance comparison.

Assumptions. Throughout the development of the multi-carrier signal model for the basic four-node setup, the following assumptions were used in Section 3.2:

1. Two transmitters operate at the same time
2. A subchannel is assigned to at most one transmitter, see (4.22)
3. At least two subchannels are assigned to a transmitter, see (4.23)
4. N uniformly distributed orthogonal subcarriers are available

Criteria. Even though the final subcarrier assignment explicitly identifies the actual transmitter (C or D), the two transmitters play an equally important role in the phase measurement. Therefore, the structure of the subcarrier mapping is expected to be similar, while the number of the assigned subcarriers identical for the two transmitters.

A further consideration is frequency diversity, which provides a means to combat narrowband interference. In case the interfering subchannels can be identified, their corresponding phase measurement may be discarded, thus, preventing the interpolation and further estimation steps from bias. Therefore, the subcarriers of both transmitters should be spread out diversely over the available bandwidth.

Allocation Strategies Based on the above assumptions and criteria, at most $N/2$ subcarriers are allocated per transmitter. That is, at most $N/2$ raw phase measurements are available to recover the rest of the N subcarrier phases. In the context of Theorem 1, this translates to a frequency estimation based on discrete samples where *the sampling points locations are design parameters*. Clearly, the signs of undersampling are expected in some form, and the following subcarrier allocation examples intend to illustrate the trade-offs between the different approaches.

In the following examples $N = 32$ is used with $N/2 = 16$ subcarriers assigned to both transmitters. Each example is labeled by its a 32-digit binary allocation map, represented as 8-digit hexadecimal numbers for convenience. A binary '1' in the map means that the corresponding subchannel is assigned to one transmitter, while '0' means that it belongs to the other one. The performance of the allocation strategies is compared through their associated periodograms.

- `0xFFFFFFFF`. All subcarriers are assigned to a single transmitter with the sole purpose to serve as a *reference*. The subcarriers are spread out evenly across the entire available bandwidth and no undersampling occurs. The corresponding periodogram in Figure 33(a) exhibits a single narrow and high peak around the true parameter.

- `0xAAAAAAAA (0x55555555).` The even (odd) subcarriers belong to the one (other) transmitter and form a staggered pattern. While this mapping is probably the most straightforward choice to spread out the subcarriers evenly in frequency for both transmitters, it actually represents decimation by a factor of two. This classical example of undersampling results in aliasing in the τ-domain, which is delineated by the two narrow peaks in Figure 33(b). In other words, the subcarriers of a transmitter are spaced $2\Delta f$ apart and ambiguity arises as $\hat{\tau} = \Delta\phi/4\pi\Delta f$ and $\hat{\tau}' = (\Delta\phi + 2\pi)/4\pi\Delta f$ become indistinguishable, see Figure 32. Consequently, *at least one subcarrier pair with at most Δf separation is required per transmitter to avoid undersampling*, or equivalently, τ-domain wrapping.

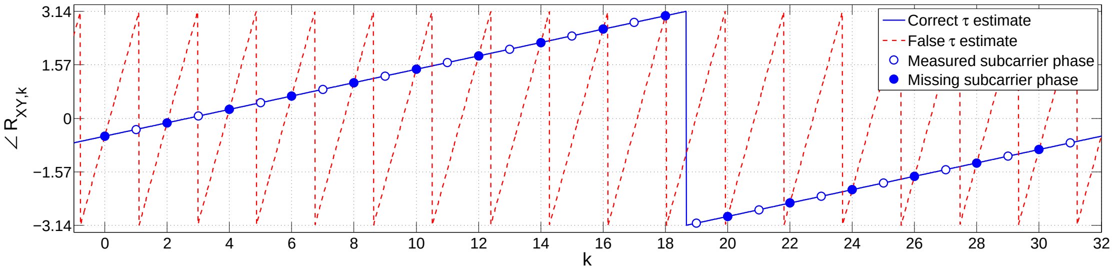


Figure 32: Illustration of the interpolation ambiguity caused by undersampling when subcarrier mapping `0xAAAAAAAA` is used.

- `0xFFFF0000 (0x0000FFFF).` The lower (upper) half of the allocated band is assigned to the one (other) transmitter. This mapping is a radical response to the ambiguity problem, evinced by the alternating pattern `0xAAAAAAAA`, through sacrificing frequency diversity. Figure 33(c) illustrates the corresponding periodogram, which now exhibits a single peak at the right τ-location, albeit of double the width. Unfortunately, the width increase of the peak translates to deteriorated estimation accuracy, which is a direct consequence of the decreased observation length.

- `0xAAAA5555 (0x5555AAAA).` The even (odd) subcarriers of the lower halfband and the odd (even) subcarriers of the upper halfband are associated with one (the other) transmitter. Such mapping offers both frequency diversity and provides means to eliminate undersampling, thus, combines the advantages of mapping patterns `0xAAAAAAAA` and `0xFFFF0000`. Correspondingly, the related periodogram has a null point at the ambiguous τ location of `0xAAAAAAAA` and the width of the main peak equals to that of the reference, see Figure 33(d). On the other hand, two smaller spurious peaks appeared around the null point, which can be distinguished as long the noise level of the subcarrier phase measurements is below a certain threshold.

- `0xCCCCCCCC (0x33333333).` The pairs of adjacent subcarriers are assigned to the one (other) transmitter. This subcarrier allocation strategy is akin to `0xAAAA5555` and provides adequate frequency diversity as well as reasonable protection against undersampling. In terms of periodogram shape, the main difference between the two approaches lays in the location of the two smaller spurious peaks, see Figure 33(e), which now surround the main peak at $\tau = \pi/2\pi\Delta f$ distances.

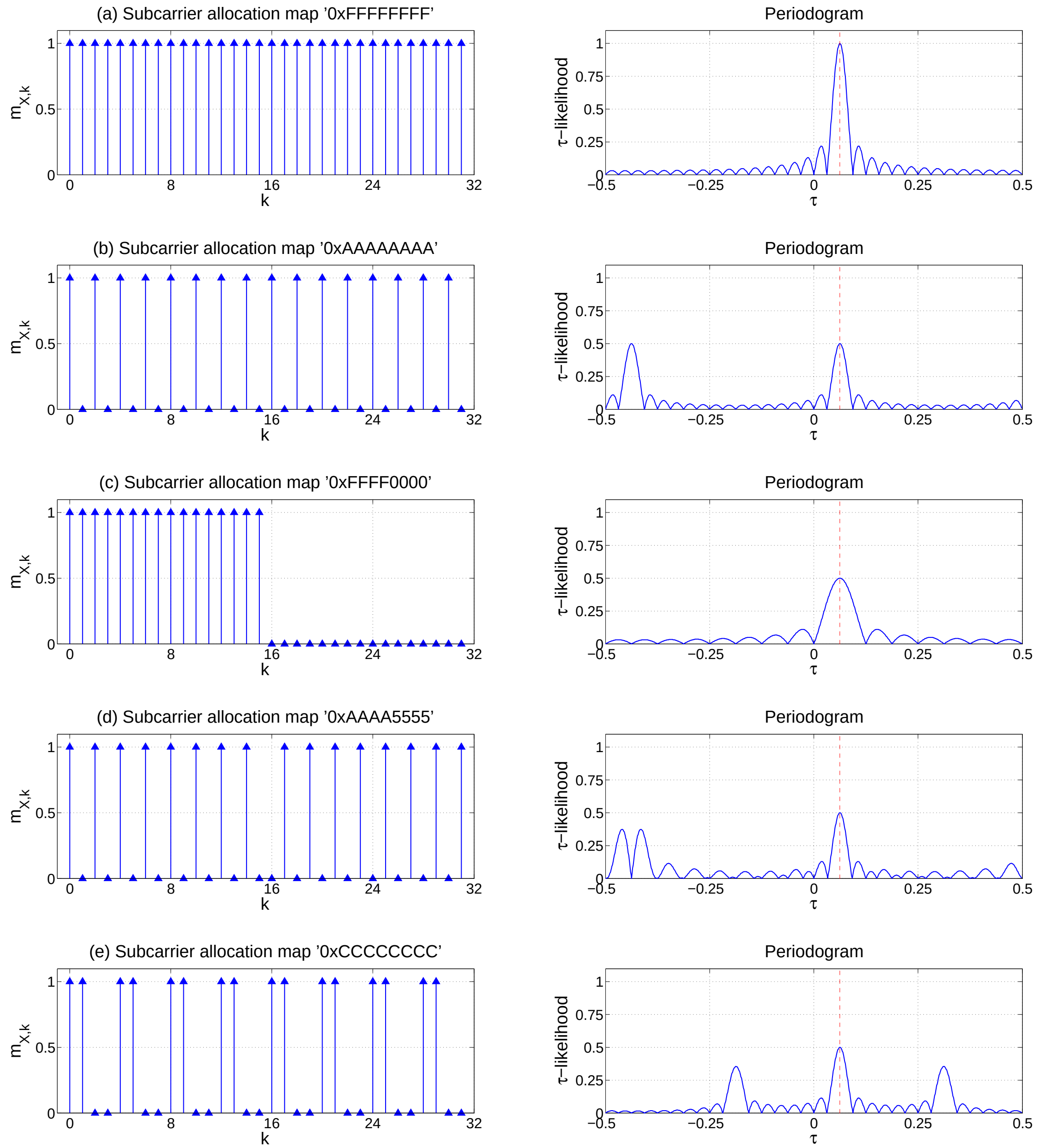


Figure 33: Comparison of various subcarrier allocation maps and their corresponding periodograms.

In summary, the assignment of subcarriers to the transmitters requires careful planning to achieve frequency diversity and avoid ambiguities at the same time. Subcarrier allocation maps `0xAAAA5555` and `0xCCCCCCCC` offer two reasonable choices for the transmitters of the basic four-node setup. The subcarrier allocation for complex setups that rely on more than two simultaneous transmitters should follow the same mapping strategy, however, the analysis of such systems is out of the scope of the current discussion.

3.4.4 Peak-to-Average Ratio Optimization

A fundamental design challenge of multi-carrier communication systems is to keep the peak-to-average ratio (PAR) of the transmitted signal under a certain bound. A high PAR is associated with large peaks in the signal that occur when the various modulated subcarriers of a symbol superimpose constructively in phase. The large peaks distort the signal through clipping, which, in turn, degrades the performance of the communication system. Due to its practical significance, several methods have been proposed to reduce the PAR in OFDM systems [93, p. 759].

The same problem arises with the complex waveforms of (4.20), albeit with relaxed constraints as the *same* symbol is being transmitted repeatedly. To illustrate the problem, observe that the reference case with $m_{X,k} \triangleq 1$ and $\phi_{X,k} \triangleq 0$ for $\forall k$ is

$$|u_X(t)| = \left| \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{j2\pi kt/T} \right| = \frac{1}{\sqrt{N}} \frac{\sin(N \cdot t/T)}{\sin(t/T)} \tag{4.52}$$

which defines a series of sharp peaks that occur with a period of $T = 1/\Delta f$. Note that with $N \to \infty$ the absolute value of $m(t)$ converges to the impluse-train

$$\lim_{N\to\infty} |u_X(t)| = \frac{1}{\sqrt{N}} \sum_{k=-\infty}^{\infty} \delta_D(t - kT) \circledast 1, \tag{4.53}$$

where $\delta_D(t)$ is the Dirac-impulse and $\circledast$ denotes convolution.

Since the $M_{X,k} = m_{X,k} e^{j\phi_{X,k}}$ complex coefficients of the transmitted symbol are fixed for the duration of the phase measurement, a simple PAR reduction approach is to assign pre-calculated phase shifts to the subcarriers. Using vector notation for m_k and ϕ_k as

$$\mathbf{m}_X = [m_{X,0}, \ldots, m_{X,N-1}]^T \qquad \mathbf{p}_X = [\phi_{X,0}, \ldots, \phi_{X,N-1}]^T, \tag{4.54}$$

the problem of finding the phase shifts can be phrased as follows.

Given a subcarrier allocation $\mathbf{m}_X$, find the corresponding $\hat{\mathbf{p}}_X$ symbol phase vector that minimizes the maximum peak of the $u_X(t)$ waveform

$$\hat{\mathbf{p}}_X = \arg\min_{\mathbf{p}} \left\{ \max \left[\text{abs} \left(\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} m_{X,k}\, e^{j\phi_{X,k}} e^{j2\pi k\Delta f t} \right) \right] \right\}, \tag{4.55}$$

where $\hat{\mathbf{p}}_A$ and $\hat{\mathbf{p}}_B$ need to be calculated separately for the two transmitters based on $\mathbf{m}_A$ and $\mathbf{m}_B$, respectively. According to constraint (4.22) a subcarrier is assigned to only one transmitter, therefore, the useful elements of $\hat{\mathbf{p}}_A$ and $\hat{\mathbf{p}}_B$ may be stored in a single $\hat{\mathbf{p}}$ vector according to

$$\phi_k = \begin{cases} \phi_{A,k} & \text{if } m_{A,k} \neq 0 \\ \phi_{B,k} & \text{if } m_{B,k} \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.56}$$

Again, note that the ϕ_k phase shifts are *a priori* calculated design parameters.

Clearly, if PAR reduction is addressed through constant phase shifts at the transmitter, then the same phase rotations appear in the received subcarriers, see (4.28). Since $\phi_{A,k}$ and $\phi_{B,k}$ are design parameters that serve only to lower the PAR, they can be immediately removed after reception by simple complex multiplication. Then, the *aligned* subcarrier phases can be written as

$$\angle R'_{AY,k}(t) = 2\pi k\Delta f(t - t_A) - 2\pi(f + k\Delta f)\tau_{AY} + \varphi_A - \varphi_Y \pmod{2\pi} \tag{4.57a}$$

$$\angle R'_{BY,k}(t) = 2\pi k\Delta f(t - t_B) - 2\pi(f + k\Delta f)\tau_{BY} + \varphi_B - \varphi_Y \pmod{2\pi}, \tag{4.57b}$$

where the received phase is now linear in k subject to a modulo 2π ambiguity.

4 Maximum-Likelihood Distance Estimation

4.1 Overview

The *relative* phase offsets obtained by the multi-carrier measurement method described in Section 3, or by some other means [9], contain the important dTDOA information regarding the spatial relationship between the sensor nodes in a form that is ordinarily unsuitable for the localization algorithms. The goal of distance estimation, the second part of the *measurement* stage, is to convert the *relative* phase offsets into a useful and concise distance metric that can be directly fed into existing localization algorithms.

The distance estimation problem can then be formulated as the precise determination of the

$$d_{ABCD} = -d_{BD} + d_{AD} + d_{BC} - d_{AC} \tag{4.58}$$

unambiguous quad-range, see Figure 27, from the *relative* phase offset estimates

$$\gamma_{ABCD}(f_k) = 2\pi f_k \left(-\tau_{BD} + \tau_{AD} + \tau_{BC} - \tau_{AC}\right) + \varsigma_k \pmod{2\pi}. \tag{4.59}$$

obtained at multiple f_k frequencies, where $\tau = d/c$ is the path delay between the respective sensor nodes, c is the speed of light and ς_k is the phase measurement noise.

Clearly, the challenge is presented by the presence of noise and the modulo 2π phase wrapping. With properly constructed phase measurements, the solution for both comes through the absolute time-independence of the *relative* phase offset, see (4.59). Since $\gamma_{ABCD}(f_k)$ is not a function of the absolute time, it may be independently estimated for a given node quadrant multiple times. Performed on the same f_k frequency promotes noise reduction through averaging, while obtained on different f_k frequencies provides a means to resolve the module 2π ambiguity.

Existing solutions follow a similar approach in that they first associate the $\gamma_{ABCD}(f_k)$ estimates with the corresponding λ_k wavelengths. Then, for a set of K measurements on frequencies f_k the problem of unambiguous quad-range estimation is generally rephrased as a collection of equations

$$d_{ABCD} = \lambda_k \left[n_k + \frac{\gamma_{ABCD}(f_k)}{2\pi}\right] \qquad 1 \le k \le K, \tag{4.60}$$

where $\lambda_k = c/f_k$ is the wavelength and n_k is some integer. The straightforward interpretation of (4.60) is that the d_{ABCD} distance is decomposed into the sum of n_k complete wavelengths plus a $\gamma_{ABCD}(f_k)/2\pi$ fractional wavelength. The unambiguous d_{ABCD} estimate may then be calculated based on the set of n_k values that minimize some error measure defined with (4.60).

The original RIPS [9] arrives at a single hard-decided d_{ABCD} estimate based on the error function

$$\sqrt{\sum_{k=1}^{K}\left(d_{ABCD}-\lambda_k\left[n_k+\frac{\gamma_{ABCD}(f_k)}{2\pi}\right]\right)^2}. \tag{4.61}$$

Recent theoretical works, based on the Chinese Remainder Theorem [136] and Lattice theory [114], consider (4.60) as the starting point and present an alternative solution to (4.61). In contrast, the SRIPS [8] calculates the d_{ABCD} error *distributions*, and processes them with a specialized and more sophisticated localization algorithm.

Common in the existing *relative* phase offset-based distance estimation algorithms is that all explicitly rely on the wavelengths through (4.60). A conceptually different approach is introduced in Section 4.2, which disregards the *absolute* value of the carrier frequencies (wavelengths) and operates exclusively on the *relative* phase offsets and the *relative* distance of their corresponding carrier frequencies. This approach also serves as a reference for the second proposed method, presented in Section 4.3, which now makes use of the *absolute* carrier frequency values and suggests an accurate and elegant solution to the original problem. Finally, the performance of both approaches is analyzed in Section 4.4 in terms of limitations and attainable accuracy.

4.2 Relative Carrier Approach

The following discussion of the *absolute* carrier approach treats (4.59) as a dTDOA estimation problem and deliberately suppresses the presence of the wavelength. That is, even though $\lambda = c/f$ clearly defines the connection between wavelength and carrier frequency, the main focus falls onto the linear relationship

$$\tau_{ABCD} \sim \frac{\partial \widetilde{\gamma}_{ABCD}(f)}{\partial f}, \tag{4.62}$$

where $\widetilde{\gamma}$ is the *unwrapped* relative phase offset. Observe that the slope $\partial\widetilde{\gamma}/\partial f$ carries *all* the information needed to unambiguously estimate (4.58) and is, therefore, independent of the actual value of the carrier frequency. The same idea is illustrated in Figure 34.

Clearly, the challenge lays in the unwrapping of the *relative* phase offset in (4.59), which is generally addressed by obtaining it at multiple frequencies first. Furthermore, since the $\gamma(f)$ measurements are corrupted by noise, intuition tells that the slope estimation accuracy may also benefit from the use of multiple measurement points.

Theorem 2. *The problem of finding $\partial\gamma/\partial f$ is equivalent to estimating the frequency of a complex sinusoid with* ***unknown*** *phase and additive phase noise from a discrete set of observations.*

Proof. Consider Figure 34 and the complex sinusoid constructed based on the *relative* phase offset

$$e^{j[2\pi(f_0+\Delta f_k)\tau_{ABCD}+\phi_0+\varsigma_k]} \qquad 0 \le k < K, \tag{4.63}$$

where f_0 is some carrier frequency, Δf_k is the relative frequency with respect to f_0, ϕ_0 is the unknown phase at f_0 and ς_k is the phase noise. Then, by swapping the interpretation of f and τ_{ABCD}, the problem can be rephrased as the estimation of the τ_{ABCD} *frequency* of the complex sinusoid from its discrete observations taken at $f_0 + \Delta f_k$ *time* points. □

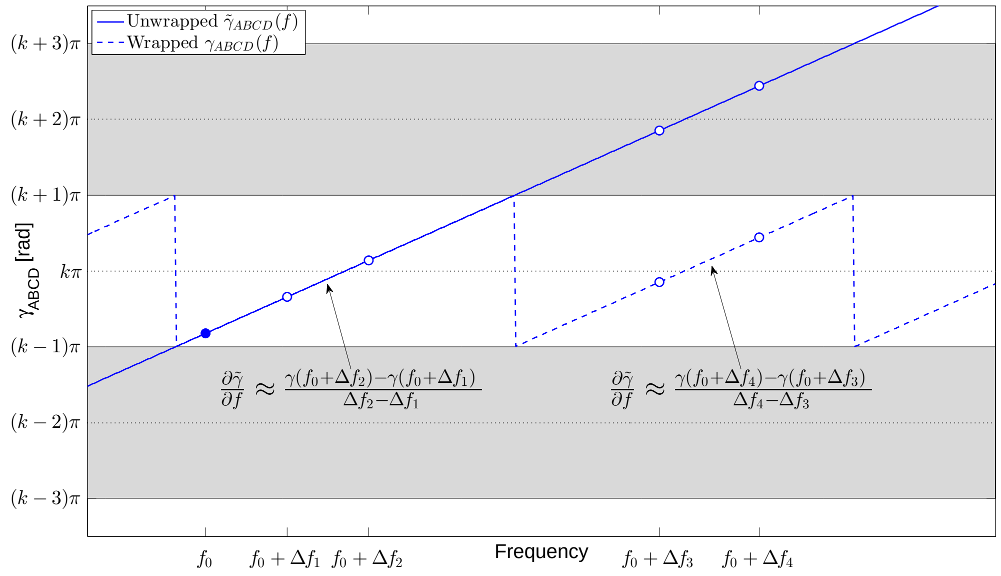


Figure 34: The d_{ABCD} range is uniquely defined by the slope $\partial\tilde{\gamma}/\partial f$, regardless of the absolute carrier frequency value at which the *relative* phase offset is obtained.

The signal description of (4.63) is often used as an approximate signal model for frequency estimation problems in high signal-to-noise ratio (SNR) scenarios [119][120]. However, when a larger range of SNRs is considered, the signal is commonly modeled as complex sinusoid in noise,

$$e^{j[2\pi(f_0+\Delta f_k)\tau_{ABCD}+\phi_0]} + W_k \qquad 0 \leq k < K, \tag{4.64}$$

where W_k are assumed to be complex white Gaussian noise (CWGN) samples with zero mean and variance σ_W. As discussed in Section 3.3.1, frequency estimation is a well studied problem that has a wide literature, see [119],[120],[121],[122],[123],[124],[125],[126],[127] and [128].

Observe that (4.64) is a completely valid model for the *relative* phase offset estimates when the channels between the static nodes are assumed to be Gaussian with no multipath. Moreover, the multi-carrier phase estimation approach, described in Section 3, promotes the use of (4.64) in several ways. First, a single multi-carrier phase measurement yields $K = N$ different *relative* phase estimates simultaneously. The output of multiple such multi-carrier estimates may also be stitched together to further increase the number and diversity of the observation points. Second, the $R_{XY,k}$ direct phase estimates of (4.28) are complex. Keeping the complex representation for $R_{XY,k}$, $\vartheta_{Y,k}$ and $\xi_{X,k}$, instead of reducing to their angle, both simplifies the intermediate operations and makes the complex notation of γ_k and (4.64) natural. Third, the subcarrier spacing is inherently uniform. Thus, for a single multi-carrier measurement (4.64) becomes

$$e^{j[2\pi(f_0+k\Delta f)\tau_{ABCD}+\phi_0]} + W_k \qquad 0 \leq k < K, \tag{4.65}$$

where the $K = N$ observation points are now placed equidistantly, $\Delta f = 1/T$ apart. Following the frequency estimation analogy of Theorem 1, this corresponds to a uniform sampling rate, which simplifies the estimation problem by allowing the use of frequency estimators that operate specifically on uniformly sampled data.

A reasonable choice for obtaining $\widehat{\tau}_{ABCD}$ is the maximum-likelihood (ML) estimator, derived in Appendix B-3,

$$\widehat{\tau}_{\text{ML}} = \arg\max_{\tau \in \mathcal{T}} \left| \frac{1}{K} \sum_{k=0}^{K-1} e^{j\gamma_k} e^{-j2\pi \Delta f_k \tau} \right|, \tag{4.66}$$

where $\gamma_k = \gamma(f_0 + \Delta f_k)$, and which essentially maximizes the corresponding periodogram. Note that the sole purpose of the f_0 reference carrier is to index the corresponding *relative* phase estimate, and both f_0 and the ϕ_0 unknown phase dropped out from the expression of $\widehat{\tau}_{\text{ML}}$. Furthermore, since the multi-carrier *relative* phase estimates are uniformly spaced in frequency, (4.66) can be written as

$$\widehat{\tau}_{\text{ML}} = \arg\max_{\tau \in \mathcal{T}} \left| \frac{1}{K} \sum_{k=0}^{K-1} e^{j\gamma_k} e^{-j2\pi k \Delta f \tau} \right|, \tag{4.67}$$

where $\gamma_k = \gamma(f_0 + k\Delta f)$. Observe that the latter representation is particularly useful as it allows to calculate the periodogram using FFT.

The proposed *relative* carrier based d_{ABCD} estimator, therefore, starts out with the K available γ_k *relative* phase offset estimates and their corresponding Δf_k sampling locations. The initial search domain for τ is set to the largest unambiguous range $\mathcal{T} = \pm 1/2\Delta f_{min}$, where Δf_{min} denotes the smallest distance between the Δf_k sampling locations, see Section 4.4.1. In case the observations are non-uniformly distributed, an exhaustive search for the location of the global maximum of (4.66) over $\mathcal{T}$ yields the $\widehat{\tau}_{\text{ML}}$ estimate.

However, if the *relative* phase offset estimates are uniformly spread, $\Delta f_{min} = \Delta f$, the initial range translates to $\mathcal{T} = \pm T/2$, where T denotes the OFDM symbol duration. Exploiting the structure of the sample locations, the process incorporates an intermediate FFT-based stage to obtain a coarse $\widehat{\tau}'_{\text{ML}}$ estimate first. To improve the resolution, γ_k should be zero padded to $M = 4K$ samples prior to calculating the FFT for the periodogram [124], see the interpolated periodograms in Figure 37(a). The location of the largest periodogram peak then provides $\widehat{\tau}'_{\text{ML}}$, which is used as the starting point for the fine estimation stage. The fine search for the $\widehat{\tau}_{\text{ML}}$ location of the periodogram global maximum is then performed over the restricted range $\mathcal{T}' = \widehat{\tau}'_{\text{ML}} \pm 1/K\Delta f$, using either grid or iterative search methods. The steps of the distance estimation process are summarized in Algorithm 4.

The key aspect of the *relative* carrier approach is that it focuses on the $\tau_{ABCD} \sim \partial\gamma/\partial f$ relationship only, therefore, the *relative* phase offset observations become independent of the reference carrier frequency, f_0. Consequently, unlike in the existing methods and in Section 4.3, the corresponding wavelengths play no role in the estimation. The performance of the approach is analyzed in Section 4.4.

```
Data: $\gamma_{ABCD}(f_0+\Delta f_k)$ relative phase offsets and corresponding $\Delta f_k$ for $0 \le k < K$
Result: $\widehat{d}_{ABCD}$ estimate
begin
    if $\Delta f_k = k\Delta f \quad 0 \le k < K$ then
        $\mathcal{T} = 1/2\Delta f$
        // Coarse estimate
        Pad $\gamma_k$ with zeros to $M = 4K$ total samples
        Calculate $\widehat{\tau}'_{\text{ML}}$ using M-point FFT and (4.66) over $\mathcal{T}'$
        // Fine estimate
        $\mathcal{T}' = (\widehat{\tau}'_{\text{ML}} - 1/K\Delta f, \widehat{\tau}'_{\text{ML}} + 1/K\Delta f)$
        Calculate (4.67) over $\mathcal{T}'$ to obtain $\widehat{\tau}_{\text{ML}}$
    else
        $\Delta f_{min} = \min(|f_i - f_j|) \quad i \ne j \quad 0 \le i,j < K$
        $\mathcal{T} = 1/2\Delta f_{min}$
        Calculate (4.66) over $\mathcal{T}$ to obtain $\widehat{\tau}_{\text{ML}}$
    end
    Calculate $\widehat{d}_{ABCD} = c \cdot \widehat{\tau}_{\text{ML}}$
end
```

Algorithm 4: Distance estimation based on multiple *relative* phase offsets with relying only on the relative distance between the corresponding carrier frequencies.

4.3 Absolute Carrier Approach

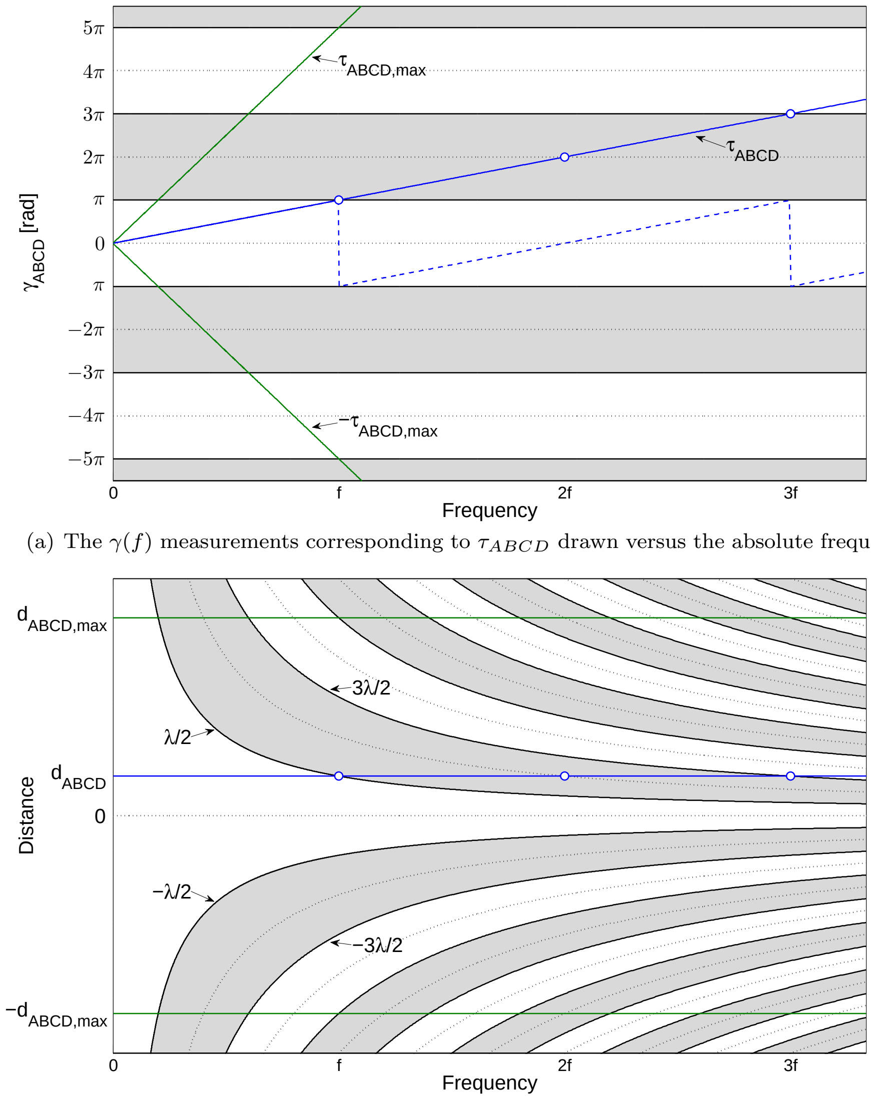
The elaboration of the *absolute* carrier approach follows a similar path as Section 4.2 and purposely suppresses the wavelength concept for most of the discussion by handling (4.59) as a DTDOA estimation problem. That is, the precise determination of the slope $\tau_{ABCD} \sim \partial\gamma/\partial f$ remains of central interest, while making an attempt to incorporate the knowledge about the actual carrier frequency into the previous model. Therefore, though the λ wavelength is never explicitly used in the model, its interpretation in the context is inevitable due to its direct $\lambda = c/f$ relationship with the carrier frequency, and because its implicit appearance in the final results.

As previously, the primary challenge is in unwrapping the *relative* phase estimates of (4.59) in the presence of noise. The implicit approach of Section 4.2 is to estimate the phase shifts of γ with respect to the *relative* distances between carriers, as illustrated in Figure 34. Now consider the relationship of the unwrapped $\widetilde{\gamma}$ *relative* phase offset and the *absolute* carrier frequency in Figure 35(a), along with that of the corresponding $d = c \cdot \tau = \lambda \cdot \widetilde{\gamma}/2\pi$ distances in Figure 35(b). Observe that a given d_{ABCD} range is represented by the slope of a line in the former, and as the position of a horizontal line in the latter. Figure 35(b) suggests that for any d_{ABCD} distance there exists a carrier frequency limit, under which it falls into the $\pm\lambda/2$ range, and can be unambiguously calculated from a single $\gamma(f)$ observation. While this generally leads to prohibitively low carrier frequencies in practice, it has an interesting implication. Visually, as $d = \pm\lambda/2$ in Figure 35(b) always corresponds to $\gamma = \pm\pi$ in Figure 35(a), it follows that

$$\lim_{f\to 0} |\pm\lambda/2| = \infty \quad \Rightarrow \quad \lim_{f\to 0} \gamma(f) = 0. \tag{4.68}$$

That is, any finite d_{ABCD} distance leads to zero *relative* phase offset as the *absolute* carrier frequency approaches zero.

A possible interpretation of (4.68) in the context of the *relative* carrier approach is that an



(a) The $\gamma(f)$ measurements corresponding to τ_{ABCD} drawn versus the absolute frequency.

(b) The $d_{ABCD} = c \cdot \tau_{ABCD}$ distance illustrated on the absolute frequency scale.

Figure 35: The relationship between $c \cdot \tau_{ABCD}$, the γ relative phase offset and the λ wavelength.

additional *relative* phase offset observation is provided, at a relative distance of f_0 with respect to the first measurement point, for *free*. However, since $\gamma(0) \triangleq 0$ is not subject to corruption by phase noise, (4.68) has a stronger implication as summarized in Theorem 3.

Theorem 3. *The problem of finding $\partial\gamma/\partial f$ and incorporating knowledge about the absolute carrier frequencies is equivalent to estimating the frequency of a complex sinusoid with* ***known*** *phase and additive phase noise from a discrete set of observations.*

Proof. Consider Figures 35(a) and 35(b), and the complex sinusoid constructed based on the *relative* phase offset

$$e^{j[2\pi f_k \tau_{ABCD} + \phi_0 + \varsigma_k]} \qquad 0 \leq k < K, \tag{4.69}$$

where f_k is the carrier frequency, ϕ_0 is the known phase and ς_k represents the phase noise. From (4.68) it follows that

$$\gamma(0) \triangleq 0 \quad \Rightarrow \quad \phi_0 \triangleq 0, \tag{4.70}$$

which is also a consequence of the original definition, see (4.59). Then, by swapping the interpretation of f and τ_{ABCD}, the problem can be rephrased as the estimation of the τ_{ABCD} *frequency* of the zero-phase complex sinusoid from its discrete observations taken at f_k *time* points. □

Note that Theorem 3 portrays a rather peculiar estimation problem as it is the *unknown frequency* to be estimated with the *phase known*. This is rarely the case with practical communication systems, since the phase of the incoming signal is almost never known. The other way around, however, with the *frequency* known and the *phase* to be estimated, is a classical receiver design problem for phase modulated signals.

The phase noise representation in (4.69) is generally used as the low-SNR approximation of a more suitable signal model, the single-tone complex sinusoid in additive noise [119][120]. Therefore, following the strategy of Section 4.2, let (4.69) be replaced with the approximate signal model

$$e^{j2\pi f_k \tau_{ABCD}} + W_k \qquad 0 \leq k < K, \tag{4.71}$$

where W_k are assumed to be complex white Gaussian noise (CWGN) samples with zero mean and variance σ_W.

Note that (4.71) adequately represents the *relative* phase offset measurements as long as the sensor nodes are static and the pairwise channels are assumed to be multipath-free and Gaussian. Furthermore, when the complex representation of $R_{XY,k}$, $\vartheta_{Y,k}$ and $\xi_{X,k}$ is retained throughout the multi-carrier phase measurement, instead of only the angles, (4.71) provides a more accurate description than (4.69). For the multi-carrier *relative* phase estimates of Section 3, the model becomes

$$e^{j2\pi(f_0+k\Delta f)\tau_{ABCD}} + W_k \qquad 0 \leq k < K, \tag{4.72}$$

where $f_0 < \Delta f$, and the K observation points are placed equidistantly, $\Delta f = 1/T$ apart.

Consider the maximum-likelihood estimator of τ_{ABCD} for the *known* phase case, derived in Appendix B-3,

$$\widehat{\tau}_{\mathrm{ML}} = \arg\max_{\tau \in \mathcal{T}} \sum_{k=0}^{K-1} \mathrm{Re}\left[e^{j\gamma(f_k)} e^{-j2\pi f_k \tau}\right], \tag{4.73}$$

where $\mathcal{T} = 1/\Delta f_{min}$ is the unambiguous search range defined by the smallest distance between the carriers. That is, $\widehat{\tau}_{\mathrm{ML}}$ is essentially obtained by correlating the *relative* phase offset samples with the conjugate of the zero-phase complex sinusoid in (4.71), searching for the location of the global maximum over the range $\mathcal{T}$.

Acknowledging that both the *relative* and *absolute* carrier approaches intend to precisely determine the slope $\partial\gamma/\partial f$, the proposed algorithm takes advantage of both. The former is exploited to obtain a coarse $\widetilde{\tau}_{ABCD}$ estimate and to restrict the $\mathcal{T}$ search range to $\mathcal{T}'$, see Figure 37(b). The latter method then refines the estimate by finding the maximum of (4.73) over $\mathcal{T}'$. Observe that since the estimation model embodies information regarding the *absolute* carrier frequencies, the corresponding likelihood function exhibits peaks at approximately wavelength distances. Further analysis of the estimation process is given in Section 4.4 and its steps are summarized in Algorithm 5.

```
Data: $\gamma_{ABCD,k}$ relative phase offsets for $\forall k : 0 \leq k < K$
Result: $\widehat{d}_{ABCD}$ estimate
begin
    // Coarse estimate
    Calculate $\widehat{\tau}'_{\mathrm{ML}}$ using Algorithm 4
    // Fine estimate
    $\mathcal{T}' = (\widehat{\tau}'_{\mathrm{ML}} - \varepsilon, \widehat{\tau}'_{\mathrm{ML}} + \varepsilon)$
    Calculate (4.73) over $\mathcal{T}$ to obtain $\widehat{\tau}_{\mathrm{ML}}$
        $\widehat{\tau}_{\mathrm{ML}} = \arg\max_{\tau \in \mathcal{T}'} \sum_{k=0}^{K-1} \mathrm{Re}\left[e^{j\gamma(f_k)} e^{-j2\pi f_k \tau}\right]$
    Calculate $\widehat{d}_{ABCD} = c \cdot \widehat{\tau}_{\mathrm{ML}}$
end
```

Algorithm 5: Distance estimation based on multiple *relative* phase offsets and incorporating knowledge about the *absolute* carrier frequencies values into the estimation model.

4.4 Performance Analysis

Both the *relative* and *absolute* carrier distance estimation approaches have inherent limitations that are primarily attributed to carrier frequency arrangement and measurement noise. The goal of this section is to analyze the ML estimator of the two approaches, (4.66) and (4.73), in order to identify such limitations and their relationship with various frequency allocation strategies for *relative* phase offset measurements. Section 4.4.1 examines the ambiguities that set limitations to the maximum uniquely determinable d_{ABCD} range, while Section 4.4.2 assess the accuracy of the ML estimators by comparing the simulated variance curves to the corresponding Cramér-Rao bounds. Interpreting the results in the context of a multi-carrier measurement, it is shown that the two leads to slightly contradictory design requirements.

4.4.1 Ambiguity

The distance estimators of both Section 4.4.1 and 4.4.2 rely on the precise determination of the $\tau_{ABCD} \sim \partial\gamma/\partial f$ slope from a finite set of γ discrete observations, that are wrapped mod 2π. Therefore, the largest uniquely resolvable d_{ABCD} is also associated with steepest unambiguously distinguishable slope.

Since the $\partial\gamma/\partial f$ slope has to be estimated from discrete $\gamma(f_k)$ observations at finite distances, ambiguity arises if the *relative spacing* of the f_k observation locations is inadequate to distinguish between the true $\partial\widetilde{\gamma}/\partial f$ and some other $(\partial\gamma + k2\pi)/\partial f$ slope, where $\widetilde{\gamma}$ is the unwrapped *relative* phase offset and $k \in \mathbb{Z}$. Note that a similar issue is addressed during the development of the subcarrier allocation strategies in Section 3.4.3 for uniformly spaced subcarriers.

In the general case, an upper bound may be established for the largest unambiguous τ_{ABCD} based on the minimum relative distance between the carriers according to

$$\tau_{max} = \frac{1}{\Delta f_{min}} = \frac{1}{\min\limits_{k,l} |f_k - f_l|}, \quad \text{where} \quad k \neq l,\ 0 \leq k, l < K \tag{4.74}$$

and K is the number of observation points. Note, however, that a bound lower than (4.74) may still exist, as suggested by the subcarrier allocation pattern `0xAAAA5555` in Section 3.4.3.

In contrast, an exact upper bound is available when the $\gamma(f_k)$ observations are uniformly spaced in frequency. Consider the *relative* phase offset estimates of a multi-carrier measurement with N orthogonal subcarriers spaced Δf apart. Then $\Delta f_{min} = \Delta f$ and (4.74) becomes

$$\tau_{max} = \frac{1}{\Delta f} = T, \tag{4.75}$$

where T is the duration of a single OFDM symbol. Observe that the smaller the Δf subcarrier spacing, the longer the OFDM symbol, and the larger the unambiguous $d_{ABCD} = c\tau$ range, see Figure 36. In an FFT-implemented OFDM design the elongation of the symbol may be achieved by increasing the FFT point-size or by reducing the sampling frequency, which provides flexibility when adjusting the complexity of the design.

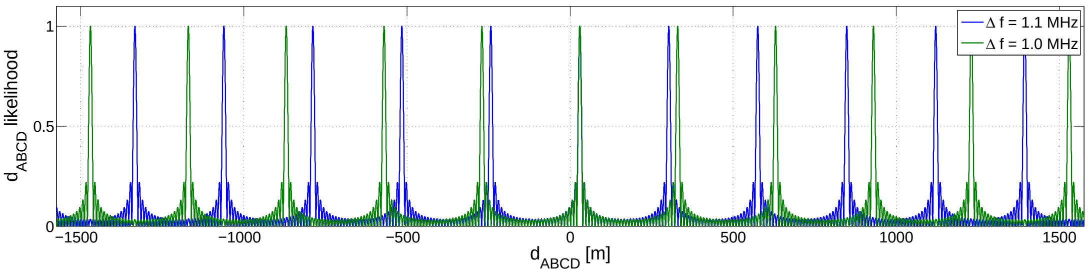


Figure 36: For equidistantly sampled *relative* phase offsets the Δf subcarrier spacing directly determines the unambiguously resolvable d_{ABCD} range. Smaller subcarrier spacing provides larger uniquely resolvable d_{ABCD} distance ($N = 32$).

The physical node arrangement also poses a constraint on the maximum d_{ABCD} distance. The path difference between transmitter X and receivers C and D is the largest, $|d_{XD} - d_{XC}| \leq d_{CD}$, when the three nodes are located collinear. Then, it follows that $|d_{ABCD}| \leq 2 \cdot d_{CD}$, which suggests that the distances between the participating nodes can be used as a reference for τ_{max}, hence for the OFDM design parameters.

In summary, the distances between the $\gamma(f_k)$ observation samples, the *relative* carrier spacing, determines the largest unambiguously resolvable d_{ABCD} range for both the *relative* and *absolute* carrier approaches. *As the $d_{ABCD,max}$ limit increases with the subcarrier denseness, it is desirable to have $\gamma(f_k)$ measurement points close to each other in frequency.* This should serve as a general carrier allocation rule regardless of the underlying phase measurement technique.

4.4.2 Bandwidth and Carrier Frequency

In an ideal environment, free from multipath propagation and noise, two closely spaced *relative* phase offset observations are sufficient to determine the $\partial\gamma/\partial f$ slope and resolve arbitrarily large unambiguous d_{ABCD} distances, as discussed in Section 4.4.1. In realistic scenarios, however, the observations are corrupted by some degree of noise and the number of measurement points, along with their arrangement, largely affect the accuracy of the distance estimation. This section analyses the impact of the measurement bandwidth and *absolute* carrier frequency on the distance estimation performance by comparing simulation results to the derived Cramér-Rao bounds.

Bandwidth. A straightforward way to define the bandwidth used for the *relative* phase offset measurements is by $B = f_{\max} - f_{\min}$, the difference between the largest and smallest carrier frequencies involved. In case all the observations are obtained through a single OFDM phase measurement with every the subcarriers utilized, this translates to $B = N\Delta f$, where N is the FFT point-size and $\Delta f = 1/T$ is the subcarrier spacing.

Due to the time-independence emphasized in (4.31), independent *relative* phase offset measurements taken at different carrier frequencies for the same node setup may be stitched together to increase the overall bandwidth. Such expansion of the bandwidth is clearly inevitable in the single-carrier case, and may also be exploited when the observations are obtained through the multi-carrier phase estimation of Section 3.

For the *relative* carrier approach, the effect of the change in measurement bandwidth is illustrated in Figure 37(a). With increasing bandwidth the peaks of the (4.67) likelihood function,

$$\left| \frac{1}{K} \sum_{k=0}^{K-1} e^{j\gamma_k} e^{-j2\pi k \Delta f \tau} \right| \tag{4.76}$$

become narrower, and actually converge to the impulse series of (4.53). Note that the periodicity of the peaks is suppressed in Figure 37(a) by keeping Δf constant, consequently increasing the number of observation points, as discussed in Section 4.4.1. Such definiteness of the periodogram is highly desirable as the location of the peak maximum corresponds to the global maximum over one unambiguous d_{ABCD} range. Therefore, a narrow peak permits less variation in the location of its maximum, especially when significant noise is present and the shape of the periodogram is distorted.

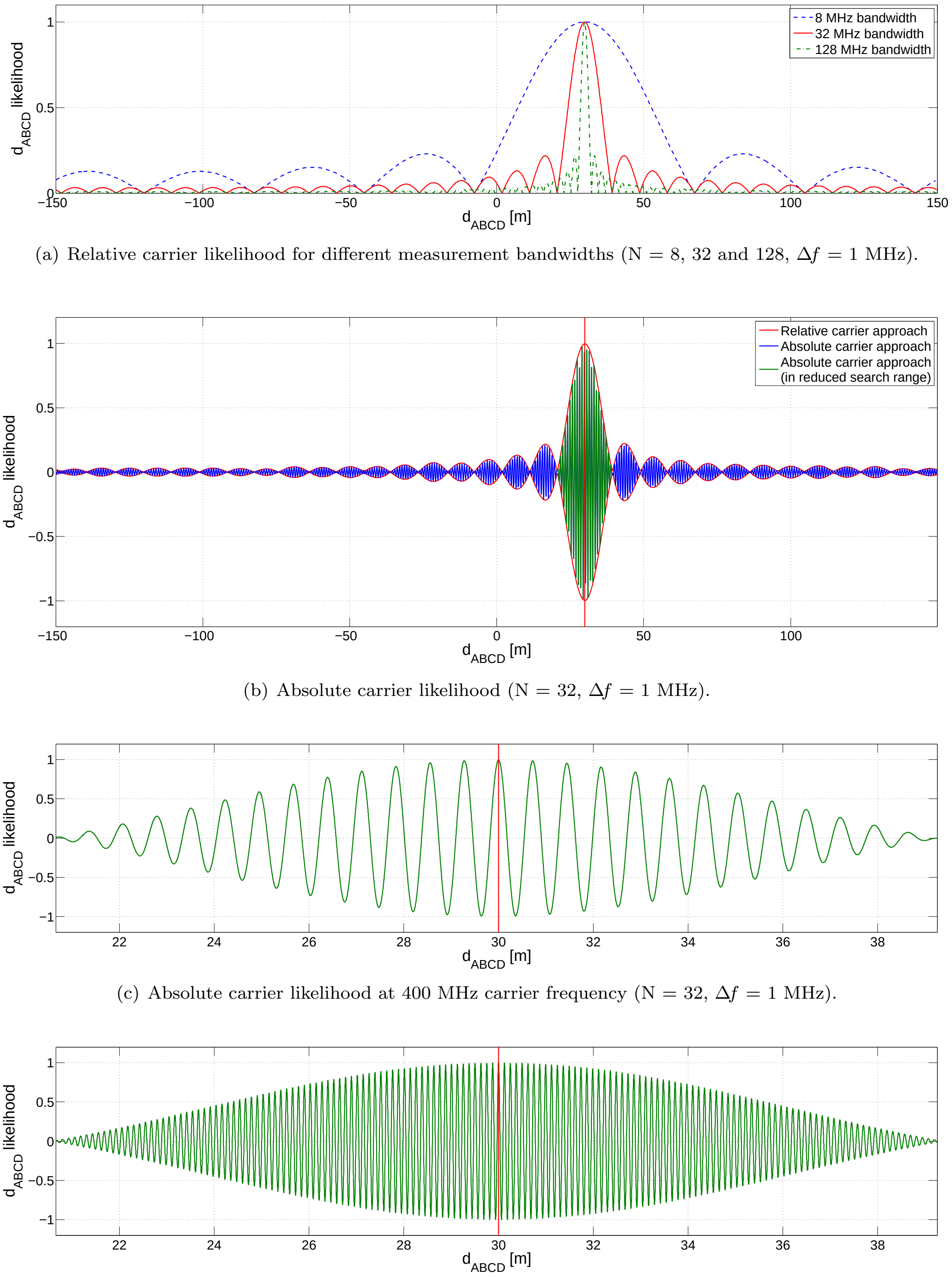
Carrier Frequency. Associating a *single* carrier frequency with a set of *relative* phase offset measurement is less straightforward by definition. A characteristic carrier frequency may be defined as the median or some mean of the f_k frequencies involved. Instead, however, the following discussion always assumes a $f_k = f_0 + k\Delta f$ block of measurements, and refers to $f_0 = f_{\min}$ as the characteristic carrier frequency.

Consider the estimator of the *absolute* carrier approach and compare the shape of its likelihood function,

$$\frac{1}{K} \sum_{k=0}^{K-1} \mathrm{Re}\left[e^{j\gamma(f_k)} e^{-j2\pi f_k \tau} \right], \tag{4.77}$$

in Figure 37(b) with that of (4.76) in Figure 37(a). Observe that the former shows significantly more transitions as a function of d_{ABCD}, while the latter provides an envelope for the magnitude. A closer look at the likelihood function, evaluated for two different f_0 carrier frequencies in the vicinity of its global maximum, suggests a clear relationship with the corresponding wavelengths, see Figures 37(c) and 37(d). Indeed, the distances of the first side-peaks are 72 cm and 12 cm, respectively, which are approximately the wavelength of the associated characteristic frequencies, 400 MHz and 2400 MHz. Note that a wavelength dependence akin to (4.60) surfaced even though the λ_k wavelengths were never explicitly used.

The relationship between bandwidth and carrier frequency is now apparent when the enveloping effect of (4.76) is emphasized. In general, larger bandwidth turns the envelope into a narrower pulse, which makes the true global maximum more distinguishable from its side-peaks.



(a) Relative carrier likelihood for different measurement bandwidths (N = 8, 32 and 128, Δf = 1 MHz).

(b) Absolute carrier likelihood (N = 32, Δf = 1 MHz).

(c) Absolute carrier likelihood at 400 MHz carrier frequency (N = 32, Δf = 1 MHz).

(d) Absolute carrier likelihood at 2400 MHz carrier frequency (N = 32, Δf = 1 MHz).

Figure 37: Likelihood functions of the *relative* carrier (a) and *absolute* carrier (b) approaches over a single unambiguous d_{ABCD} range. The latter evaluated over a reduced range for 400 MHz and 2400 MHz carrier frequencies in (c) and (d), respectively.

1. $\mathbf{B} \to \infty$. At one extreme, as the bandwidth $B = N\Delta f$ converges to infinity, the envelope takes the shape of to the (4.53) pulse train, and the peak-width eventually becomes narrower than the wavelength associated with f_0. Therefore, the *absolute* carrier approach provides no additional benefits, especially since the definition of the f_0 characteristic wavelength loses validity.

2. $\mathbf{B} \approx \mathbf{f_0}$. The bandwidth is comparable to the carrier frequency and the width of the envelope peak is c/B. Since the first side-peaks are approximately $\lambda = c/f_0$ away from the global maximum, they are completely suppressed by the envelope. Thus, the *absolute* carrier approach offers negligible advantages compared to the *relative* one.

3. $\mathbf{0} \ll \mathbf{B} \ll \mathbf{f_0}$. The bandwidth is sufficiently high to create a narrow peak in the envelope, but significantly smaller than the carrier frequency. The drop-off of the envelope peak aids the isolation of the global maximum in (4.77) to a certain degree. In the presence of noise, the envelope is distorted, therefore, the selection of bandwidth and carrier frequency represents a design trade-off between the ability to select the proper peak and the accuracy that peak offers. Clearly, Figures 37(c) and 37(d) suggest that higher carrier frequency makes the peak representing the global maximum less distinguishable, but offers improved accuracy when found. In terms of (4.60), this translates to smaller wavelengths providing less accurate n_k estimate, but also smaller fractional error due to $\lambda_k \cdot \gamma(f_k)/2\pi$ in general, see also Figures 35(a) and 35(b).

4. $\mathbf{B} \approx \mathbf{0}$. At the other extreme, the bandwidth is close to zero and the envelope is almost flat. In this case, the envelope offers imperceptible help and the global maximum of (4.77) becomes indistinguishable from the neighboring local maximum.

From a practical viewpoint, the case with $0 \ll B \ll f_0$ is realizable and is of most interest. To characterize the performance dependence of the ML estimators on bandwidth, carrier frequency and measurement noise, let $f_k = (k_0 + k)\Delta f$, where $f_0 = k_0 \Delta f$ is the characteristic carrier frequency. With this notation, a theoretical limit on the attainable accuracy by any unbiased $\widehat{\tau}_{ABCD}$ estimator, the Cramér-Rao lower bound (CRLB), is derived in Section B-2. The Cramér-Rao bound for the *relative* carrier approach is

$$\operatorname{var}(\widehat{\tau}_{\text{rel}}) \geq \frac{6}{\text{SNR}_\gamma (2\pi\Delta f)^2 N(N^2-1)}, \tag{4.78}$$

while that for the *absolute* carrier method is

$$\operatorname{var}(\widehat{\tau}_{\text{abs}}) \geq \frac{1/2}{\text{SNR}_\gamma (2\pi\Delta f)^2 (k_0^2 N + 2k_0 P + Q)}, \tag{4.79}$$

where $P = N(N-1)/2$ and $Q = N(N-1)(2N-1)/6$ are constants that depend only on the total number of observations. Note that SNR_γ is defined on the γ *relative* phase offset observations with respect to its measurement noise and not directly on the received baseband signals. Therefore, the perceived value of SNR_γ may be improved through averaging, see Section 3.4.2.

Non-linear parameter estimation generally exhibits a rapid performance degradation below a certain SNR, which is referred to as the *threshold* effect. Since τ is a non-linear parameter of the complex sinusoid in (4.72), the performance of its estimator is also expected to break down quickly when observed as a function of bandwidth, carrier frequency or SNR. In the following, the Cramér-Rao bounds (4.78) and (4.79) serve as a benchmark, against which the corresponding parametrized ML estimators, (4.66) and (4.77), are compared. The departure of the simulated estimator variances from the CRLB is then used to identify the threshold level and analyze the estimator behavior in the surrounding region.

Carrier Frequency Threshold The carrier frequency dependence of the d_{ABCD} estimator performance is shown in Figure 38. By definition, the *relative* carrier approach is independent of the actual f_0 carrier, therefore, its CRLB (dashed line) is constant. In contrast, the CRLB of the *absolute* carrier approach (solid line) decreases $\sim 1/f_0^2$. Observe that the simulated root mean square (RMS) estimation error closely follows the solid line up to a certain carrier frequency, then abruptly depart, indicating a strong performance degradation. The frequency threshold increases with the bandwidth and the number of samples, being located at approximately 100 MHz for $N = 16$, 400 MHz for $N = 32$, 1400 MHz for 64 and 3000 MHz (not shown) for 128 uniformly spaced, $\Delta f = 1$ MHz, *relative* phase offset observations. Above the carrier frequency threshold, the RMS error converges to the dashed line, which has a severe implication. In this region the *absolute* carrier approach provides no performance gain compared to the *relative* approach.

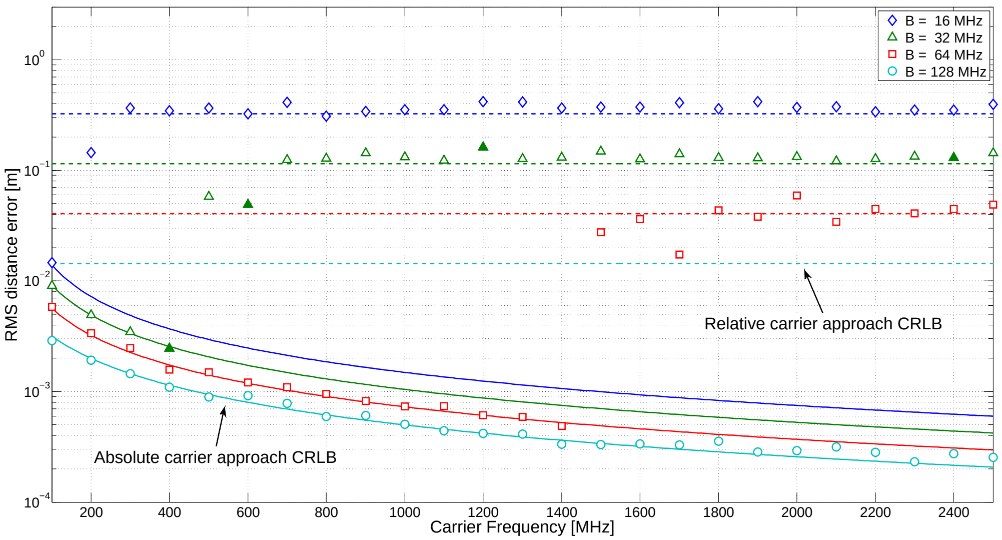
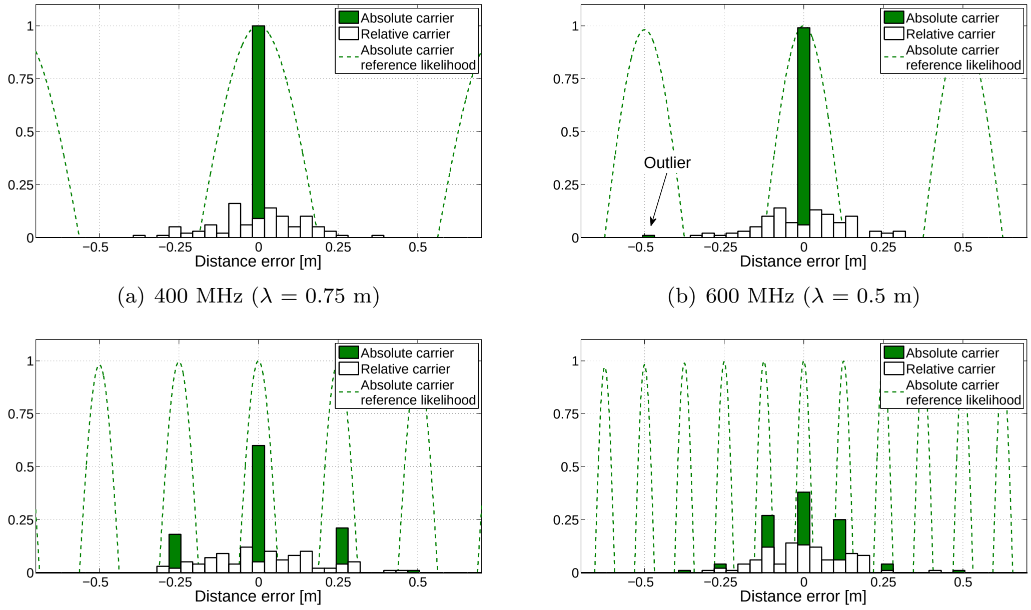


Figure 38: Simulated RMS distance estimation error as a function of the characteristic carrier frequency. Each trial used N observations with fixed $\Delta f = 1$ MHz frequency separation. Dashed and solid lines represent the CRLB of the *relative* and *absolute* carrier approaches, respectively. The error distributions corresponding to the four solid markers are shown in Figures 39(a)–(d). ($\text{SNR}_\gamma = 15$ dB, N = 16, 32, 64 and 128, L = 100 trials.)

To explain the performance break-down phenomena, consider the error distributions corresponding to $N = 32$ and carrier frequencies 400 MHz, 600 MHz, 1200 MHz and 2400 MHz (solid markers). Operating below the carrier frequency threshold, the distance estimates of the *absolute* carrier approach gather closely around the true value and exhibit significantly lower variance than the *relative* approach, see Figure 39(a). In the vicinity of the frequency threshold, most of the distance estimates are accurate but outliers appear at approximately wavelength apart from the true value, see Figure 39(b). Note that even a single outlier represents a significant drop in the estimator performance. Above the threshold, the decreased bandwidth-to-carrier frequency ratio makes the likelihood local maxima even less distinguishable and outliers at multiple wavelengths emerge, see Figures 39(c) and 39(d).



(a) 400 MHz ($\lambda = 0.75$ m)

(b) 600 MHz ($\lambda = 0.5$ m)

(c) 1200 MHz ($\lambda = 0.25$ m)

(d) 2400 MHz ($\lambda = 0.125$ m)

Figure 39: Distance estimate error distributions corresponding to the solid markers in Figure 38 with 32 MHz bandwidth at carrier frequencies 400 MHz (a), 600 MHz (b), 1200 MHz (c) and 2400 MHz (d).

Although the variance of the *relative* and *absolute* carrier based estimators is identical above the carrier frequency threshold, Figures 39(c) and 39(d) suggest that their corresponding error distribution functions are utterly different. Now as long as the initial assumptions hold and the estimator is unbiased, the structure of the error distribution function may be exploited.

One possible approach leads through estimating the *mean* of multiple d_{ABCD} estimates. Given M independent $d_{ABCD} \sim \mathcal{N}(\mu_d, \sigma_d)$ observations, the mean of the *sample mean*

$$\bar{d}_M \triangleq \frac{1}{M} \sum_{m=0}^{M} d_{ABCD,m} \tag{4.80}$$

remains the same, $\mu_{\bar{d}} = \mu_d$, while its standard deviation reduces to $\sigma_{\bar{d}} = \sigma_d/\sqrt{M}$. Therefore, the idea is to decrease $\sigma_{\bar{d}}$, so that $\bar{d}$ resides within the $\mu_d \pm \lambda/2$ region with some predefined confidence, as illustrated in Figure 40. Then, by rejecting all the outliers outside the $\bar{d} \pm \lambda/2$ region, the variance of the remaining observations approaches the CRLB of the *absolute* carrier approach.

Indeed, with the carrier frequency decreasing, the simulated RMS errors abandon the constant *relative* approach CRLB as it becomes less than approximately $\lambda/4$, observe in Figure 38. In other words, with $2\sigma_d \leq \lambda/2$ the d_{ABCD} estimate falls into the $\mu_d \pm \lambda/2$ region with more than 95% confidence, and starts to naturally snap onto the true *absolute* carrier likelihood peak.

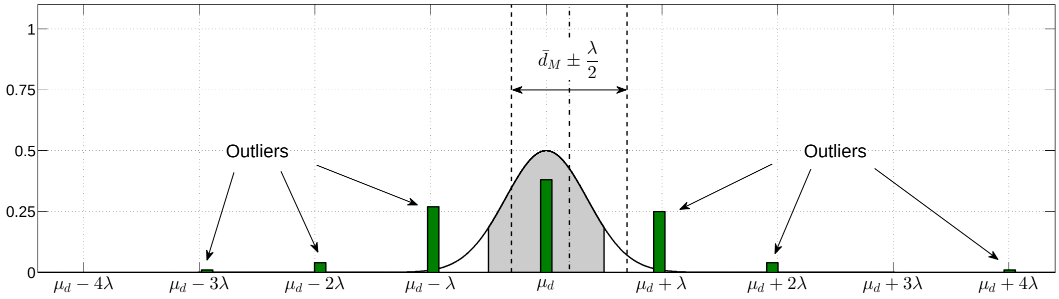


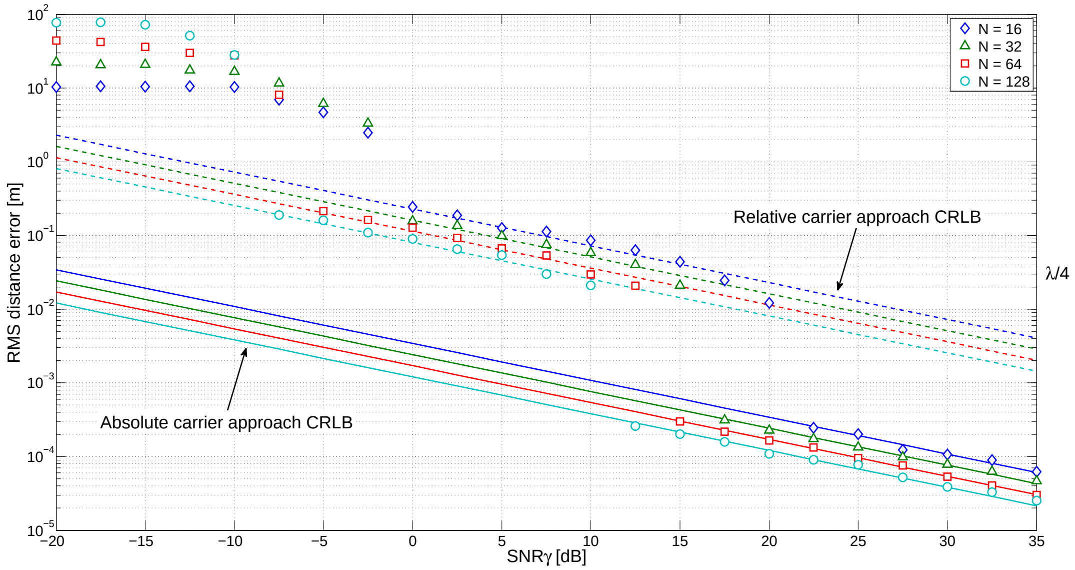
Figure 40: Using M independent d_{ABCD} estimates available, the standard deviation of the $\bar{d}_M$ sample mean is forced into the $\mu_d \pm \lambda/2$ (shaded) region with a certain confidence. Then, by discarding all the outlier d_{ABCD} estimates outside the $\bar{d}_M \pm \lambda/2$ region the estimation accuracy significantly improves.

SNR Threshold The performance impact of the SNR$_\gamma$ phase measurement noise on the *absolute* carrier based d_{ABCD} estimator is examined in two scenarios to cover its bandwidth dependence simultaneously. Figure 41(a) illustrates the Cramér-Rao bounds for the *relative* and *absolute* carrier approaches along with the simulated RMS distance estimation errors for the constant $B = N\Delta f$ bandwidth case. In the region above 20 dB SNR$_\gamma$ the estimator reaches its CRLB bound regardless of the number of γ_k observations spread out over B. At approximately 12-22 dB a breakdown occurs and the performance drops to that of the *relative* carrier based estimator. A more accurate characterization of the threshold level is that it occurs when the RMS error corresponding to (4.78) drops below approximately $\lambda/4$. In other words $2\sigma_{d,\mathrm{rel}}$ becomes less than $\lambda/2$, and the maximum of the enveloping (4.76) falls into the shaded $\pm\lambda/2$ region of Figure 40 with a probability above 95%. Consequently, (4.67) starts to select the correct local maximum of the likelihood function with high and increasing confidence.

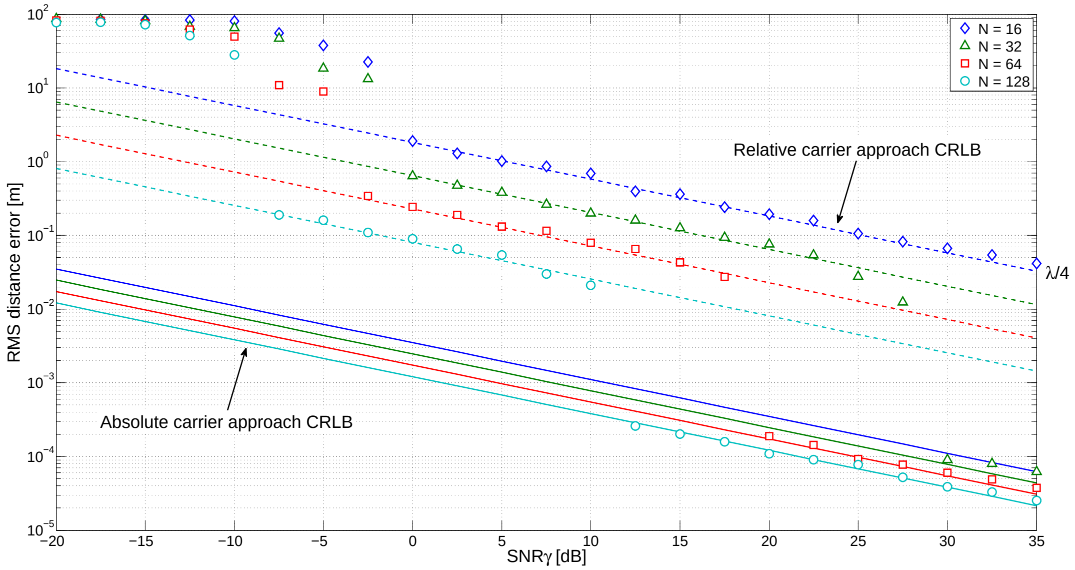
Observe that a second threshold is crossed between -7 dB and -2 dB, and with SNR$_\gamma$ decreasing the RMS error rapidly rises orders of magnitude. While this is not the typical operating region of interest, note that the departure of the simulated errors from the *relative* carrier approach CRLB is associated with the $B = N\Delta f$ bandwidth, consequently the width of the main lobe in Figure 37(a). Furthermore, below this second threshold, the RMS error converges to the maximum uniquely resolvable range, indicating the total collapse of the estimator performance.

In contrast, Figure 41(b) depicts the Cramér-Rao bounds and simulated RMS distance estimation errors using same N numbers of observations for each trial, but with fixed subcarrier spacing, consequently with varying bandwidth. The looser spacing of the *relative* carrier approach CRLBs now implies performance degradation with N decreasing, due to the $1/N\Delta f$ bandwidth dependence. On the other hand, the *absolute* carrier CRLBs remain in place as the $1/k_0\Delta f$ dominates the $1/N\Delta f$ dependence.

The threshold locations are determined by the same rules as for the fixed bandwidth case. The distance estimate errors start to depart the CRLBs of the *relative* carrier approach, and attain those of the *absolute* one, when the former become smaller than approximately $\lambda/4$. Since these bounds are now higher, the corresponding thresholds are spread out over a larger SNR$_\gamma$ region above 12 dB. Since the *relative* carrier threshold increases proportionally with the width of the main lobe in Figure 37(a), the second thresholds remain between -7 dB and -2 dB. Furthermore, the since the subcarrier spacing is independent of N, all the simulated errors converge to the same uniquely resolvable range.



(a) Fixed $B = N\Delta f$ bandwidth (B = 128 MHz, Δf = 8 MHz, 4 MHz, 2 MHz and 1 MHz).



(b) Fixed $\Delta f = B/N$ carrier separation (Δf = 1 MHz, B = 16 MHz, 32 MHz, 64 MHz and 128 MHz).

Figure 41: Simulated RMS distance estimation error versus SNR$_\gamma$, with each trial using N observations and f_0 carrier frequency with fixed B bandwidth (a) and Δf carrier separation (b). Dashed and solid lines represent the CRLB of the *relative* and *absolute* carrier approaches, respectively. (f_0 = 2400 MHz, N = 16, 32, 64 and 128, L = 100 trials.)

The above analysis of the bandwidth–carrier frequency relationship suggests that the accuracy *offered* by the *absolute* carrier approach is primarily determined by the carrier frequency. However, due to the non-linearity of the estimator, the corresponding Cramér-Rao bound is attained only when operating in regions determined by other parameters. To control the threshold of these regions, hence performance of the distance estimation, the following strategies may be employed:

1. **Extend the measurement bandwidth.** Increasing the $B = N\Delta f$ bandwidth either through additional subcarriers or expanded subcarrier spacing reduces the main-lobe-width of the *absolute* subcarrier based likelihood envelope, see Figure 37(b). In turn, an adequately narrow main lobe of the envelop ensures that the location of the largest peak of the likelihood function corresponds to the true d_{ABCD} distance. These make bandwidth extension a convenient way to adjust the threshold levels responsible for rapid performance breakdowns without altering the attainable accuracy. Note, however, that simply increasing the Δf subcarrier spacing decreases the largest uniquely identifiable d_{ABCD} region and may result in ambiguously overlapping results, see Section 4.4.1.

2. **Lower the carrier frequency.** An alternative strategy for avoiding the collapse of the estimator performance is to reduce the carrier frequency, as suggested by Figure 38. Increasing the characteristic wavelength makes the peaks of the likelihood function more distinguishable, see Figures 37(c)–(d), with respect to the envelope defined by a given bandwidth. However, as Figure 38 also implies, the price for maintaining a healthy bandwidth-carrier frequency ratio this way is paid by the degraded attainable accuracy.

3. **Improve the phase measurement accuracy.** Reducing the phase measurement noise both improves the CRLB and helps to actually attain it, see Figures 41(a)–(b). While the measurement noise primarily depends on the channel noise and the phase measurement method in general, it can be significantly improved by averaging multiple phase measurements, as described in Section 3.4.2. Phase measurement averaging can be efficiently performed at the receiver and imposes no additional communication overhead. Consequently, it is the most straightforward way to improve the estimation performance, especially in scenarios where the operating frequency band is constrained.

4. **Increase the number of γ_k observations.** Performing the distance estimation based on a larger collection of *relative* phase offset measurements improves SNR_γ, which both lowers the CRLB and extends the useful operating region. Independent phase offset measurements obtained at the same f_k frequencies can be explicitly averaged to increase SNR_γ. Furthermore, measurements from widely different frequencies may be stitched together and considered as a single set of γ_k observations for distance estimation, which is an implicit form of noise averaging.

5. **Generate distance estimate statistics.** As a last resort, repeating the d_{ABCD} distance estimation based on independent sets of γ_k gives a means to exploit the characteristics of the error distribution function, see Figure 39. The properly reduced sample mean variance allows to isolate the outliers, as illustrated in Figure 40, consequently, to reduce the estimation error.

5 MarmotE SDR Implementation

A key feature of the radio interferometric phase measurement is that it employs unmodulated sinusoid carriers only, therefore, requires no custom waveforms to estimate the *absolute* phase offsets. This, in conjunction with the minimal receiver-side signal processing requirement, enabled its implementation and experimental evaluation using the CC1000 radio chip equipped MICA2 motes [9].

In contrast, the multi-carrier phase estimation, proposed in Section 3, assumes direct access to the baseband complex signals and moderate signal processing capability on both the transmitter and the receiver sides. The MarmotE SDR platform naturally lends itself for hosting such OFDM-based architectures, therefore, it is configured to implement the entire transmitter functionality. Even though, the available logic resources allow for the receiver functionality to be implemented on the same MarmotE SDR node, the received baseband signals are recorded using USRP N210 desktop SDRs for offline evaluation. The rest of this section discusses the MarmotE SDR design of the proposed multi-carrier phase estimation method.

5.1 Baseband Waveforms

The design of the multi-carrier baseband waveforms translates to the construction of cyclic prefix (CP) free OFDM symbols subject to constraints (4.22) and (4.23) in Section 3.2, and the recommendations of Sections 3.4.3 and 3.4.4. That is, each subcarrier in

$$u_A(t) = \sum_{k=0}^{N-1} m_{A,k} e^{j[2\pi k \Delta f t + \phi_{A,k}]} \quad \text{and} \quad u_B(t) = \sum_{k=0}^{N-1} m_{B,k} e^{j[2\pi k \Delta f t + \phi_{B,k}]} \tag{4.81}$$

needs to be assigned to one of the transmitters A or B exclusively, and at least two subcarriers are required per transmitter.

Setting the number of subcarriers to $N = 32$ and restricting the domain of the subcarrier magnitudes and phases to

$$m_k \in \mathcal{M} = \{0, 1\} \quad \text{and} \quad \phi_k \in \mathcal{P} = \{0, \pi\}, \tag{4.82}$$

respectively, the unit amplitude subcarriers are assigned to the two transmitters based on the allocation maps `0xCCCCCCCC` and `0x33333333`. Such an allocation scheme prevents ambiguities in the later processing stages, see Section 3.4.3. The subcarrier phases are calculated based on (4.55) for each transmitter individually, although, the resulting phase map is stored in a common vector as described in (4.56). Observe that due to (4.82) the ϕ_k phase shifts represent the multiplication of the subcarrier by $s_k \triangleq e^{j\phi_k} \in \{\text{-}1, 1\}$.

The discrete time baseband waveforms are constructed by elementwise multiplying the vectors

$$\mathbf{m} = [m_0, \ldots, m_{N-1}] \quad \text{and} \quad \mathbf{s} = [s_0, \ldots, s_{N-1}], \tag{4.83}$$

and calculating the inverse Discrete Fourier Transform (IDFT),

$$u[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} m_k s_k \, e^{j2\pi kn/N} \quad 0 \leq n < N. \tag{4.84}$$

The $u[n]$ vector is then extended periodically, $u[n] = u[n + iN]$ for $\forall i \in \mathbb{Z}$, by repeatedly calculating the IDFT. Note that equivalently, a single period of $u[n]$ may be calculated, stored in a memory and fetched back with circular addressing to construct the continuous baseband waveform.

5.2 Transmitter Design

In the MarmotE SDR multi-carrier transmitter, the SoPC microcontroller is responsible for constructing the OFDM symbols and for handling the high-level parameters, such as digital gain control or cyclic prefix length. For the multi-carrier phase measurements, the microcontroller sets the cyclic prefix length to zero, and preloads the constant OFDM symbol based on the parameters $\mathbf{m}_A, \mathbf{m}_B$ and $\mathbf{s}$, and the actual transmitter role, A or B. The parameters and symbols are transferred to the FPGA fabric through the AMBA bus, where the interface logic stores them in registers and a FIFO, respectively.

The OFDM waveform generation then takes place entirely in the FPGA fabric, according to the Simulink model shown Figure 42, which operates on a single 20 MHz clock domain. This implicitly defines the 20 MHz DAC sampling rate and sets the timing constraint for every processing block. Therefore, the 32-point IDFT operation of (4.84) is calculated by a full-parallel IFFT module, capable of producing a new $u[n]$ sample in every clock period with a constant 71-cycle delay. The cycle accurate synchronization of the IFFT and the optional cyclic prefixer blocks is then ensured by the timing controller.

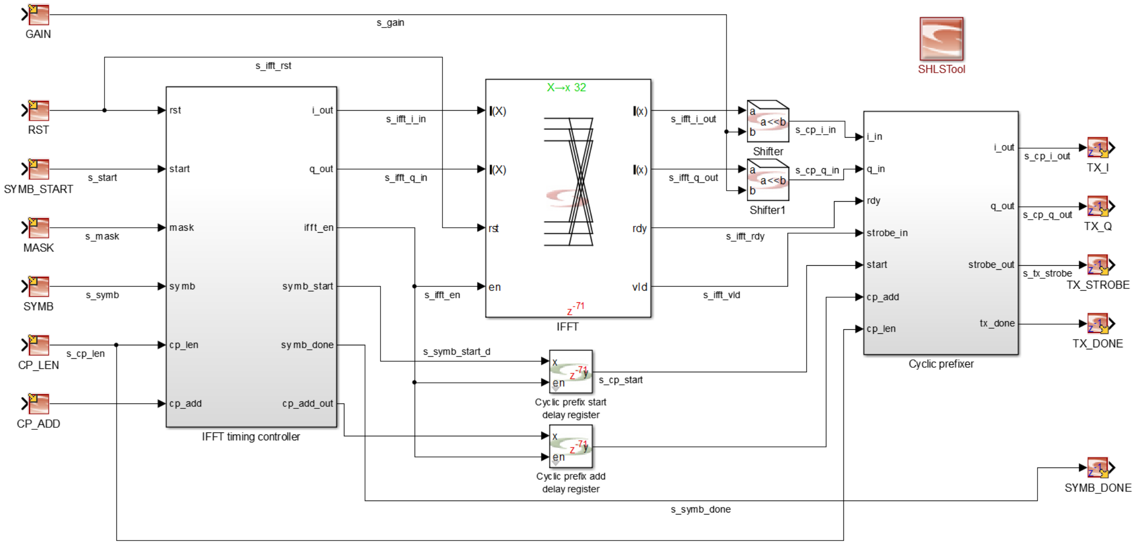


Figure 42: Simulink model of the HDL synthesizable OFDM transmitter module.

Observe that the MarmotE SDR transmitter design is able to produce arbitrary OFDM waveforms, such as the one shown in Figure 43, although the multi-carrier phase measurement of Section 3.2 requires only the repeated transmission of a single OFDM symbol. On the other hand, the current design provides no automated way to perform frequency synchronization to another transmitter due to the lack of receiver functions.

The corresponding FPGA logic utilization of the OFDM transmitter path is summarized in Table 15. The largest component is clearly the 32-point IFFT block, demanding over 45% of the available general logic resources and 25% of the block RAMs, primarily due to the parallel nature of the block and the lack of hardware multipliers in the FPGA fabric. In comparison, the logic use of the timing control and cyclic prefixer blocks is almost negligible, each being under 2%. The rest

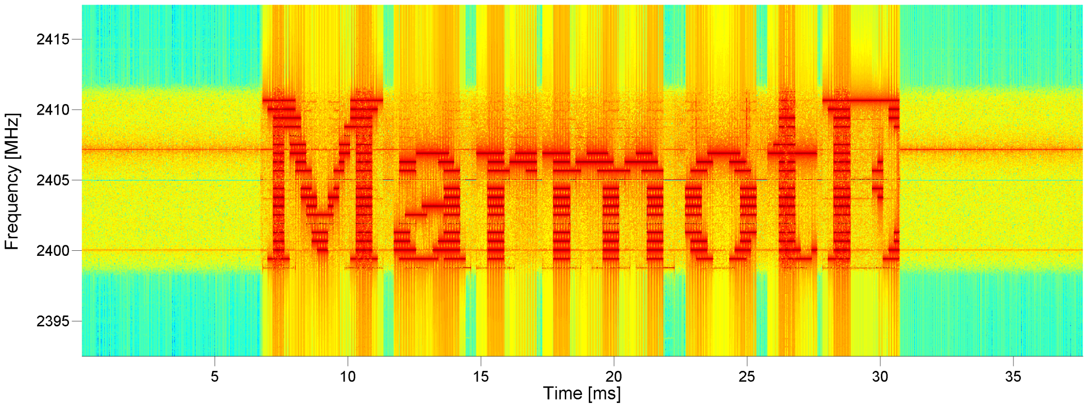


Figure 43: Spectrogram of the measured OFDM waveform displaying 'MarmotE'.

of the Simulink synthesized design in Figure 42 consumes 7.2%, where the major contributors are the two 71-delay register lines. Finally , the AMBA bus interface takes one block RAM and 7.5% of the logic cells, however, the latter is divided approximately equally between the registers accessible from the AMBA bus and their corresponding control logic.

Component	**Logic cells**	**Block RAM**
AMBA interface	883 (7.5%)	1 (4%)
Timing controller	221 (1.9%)	0 (0%)
IFFT	5272 (45.6%)	6 (25%)
Cyclic prefixer	205 (1.8%)	0 (0%)
Other (Simulink)	831 (7.2%)	0 (0%)
Total	**7412 (64.0%)**	**7 (29%)**

Table 15: FPGA logic resource utilization of the OFDM transmit path.

6 Performance Evaluation

The development and analysis of the proposed *measurement* stage for radio signal phase-based localization assumed a channel model with perfect line-of-sight component corrupted only by additive white Gaussian noise so far. The goal of the following experiment is to provide a proof-of-concept evaluation of both the multi-carrier phase and the distance estimation algorithms in a real-world scenario. For that, a set of MarmotE SDR nodes and two USRP N210s were deployed in an low-multipath environment and the estimated d_{ABCD} distances are compared to an independently established ground truth.

6.1 Measurement Setup

The field measurements employed four MarmotE SDR nodes as transmitters, configured with the OFDM transmitter design described in Section 5.2, and two RFX2400 daughterboard equipped USRP N210 desktop SDRs as receivers, each connected to a laptop computer to record the raw received baseband waveforms. Both the transmitters and the receivers were mounted on tripods, extended to 115 cm height, and deployed in an open outdoor environment to minimize ground reflection and other sources of multipath propagation. The six nodes were arranged around a 30 m diameter circle, as shown in Figure 44, however, their pairwise distance was measured using a laser rangefinder to establish an accurate ground truth. The measured antenna distances are summarized in Table 16, where the errors are assumed to be in the cm order.

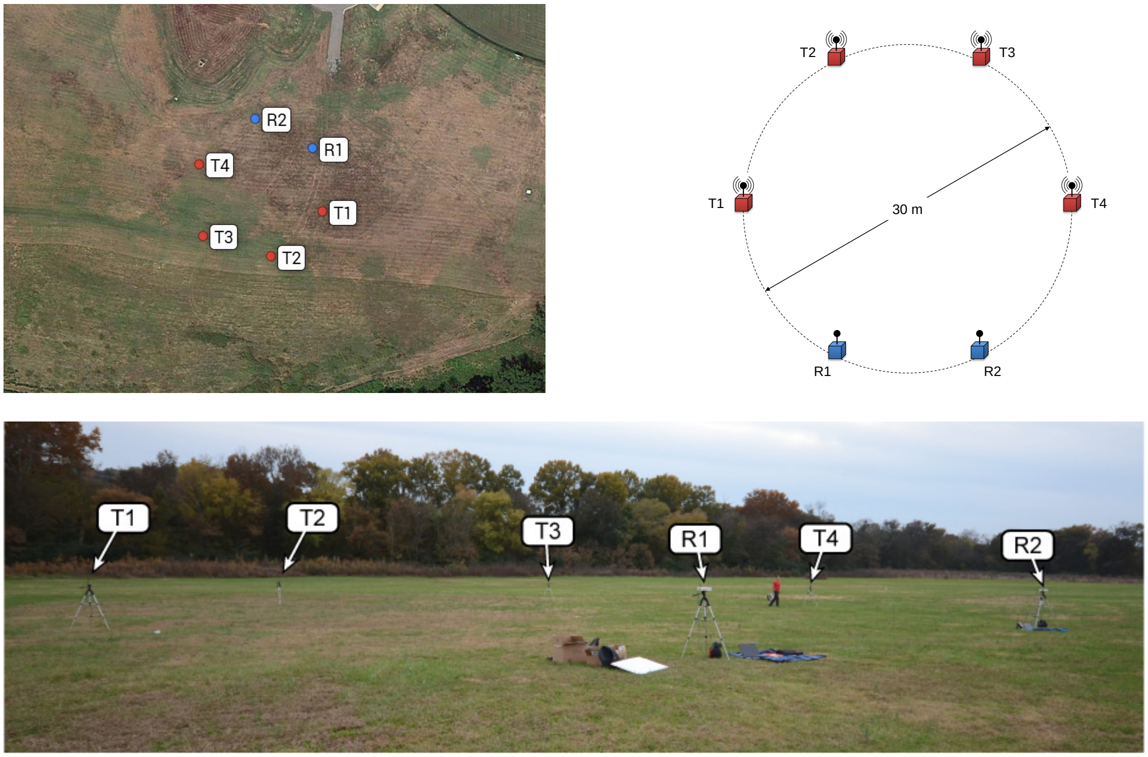


Figure 44: Satellite view (top-left), schematic arrangement (top-right) and photo (bottom) of the multi-carrier phase measurement setup. Map data © 2014 Google.

	T1	**T2**	**T3**	**T4**
R1	14.88	25.92	29.97	26.04
R2	25.88	29.96	26.02	15.14

Table 16: Ground truth pairwise node distances measured with laser rangefinder.

The multi-carrier phase measurements were carried out over the 2.4-2.5 GHz frequency band, which was divided into 11 overlapping channels. Each channel occupied 20 MHz bandwidth and their center was spaced 10 MHz apart, starting at 2400 MHz.

For every measurement, two MarmotE SDR nodes and the two USRP N210s tuned to the center of a given channel, but the former transmitted only on a subset of the available subcarriers. That is, while $N = 32$ subcarrier locations were arranged nearly symmetrically around the carrier, the center one and several on both sides were disabled to account for the direct-conversion receiver architecture and to relax the baseband filter requirements, respectively. Thus, restricting the operation to the central half of the available subcarriers reduced the effective channel bandwidth to 10 MHz, and allowed to concatenate the 11 individual measurements into a continuous set over the 2.4-2.5 GHz band, see Figures 45 and 46. Note that the missing central subcarrier was always recovered through the interpolation algorithm described in Section 3.3.1.

6.2 Results

The *relative* phase offset observations spaced $\Delta f = 0.625$ MHz apart over an effective bandwidth of 110 MHz are shown for all possible transmitter-pair combinations in Figures 45 and 46. The two figures respectively show the $\gamma_{ABCD}(f)$ values calculated from phase estimates based on a single OFDM symbol and on the average of $M = 16$ consecutive symbols. Visibly, reducing the phase measurement noise by averaging multiple raw phase estimates lowered the variance of the calculated *relative* phase offsets, as discussed in Section 3.4.2.

The d_{ABCD} distances were estimated in two steps. First, the coarse estimate was obtained through the relative carrier approach, then the fine estimate using the absolute carrier approach. Note that in both cases, the $\partial\gamma/\partial f$ slope of the relative phase offsets contains the *only* information of interest.

The corresponding d_{ABCD} distance estimates with different MarmotE SDR nodes playing the transmitter roles C and D are summarized in Tables 17 and 18 for the non-averaging and the averaging cases, respectively. Comparison of the ground truth and the estimated distances confirms that the proposed algorithm reliably calculates the unambiguous d_{ABCD} ranges. The mean distance estimation error is only a few centimeters, indicating that the observed estimates are unbiased. However, the standard deviation of the errors is approximately 40 cm, which is in the same order as the results obtained with interferometric 2.4 GHz measurements in [8].

C	D	Ground truth	Coarse estimate	Fine estimate	Error [m]
T1	T2	−6.96	−6.19	−6.21	−0.75
T1	T3	−14.95	−14.87	−14.84	−0.11
T1	T4	−21.90	−21.86	−21.91	0.01
T2	T3	−7.99	−8.59	−8.64	0.65
T2	T4	−14.94	−15.28	−15.22	0.28
T3	T4	−6.95	−6.69	−6.71	−0.24

Table 17: d_{ABCD} distance estimates and the error of fine estimates without symbol averaging.

C	D	Ground truth	Coarse estimate	Fine estimate	Error [m]
T1	T2	−6.96	−6.23	−6.20	−0.76
T1	T3	−14.95	−14.73	−14.72	−0.23
T1	T4	−21.90	−21.81	−21.79	−0.11
T2	T3	−7.99	−8.34	−8.40	0.41
T2	T4	−14.94	−15.30	−15.34	0.40
T3	T4	−6.95	−6.75	−6.71	−0.24

Table 18: d_{ABCD} distance estimates and the error of fine estimates with 16-times symbol averaging.

Assuming that the observed errors are unbiased, Figure 41(a) suggests that the system operates below the SNR_γ threshold. Indeed, the error variance is clearly above the *relative* carrier approach CRLB and the fine estimates provide negligible improvement compared to the coarse ones. Moreover, Figure 46 indicates that phase measurement averaging visibly improved the $\gamma(f)$ observations.

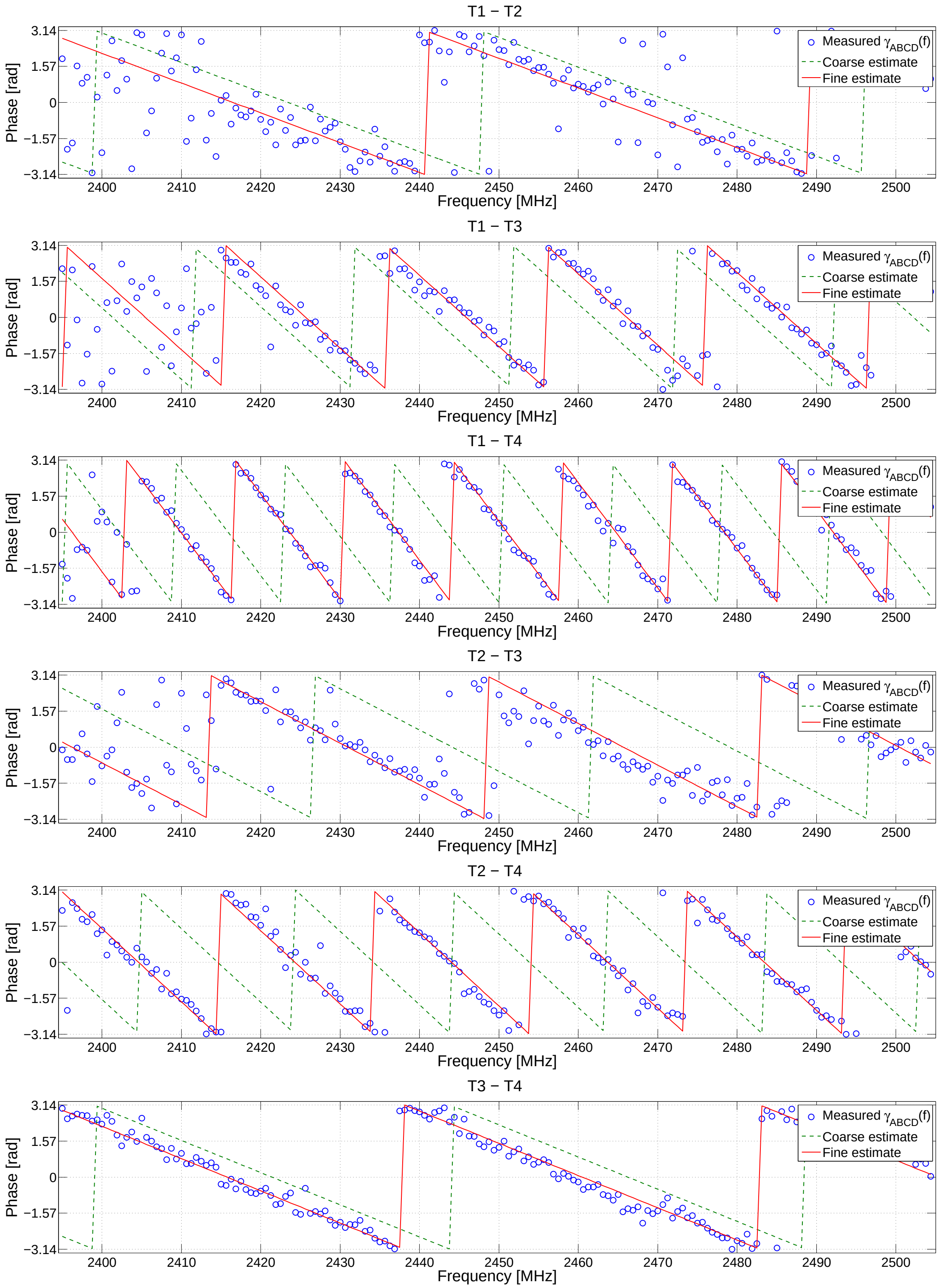


Figure 45: Multi-carrier distance estimation results without symbol averaging. Coarse and fine estimates correspond to the $\hat{\tau}_{\mathrm{ML}}$ output of Algorithms 4 and 5, respectively.

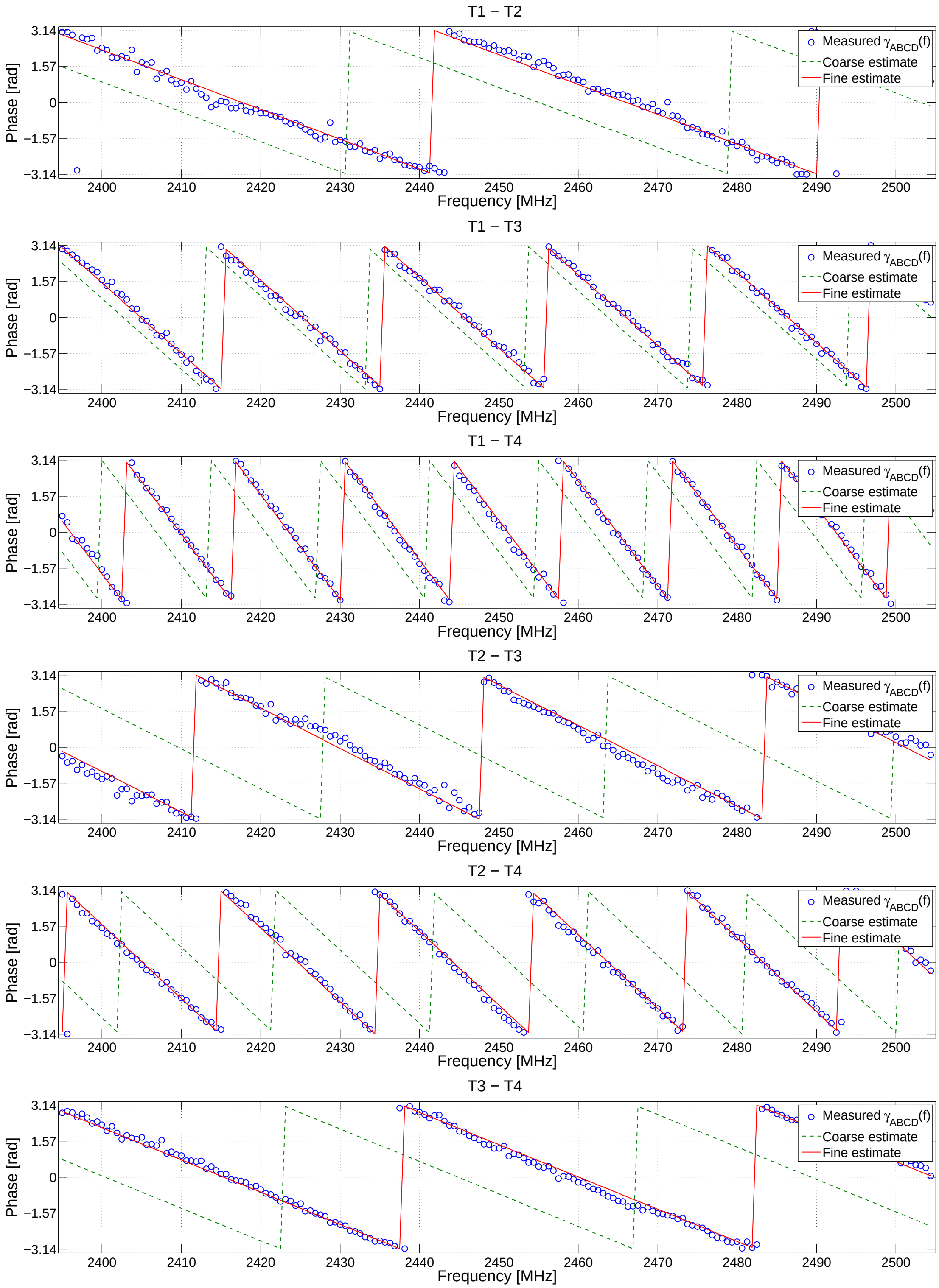


Figure 46: Multi-carrier distance estimation results with 16-times symbol averaging. Coarse and fine estimates correspond to the $\hat{\tau}_{\mathrm{ML}}$ output of Algorithms 4 and 5, respectively.

However, the overall fine estimate errors of Table 18 show negligible improvement and suggest that the attained SNR_γ improvement was insufficient to leave the sub-threshold region.

The two most probable sources of the phase measurement inaccuracies are the multipath effects and the synchronization errors. Although the measurements were performed in an open environment and the nodes were reasonably elevated, a secondary propagation path between either of the transmitter-receiver pairs immediately offsets the $\gamma(f)$ *relative* phase offsets, hence the d_{ABCD} estimates.

Furthermore, the OFDM transmitter design on the MarmotE SDR lacked a mechanism for frequency synchronization. Consequently, the carrier frequency offset between the transmitters was left uncompensated. The frequency offset was estimated to be in the kHz order, which introduced multi-carrier phase measurement errors through both receiver timing offset and inter-carrier interference, see Section 3.4.1. Receiver timing synchronization was established by manually detecting the onset of the signal from transmitter B in the recorded waveforms, with the impinging signal present from transmitter A already. The resulting timing offset was estimated to be in the order of tens of OFDM symbols. Its effect on the *relative* phase offset estimation is also analyzed in Section 3.4.1.

Another important performance metric is the time required to perform the phase *measurements* and the accompanying *calibration*. Table 19 summarizes the phase measurement related attributes of the single-carrier RIPS [9], SRIPS [8] and the proposed multi-carrier method. The RIPS and SRIPS rely on different radio architectures, which determines their characteristic carrier frequency and their calibration approach. The CC1000 operates in the 433 MHz, and its fine tuning capability allows for a precise, albeit slow, calibration process to deliberately introduce the 0.2–0.8 kHz carrier frequency offset (CFO) between the transmitters. This, in turn, makes the 9 kHz RSS sampling rate sufficient at the receiver to measure the *absolute* phase offset in 29 ms.

Method	**RIPS**	**SRIPS**	**Proposed**
Radio architecture	CC1000	CC2430	MarmotE SDR
Carrier frequency	433 MHz	2400 MHz	2400 MHz
Subcarrier count	1	1	32
Calibration time	$\gg$1 ms	-	$P \cdot 1.6\ \mu$s
Measurement time	29 ms	0.8–8 ms	$M \cdot 1.6\ \mu$s
Required CFO	0.2–0.8 kHz	0.2–14 kHz	-
Measured signal	RSS	RSS	Baseband I/Q
Sampling rate	9 kHz	62.5 kHz	20 MHz

Table 19: Performance comparison of the proposed and two existing phase estimation methods.

In contrast, the 2.4 GHz band CC2430 lacks such fine tuning capability, but offers an order higher sampling rate of the RSS signal. Therefore, SRIPS omits the calibration completely and expects the nominally set transmitter frequencies to generate a CFO between 0.2–14 kHz. When the CFO indeed falls in this range, the *absolute* phase offset is measured in 0.8–8 ms.

The MarmotE SDR also uses a radio front-end that operates in the 2.4 GHz band, and allows to sample the baseband I/Q signals at 20 MHz, as opposed to the RSS only. Contrary to the interferometric methods, the proposed phase estimation approach seeks zero CFO. Thus, preamble-based OFDM synchronization methods [131][133][134] may be employed to calibrate out the CFO between the two transmitters, as well as between the transmitter and the receivers, within a few OFDM symbol duration. Assuming $N = 32$ subcarriers, this calibration takes $P \cdot T_s$ time, where P is the number of preamble symbols, typically in the range of 2 to 4, and $T_s = 32/20$ MHz $= 1.6$ μs is the length of the cyclic prefix-free OFDM symbol. The actual multi-carrier phase measurement exhibits a similar speedup. The 11 phase measurements underlying the *relative* phase observations in Figures 45 took one OFDM symbol duration, 1.6 μs, each. Presuming that all $N = 32$ subcarriers are actively used, the effective measurement time per $\gamma(f)$ observation reduces to the 50 ns (!) sampling rate. When M consecutive symbols are averaged, as in the underlying measurements of Figure 46 for $M = 16$, the measurement duration clearly increases in proportion.

7 Conclusion

Although a large variety of node localization techniques have been proposed with substantially different trade-offs between size, cost, accuracy and infrastructure complexity, approaches that attain high accuracy with minimal hardware and infrastructure support are preferred in general. The RIPS took one step in this direction by introducing a phase measurement technique that required no transmitter-side and only minimal receiver-side signal processing. Thus, it allowed the measurement to be performed with traditional WSN nodes that rely on COTS radio chips and simple microcontrollers.

This chapter made to major contributions by proposing both an alternative phase measurement and a distance estimation method for sensor node localization. First, the *relative* phase offset measurement was generalized and related to TDOA estimation to point out that radio interferometry is only one specific approach to the problem. Then, assuming access to the baseband signals on both the transmitter and receiver sides, the problem was rephrased as a search for alternative waveforms that would enable the precise measurement of the *relative* phase offset. Eventually, a multi-carrier phase measurement scheme was proposed and thoroughly analyzed. The scheme assumed moderate baseband signal processing capability to employ OFDM waveforms. In return, it offered the following advantages compared to the single-carrier interferometric approaches:

- The transmitter power levels require no tuning because the subcarriers are allocated mutually exclusively and their phase is measured directly.
- The transmitter carrier frequencies offsets can and should be compensated for to relax the time synchronization requirement and reduce the inter carrier interference.
- The measurement time is reduced by more than four orders of magnitude, furthermore, the phases are inherently estimated at multiple frequencies simultaneously.

Finally, a working prototype was implemented using the MarmotE SDR platform and evaluated through field experiments.

The distance estimation method presumed $\gamma(f)$ relative phase offsets obtained through either the single or the multi-carrier approach. It modeled the $\gamma(f)$ observations as discrete samples of a complex sinusoid and related the problem to frequency estimation. Based on direct analogies with frequency estimation, two maximum-likelihood distance estimators and their corresponding Cramér-Rao bounds were derived. The introduction of the model provides the following benefits:

- The modulo 2π ambiguity of the $\gamma(f)$ observations is treated inherently through the complex sinusoid the model.
- The maximum-likelihood functions are concise and straightforward to calculate.
- The theoretical bounds provide both insight into the relationship between phase measurement noise, carrier frequency and effective bandwidth, and a benchmark for performance analysis.

Most importantly, however, the theoretical framework provided an explanation to the attained accuracy at 433 MHz with RIPS, and at 2400 MHz with SRIPS and our proposed methods.

CHAPTER V

CONCLUSION

1 Contributions

The software-defined radio (SDR) approach offers tremendous flexibility for prototyping and experimenting with novel radio communication protocols. However, the power consumption of existing such platforms renders them inapplicable for low-power wireless sensor networking. This dissertation attempted to introduce the SDR concept to wireless sensor networks. First, it demonstrated that with judicious architectural choices design flexibility and low-power operation are attainable at the same time. The proposed flash SoPC-based MarmotE SDR platform supports duty cycling and consumes 71 mW, 287 mW and 852 mW in sleep, receive and transmit (0 dBm) modes, respectively. These values are orders of magnitude less than the power consumption of traditional SDRs. Nevertheless, the platform offers sufficient computational resources to approach the WSN research from the PHY layer perspective, free from the architectural constraints of highly integrated radio chips.

The potential of the SDR approach in WSNs was demonstrated through the design of a spread-spectrum communication protocol and a multi-carrier phase measurement method for radio frequency node localization. The spread-spectrum PHY layer enabled an asynchronous multiple-access scheme and to increase the attainable hop-distance between the sensor node and the basestation without increasing the transmit power. The protocol was evaluated in various experiments using the MarmotE SDR platform, which could not have been performed with traditional WSN nodes.

The multi-carrier phase measurement method employed custom OFDM waveforms, which required both reasonable amount of configurable logic resources and direct access to the baseband signals. In return, it reduced the phase measurement time of 32 subcarriers to 1.6 μs, corresponding to 50 ns on average and a speedup of more than four orders of magnitude compared to the existing interferometric approaches. Furthermore, it offered a possible means to compensate the carrier frequency offset between the transmitter completely. The multi-carrier approach also inspired the mathematical model for distance estimation, which then led to the development of the theoretical performance bounds. In turn, the bounds served as performance benchmarks for distance estimation simulations, gave insight into the parameter dependence of the estimation and, consequently, influenced the design of the field experiments.

The communication protocol and the radio frequency node localization method presented in this dissertation are merely two examples of what the SDR concept might inspire in the WSN context. The wider adaptation of the SDR approach will hopefully put the WSN research into another perspective — the PHY layer perspective.

2 Future Work

Wireless platform architectures. The transmit and receive mode power consumptions of the MarmotE SDR platform measured to be more than an order of magnitude less than that of traditional SDRs. However, the sleep mode power consumption ended up outside of the originally targeted range. This is because the flash FPGA-based SoPC devices that were available during the design of the MarmotE SDR platform lacked an important power-saving feature common in stand-alone flash FPGAs. Actually, sacrificing the Flash*Freeze feature was a design trade-off for the low-latency and high communication bandwidth AMBA bus of the SoPC, expecting that future models will benefit of both. Indeed, the subsequent generation of flash FPGA-based SoPC devices included not only this low-power trait, desired for efficient duty cycling, but also hardware multipliers, which translate to both more computational power and higher power efficiency. This confirms that flash CMOS technology based FPGAs and SoPCs are a promising direction for the design of future low-power SDR architectures.

Sensor node localization. The experimental evaluation of the multi-carrier phase measurement method employed MarmotE SDR nodes as OFDM transmitters and USRP N210 receivers to capture the entire 20 MHz bandwidth of the received signals for offline analysis. As the phase measurement method has been validated and the MarmotE SDR has sufficient resources to host the OFDM receiver as well, a straightforward next step would be to design large-scale experiments with all-MarmotE SDR nodes. Given a complete OFDM transceiver, in turn, enables (i) actual communication between the sensor nodes using an OFDM-based PHY layer, (ii) preamble-based OFDM-synchronization methods [131][133][134] to compensate the timing and carrier frequency offset between all sensor nodes in only few OFDM symbol duration and (iii) experimentation with simultaneous communication and localization. As for (iii), observe that in an OFDMA scheme, the existence of a working OFDM PHY layer already implies mechanisms for carrier frequency offset compensation and receiver timing synchronization. Furthermore, Algorithm 3 can potentially operate on OFDM symbols with unknown constellations, as long as the same transmitted symbols can be matched from the receivers.

APPENDIX A

WIRELESS NODE ARCHITECTURES

1 GMSK Transceiver Design

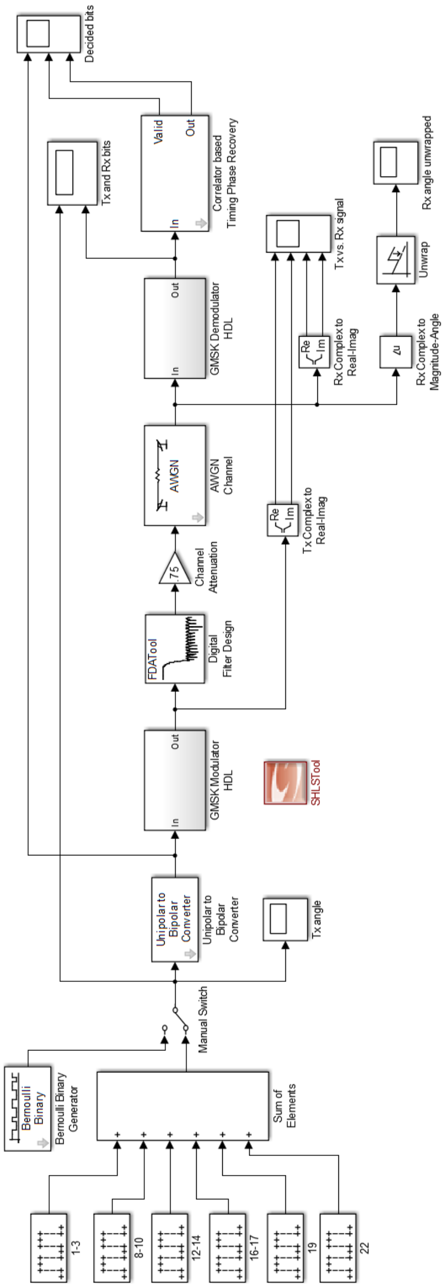


Figure 47: Simulink test setup of the HDL synthesizable GMSK modulator and demodulator.

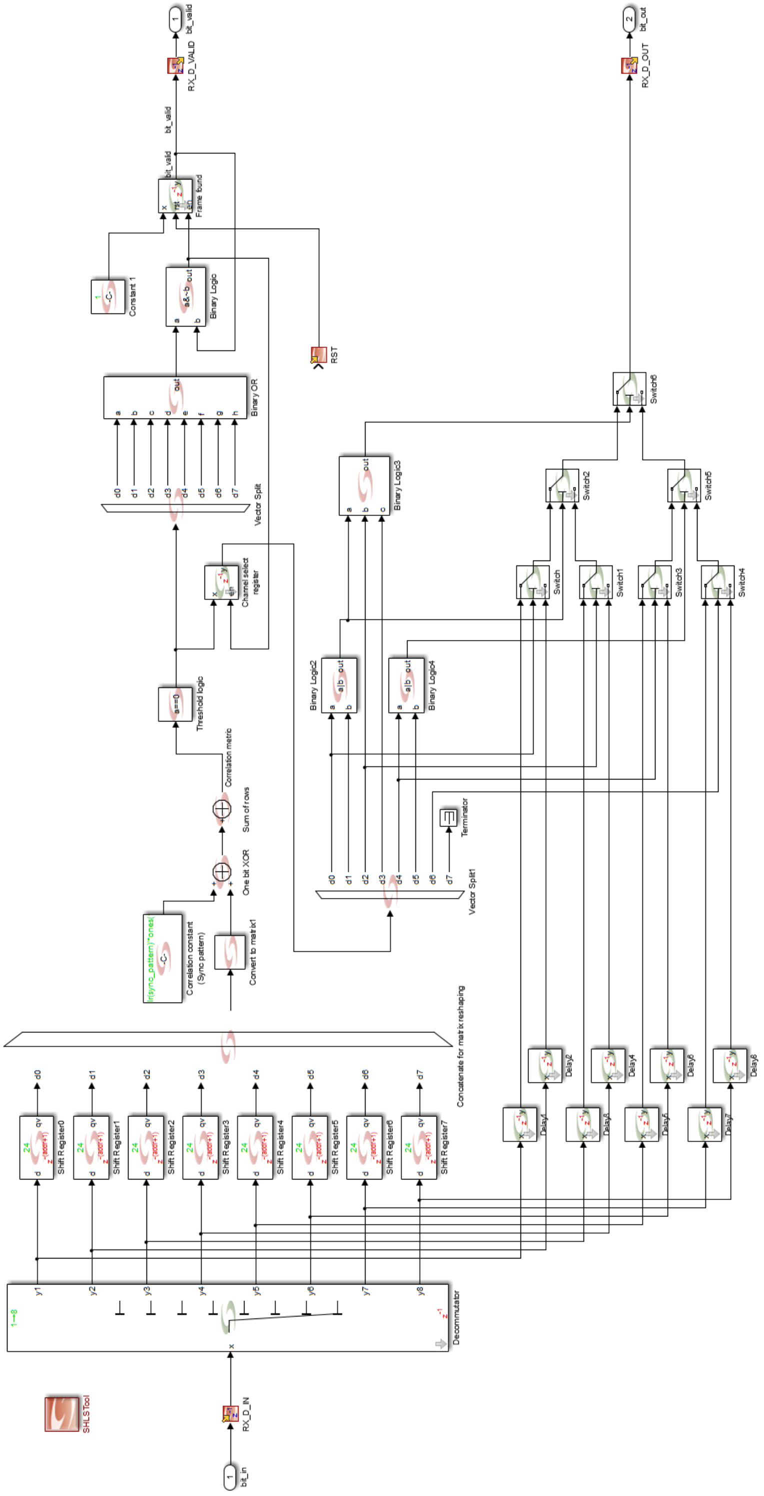


Figure 48: Simulink block diagram of the GMSK receiver symbol and frame synchronizer.

APPENDIX B

SENSOR NODE LOCALIZATION

1 Single-Carrier Phase Estimation

In RIPS [9] and SRIPS [8] transmitters A and B are tuned to slightly different frequencies, f_A and f_B with $\delta = (f_A - f_B)/2$, $\delta \ll f_A, f_B$, and transmit the unmodulated carrier, see Figure 27. Following the notation of Section 3.1.1 this can alternately be modeled as carriers tuned to $f = (f_A + f_B)/2$ and modulated with baseband complex signals

$$u_A(t) = e^{+j2\pi\delta t} \quad \text{and} \quad u_B(t) = e^{-j2\pi\delta t}, \tag{B.1}$$

respectively. The transmitted radio signals then take the form

$$s_A(t,f) = \text{Re}\left[e^{+j2\pi\delta(t-t_A)}\, e^{j(2\pi f t+\varphi_A)}\right] \tag{B.2a}$$

$$s_B(t,f) = \text{Re}\left[e^{-j2\pi\delta(t-t_B)}\, e^{j(2\pi f t+\varphi_B)}\right], \tag{B.2b}$$

where δ is defined as the *interference* frequency, t_A and t_B are the onset of the baseband waveforms, f is the carrier frequency, while φ_A and φ_B are the initial phases of the corresponding local oscillators. Note that with the notation $\varphi'_A = -2\pi\delta t_A + \varphi_A$ and $\varphi'_B = 2\pi\delta t_B + \varphi_B$, the above equations indeed reduce to the umodulated single-carrier case

$$s_A(t,f) = \text{Re}\left[e^{j2\pi(f+\delta)t+\varphi'_A}\right] \tag{B.3a}$$

$$s_B(t,f) = \text{Re}\left[e^{j2\pi(f-\delta)t+\varphi'_B}\right]. \tag{B.3b}$$

The transmitted signals from nodes A and B observe $\tau_{AY} = d_{AY}/c$ and $\tau_{BY} = d_{BY}/c$ path delays before they reach node Y, where Y denotes either receiver C or D. Therefore, the two received passband signal components are

$$s_{AY}(t,f) = \text{Re}\left[e^{j\left[2\pi(f+\delta)(t-\tau_{AY})+\varphi'_A\right]}\right] \tag{B.4}$$

$$s_{BY}(t,f) = \text{Re}\left[e^{j\left[2\pi(f-\delta)(t-\tau_{BY})+\varphi'_B\right]}\right] \tag{B.5}$$

and their superposition is

$$s_Y(t,f) = \text{Re}\left[e^{j\left[2\pi(f+\delta)(t-\tau_{AY})+\varphi'_A\right]} + e^{j\left[2\pi(f-\delta)(t-\tau_{BY})+\varphi'_B\right]}\right]. \tag{B.6}$$

After down-mixing and low-pass filtering the complex baseband signal can be written as

$$\begin{aligned} r_Y(t,f) \quad &= \quad e^{j\left[+2\pi\delta t+2\pi(f+\delta)(-\tau_{AY})+\varphi'_A-\varphi_Y\right]} && \text{(B.7)} \\ &+ \quad e^{j\left[-2\pi\delta t+2\pi(f-\delta)(-\tau_{BY})+\varphi'_B-\varphi_Y\right]} && \text{(B.8)} \end{aligned}$$

Now consider the absolute square of the baseband complex envelope signal, which can be written as

$$\begin{aligned}
|r_Y(t,f)|^2 &= r_Y(t,f)\, r_Y^*(t,f) \\
&= 2 \,+ e^{+j\left[4\pi\delta t - 2\pi(f+\delta)\tau_{AY} + 2\pi(f-\delta)\tau_{BY} + \varphi'_A - \varphi'_B\right]} \\
&\qquad + e^{-j\left[4\pi\delta t - 2\pi(f+\delta)\tau_{AY} + 2\pi(f-\delta)\tau_{BY} + \varphi'_A - \varphi'_B\right]} \\
&= 2 \,+ e^{+j\left[4\pi\delta t + 2\pi f(\tau_{BY} - \tau_{AY}) + 2\pi\delta(-\tau_{AY} - \tau_{BY}) + \varphi'_A - \varphi'_B\right]} \\
&\qquad + e^{-j\left[4\pi\delta t + 2\pi f(\tau_{BY} - \tau_{AY}) + 2\pi\delta(-\tau_{AY} - \tau_{BY}) + \varphi'_A - \varphi'_B\right]} \\
&= 2 \,+ 2\cos[2\pi f(-\tau_{BY} + \tau_{AY}) + 2\pi\delta(\tau_{BY} + \tau_{AY}) - 4\pi\delta t + \varphi'_B - \varphi'_A].
\end{aligned} \tag{B.9}$$

The *absolute* phase offset information is carried in the argument of the cosine function

$$\vartheta_Y(t,f) = 2\pi f(-\tau_{BY} + \tau_{AY}) + 2\pi\delta(\tau_{BY} + \tau_{AY}) - 4\pi\delta t + \varphi'_B - \varphi'_A, \tag{B.10}$$

where the terms containing δ are the unwanted side-effects of the interferometric approach and φ'_A and φ'_B are the also undesirable transmitter initial phases. Observe that, on the one hand, the φ_Y receiver local oscillator phase disappears. While, on the other hand, the method introduces the $4\pi\delta t$ term, which depends on the measurement start time of the receivers.

To remove the the transmitter initial phases φ_A and φ_B, consider the *relative* phase difference between receivers C and D, taken at time instants t_C and t_D, respectively:

$$\vartheta_D(t_D,f) - \vartheta_C(t_C,f) = 2\pi f\,(-\tau_{BD} + \tau_{AD} + \tau_{BC} - \tau_{AC}) \tag{B.11a}$$

$$+\, 2\pi\delta\,(\tau_{BD} + \tau_{AD} - \tau_{BC} - \tau_{AC}) \tag{B.11b}$$

$$-\, 4\pi\delta\,\underbrace{(t_D - t_C)}_{\text{relative TO}}, \tag{B.11c}$$

where $t_D - t_C$ is the *relative timing offset* between the two receivers, and the last term containing δ is usually considered negligible as $\delta \ll f$. Comparison of (4.11) with (B.11) suggests that these two unwanted terms represent the price paid for feasibility.

Note that δ is a design parameter, furthermore, both the static and time dependent error terms are linear in δ. Therefore, the constant error term (B.11b) can be arbitrarily reduced by choosing an appropriately small interference frequency, which, in turn, also relaxes the time synchronization requirements determined by (B.11c). Even though $\delta \to 0$ makes (B.11) converge to (4.11), the use of excessively low interference frequencies introduces several technical issues. First, the measurement time required to perform reliable phase measurements increases for low frequencies approximately $\propto 1/\delta$. Second, the short-term frequency stability of the transmitter local oscillators severely limits the attainable carrier frequency accuracy.

Observe that with the interferometric approach, the measured *relative* phase offset is independent of the actual IF frequency. Thus, after down-mixing and proper band-pass filtering the complex baseband signal can be written as

$$r_Y(t,f) = \left[e^{j\left[2\pi(f+\delta)(t-\tau_{AY}) + \varphi'_A\right]} + e^{j\left[2\pi(f-\delta)(t-\tau_{BY}) + \varphi'_B\right]}\right] \cdot \underbrace{e^{-j[2\pi(f-f_{IF})t + \varphi_Y]}}_{|\cdot|=1}. \tag{B.12}$$

2 Cramér-Rao Lower Bound for Distance Estimation

2.1 Derivation

Let us model the *relative* phase offsets at frequencies f_n as a slightly more general single complex exponential in CWGN

$$\gamma(f_n) = A \exp\left[j\left(2\pi f_n \tau + \phi\right)\right] + W \qquad n = 0, 1, \ldots, N-1 \tag{B.13}$$

where $W \sim \mathcal{CN}(0, \sigma_\gamma)$. Assume that the *relative* phase offsets are measured equidistantly in frequency with a separation of Δf and the lowest frequency used is f_0

$$f_n = f_0 + n\Delta f = (n_0 + n)\Delta f \tag{B.14}$$

Then, the data model can be written as

$$\gamma[n] = A \exp\left[j\left(2\pi(n_0 + n)\Delta f \tau + \phi\right)\right] + W[n] \qquad n = 0, 1, \ldots, N-1 \tag{B.15}$$

The joint probability density function of the elements of the *relative* phase offset sample vector $\boldsymbol{\gamma}$

$$p(\boldsymbol{\gamma}, \boldsymbol{\alpha}) = \frac{1}{(2\pi\sigma^2)^N} \exp\left[-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(\gamma[n] - A\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right]\right)^2\right] \tag{B.16}$$

where $\boldsymbol{\alpha}$ is the vector of unknown parameters is

$$\boldsymbol{\alpha} = [A, \tau, \phi]^T \tag{B.17}$$

To determine Cramér-Rao lower bound we first calculate the Fisher information for CWGN based on [137, p. 525]

$$[\boldsymbol{J}(\boldsymbol{\alpha})]_{ij} = \frac{2}{\sigma^2}\,\mathrm{Re}\left[\sum_{n=0}^{N-1}\frac{\partial s[n,\boldsymbol{\alpha}]}{\partial \alpha_i}\frac{\partial s[n,\boldsymbol{\alpha}]}{\partial \alpha_i}\right] \tag{B.18}$$

where the subscripts i and j refer only to the unknown elements of $\boldsymbol{\alpha}$ and

$$s[n, \boldsymbol{\alpha}] = A\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right] \tag{B.19}$$

The corresponding partial derivatives of $s[n, \boldsymbol{\alpha}]$ are

$$\frac{\partial s[n,\boldsymbol{\alpha}]}{\partial A} = \exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right] \tag{B.20}$$

$$\frac{\partial s[n,\boldsymbol{\alpha}]}{\partial \tau} = j2\pi(n_0+n)\Delta f A\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right] \tag{B.21}$$

$$\frac{\partial s[n,\boldsymbol{\alpha}]}{\partial \phi} = jA\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right] \tag{B.22}$$

Thus, the Fisher information matrix in the most general case is

$$\boldsymbol{J}(\boldsymbol{\alpha}) = \frac{2}{\sigma^2}\begin{bmatrix} \sum\limits_{n=0}^{N-1} 1 & 0 & 0 \\ 0 & \sum\limits_{n=0}^{N-1} (2\pi(n_0+n)\Delta f A)^2 & \sum\limits_{n=0}^{N-1} 2\pi(n_0+n)\Delta f A^2 \\ 0 & \sum\limits_{n=0}^{N-1} 2\pi(n_0+n)\Delta f A^2 & \sum\limits_{n=0}^{N-1} A^2 \end{bmatrix} \tag{B.23}$$

or

$$\boldsymbol{J}(\boldsymbol{\alpha}) = \frac{2}{\sigma^2}\begin{bmatrix} N & 0 & 0 \\ 0 & (2\pi\Delta f A)^2(n_0^2 N + 2n_0 P + Q) & 2\pi\Delta f A^2(n_0 N + P) \\ 0 & 2\pi\Delta f A^2(n_0 N + P) & NA^2 \end{bmatrix} \tag{B.24}$$

where

$$P = \sum_{n=0}^{N-1} n = \frac{N(N-1)}{2} \tag{B.25}$$

$$Q = \sum_{n=0}^{N-1} n^2 = \frac{N(N-1)(2N-1)}{6} \tag{B.26}$$

Since we are interested in estimating τ there are four different cases depending on whether A and ϕ is known or unknown. The corresponding Fisher matrices and their inverses are

1. A is known and ϕ is known

$$\boldsymbol{J}(\boldsymbol{\alpha}) = \frac{2}{\sigma^2}(2\pi\Delta f A)^2(n_0^2 N + 2n_0 P + Q) \tag{B.27}$$

$$\boldsymbol{J}^{-1}(\boldsymbol{\alpha}) = \frac{\sigma^2}{2}\frac{1}{(2\pi\Delta f A)^2(n_0^2 N + 2n_0 P + Q)} \tag{B.28}$$

2. A is known and ϕ is unknown

$$\boldsymbol{J}(\boldsymbol{\alpha}) = \frac{2}{\sigma^2}\begin{bmatrix} (2\pi\Delta f A)^2(n_0^2 N + 2n_0 P + Q) & 2\pi\Delta f A^2(n_0 N + P) \\ 2\pi\Delta f A^2(n_0 N + P) & NA^2 \end{bmatrix} \tag{B.29}$$

$$\det(\boldsymbol{J}(\boldsymbol{\alpha})) = \frac{4}{\sigma^4}(2\pi\Delta f A^2)^2 N^2(N^2-1)/12 \tag{B.30}$$

$$\boldsymbol{J}^{-1}(\boldsymbol{\alpha}) = \frac{\sigma^2}{2}\begin{bmatrix} \frac{12}{(2\pi\Delta f A)^2 N(N^2-1)} & - \\ - & - \end{bmatrix} \tag{B.31}$$

3. A is unknown and ϕ is known

$$\boldsymbol{J}(\boldsymbol{\alpha}) = \frac{2}{\sigma^2}\begin{bmatrix} N & 0 \\ 0 & (2\pi\Delta f A)^2(n_0^2 N + 2n_0 P + Q) \end{bmatrix} \tag{B.32}$$

$$\det(\boldsymbol{J}(\boldsymbol{\alpha})) = \frac{4}{\sigma^4}(2\pi\Delta f A)^2 N(n_0^2 N + 2n_0 P + Q) \tag{B.33}$$

$$\boldsymbol{J}^{-1}(\boldsymbol{\alpha}) = \frac{\sigma^2}{2}\begin{bmatrix} - & - \\ - & \dfrac{1}{(2\pi\Delta f A)^2(n_0^2 N + 2n_0 P + Q)} \end{bmatrix} \tag{B.34}$$

4. A is unknown and ϕ is unknown

$$\boldsymbol{J}(\boldsymbol{\alpha}) = \frac{2}{\sigma^2}\begin{bmatrix} N & 0 & 0 \\ 0 & (2\pi\Delta f A)^2(n_0^2 N + 2n_0 P + Q) & 2\pi\Delta f A^2(n_0 N + P) \\ 0 & 2\pi\Delta f A^2(n_0 N + P) & NA^2 \end{bmatrix} \tag{B.35}$$

(same as general case)

$$\det(\boldsymbol{J}(\boldsymbol{\alpha})) = \frac{8}{\sigma^6}\frac{(2\pi\Delta f A^2)^2 N^3(N^2-1)}{12} \tag{B.36}$$

$$\boldsymbol{J}^{-1}(\boldsymbol{\alpha}) = \frac{\sigma^2}{2}\begin{bmatrix} - & - & - \\ - & \dfrac{12}{(2\pi\Delta f A)^2 N(N^2-1)} & - \\ - & - & - \end{bmatrix} \tag{B.37}$$

2.2 Summary

Regardless of whether the amplitude is known or not

1. If ϕ is unknown then

$$\operatorname{var}(\hat{\tau}) \geq \frac{6\sigma^2}{(2\pi\Delta f A)^2 N(N^2-1)} \tag{B.38}$$

2. If ϕ is known then

$$\operatorname{var}(\hat{\tau}) \geq \frac{\sigma^2/2}{(2\pi\Delta f A)^2(n_0^2 N + 2n_0 P + Q)} \tag{B.39}$$

The CRLB is proportional to

$$\operatorname{var}(\hat{\tau}) \sim \frac{1}{N^3} \qquad \text{number of sample points} \tag{B.40}$$

$$\operatorname{var}(\hat{\tau}) \sim \frac{1}{(2\pi\Delta f N)^2} \qquad \text{periodogram resolution} \tag{B.41}$$

$$\operatorname{var}(\hat{\tau}) \sim \frac{1}{n_0^2} \sim \frac{1}{f_0^2} \qquad \text{carrier frequency} \tag{B.42}$$

$$\operatorname{var}(\hat{\tau}) \sim \frac{\sigma^2}{A^2} \sim \frac{1}{\mathrm{SNR}_\gamma} \qquad \text{phase measurement noise} \tag{B.43}$$

3 Maximum-Likelihood Distance Estimation

Goal is to maximize the likelihood function

$$p(\boldsymbol{\gamma}, \boldsymbol{\alpha}) = \frac{1}{(2\pi\sigma^2)^n} \exp\left[-\frac{1}{\sigma^2}\sum_{n=0}^{N-1} \left|\gamma[n] - A\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right]\right|^2\right] \tag{B.44}$$

Equivalently maximize the log-likelihood function

$$\begin{aligned} L_0(\boldsymbol{\gamma}, \boldsymbol{\alpha}) &= -\frac{1}{\sigma^2}\sum_{n=0}^{N-1} \left|\gamma[n] - A\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right]\right|^2 \\ &= -\frac{1}{\sigma^2}\sum_{n=0}^{N-1} \left(\gamma[n] - A\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right]\right) \\ &\quad \cdot \left(\gamma[n] - A\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right]\right)^* \\ &= -\frac{1}{\sigma^2}\sum_{n=0}^{N-1} \gamma[n]\gamma^*[n] - \gamma[n]A\exp\left[-j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right] \\ &\quad - \gamma[n]^*A\exp\left[j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right] + A^2 \\ &= -\frac{1}{\sigma^2}\sum_{n=0}^{N-1} \gamma[n]\gamma^*[n] - 2\,\mathrm{Re}\left[\gamma[n]A\exp\left[-j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right]\right] + A^2 \end{aligned} \tag{B.45}$$

Since $\gamma[n]$ is is already fixed, assuming that $A > 0$ this is equivalent to maximizing

$$L(\boldsymbol{\gamma}, \boldsymbol{\alpha}) = 2A\,\mathrm{Re}\left[\frac{1}{N}\sum_{n=0}^{N-1} \gamma[n]\exp\left[-j\left(2\pi(n_0+n)\Delta f\tau + \phi\right)\right]\right] - A^2 \tag{B.46}$$

$$= 2A\,\mathrm{Re}\left[\Gamma(\tau)\exp\left[-j\left(2\pi n_0\Delta f\tau + \phi\right)\right]\right] - A^2 \tag{B.47}$$

where

$$\Gamma(\tau) = \frac{1}{N}\sum_{n=0}^{N-1} \gamma[n]\exp\left[-j2\pi n\Delta f\tau\right]. \tag{B.48}$$

1. Initial relative phase offset ϕ is unknown

   After applying Euler's formula and taking the derivatives with respect to ϕ for τ fixed

$$\frac{\partial}{\partial\phi}L(\boldsymbol{\gamma}, \boldsymbol{\alpha}) = 2A\sin(\arg(\Gamma(\tau)) - 2\pi n_0\Delta f\tau - \phi) \triangleq 0 \tag{B.49}$$

$$\frac{\partial^2}{\partial\phi^2}L(\boldsymbol{\gamma}, \boldsymbol{\alpha}) = -2A\cos(\arg(\Gamma(\tau)) - 2\pi n_0\Delta f\tau - \phi) < 0 \tag{B.50}$$

   $L(\boldsymbol{\gamma}, \boldsymbol{\alpha})$ takes its maximum at

$$\phi' = \arg\left(\Gamma(\tau)\right) - 2\pi n_0\Delta f\tau \mod(2\pi) \tag{B.51}$$

   Substituting ϕ' into $L(\boldsymbol{\gamma}, \boldsymbol{\alpha})$

$$L(\boldsymbol{\gamma}, \boldsymbol{\alpha})\Big|_{\phi=\phi'} = 2A\,\Gamma(\tau)\Gamma^*(\tau) - A^2 \tag{B.52}$$

Therefore,

$$\hat{\tau} = \arg\max_{\tau} \Gamma(\tau)\Gamma^*(\tau) \tag{B.53}$$

which, in frequency estimation terms, is the maximum of the periodogram.

2. Initial relative phase offset ϕ is known

$$\hat{\tau} = \arg\max_{\tau} \mathrm{Re}\left[\Gamma(\tau)\exp\left[-j\left(2\pi n_0 \Delta f \tau + \phi\right)\right]\right] \tag{B.54}$$

or alternately

$$\hat{\tau} = \arg\max_{\tau} \mathrm{Re}\left[\frac{1}{N}\sum_{n=0}^{N-1} \gamma[n]\exp\left[-j\left(2\pi f_n \tau + \phi\right)\right]\right] \tag{B.55}$$

LIST OF PUBLICATIONS

Journal Papers

S. Szilvasi, B. Babjak, P. Volgyesi and A. Ledeczi: Marmote SDR: Experimental Platform for Low-Power Wireless Protocol Stack Research, *Journal of Sensor and Actuator Networks (JSAN)*, Vol. 2, No. 3, pp 631-652, 2013.

S. Szilvasi, B. Babjak, A. Ledeczi and P. Volgyesi: Towards a Versatile Wireless Platform for Low-Power Applications, *International Journal of Digital Information and Wireless Communications*, Vol. 1, No. 2, February, 2012.

Book Chapters

B. Babjak, S. Szilvasi, A. Pedchenko, M. Hofacker, E. Barth, P. Volgyesi, and A. Ledeczi: Experimental Research Platform for Structural Health Monitoring, *Advancement in Sensing Technology*, vol. SSMI 1, Berlin Heidelberg, Springer-Verlag, pp. 43-68, 2012.

S. Szilvasi, P. Volgyesi, J. Sallai, A. Ledeczi and M. Maroti: Interferometry in Wireless Sensor Networks, *Interferometry - Research and Applications in Science and Technology*, Dr. Ivan Padron (Ed.), ISBN: 978-953-51-0403-2, InTech, 2012.

Conference Papers

B. Babjak, S. Szilvasi, P. Volgyesi, O. Yapar, and P. K. Basu, Analysis and Efficient Onset Time Detection of Acoustic Emission Signals with Power Constrained Sensor Platforms, *IEEE Sensors 2013*, Baltimore, MD, USA, 2013.

B. Babjak, S. Szilvasi, and P. Volgyesi, On accurate, low-complexity quasi doppler based localization, *The Third International Conference on Digital Information and Communication Technology and its Applications (DICTAP2013)*, Ostrava, Czech Republic, 2013.

P. Volgyesi, S. Szilvasi, J. Sallai, and A. Ledeczi, External Smart Microphone for Mobile Phones, *2011 Fifth International Conference on Sensing Technology (ICST)*, Palmerston North, New Zeland, pp. 171–176, 12/2011.

S. Szilvasi, B. Babjak, A. Ledeczi and P. Volgyesi, Software-Defined Radio for Versatile Low-Power Wireless Sensor Systems, *The International Conference on Digital Information Processing and Communications*, Ostrava, Czech Republic, July, 2011.

S. Szilvasi and P. Volgyesi, An Experimental Wireless platform for Acoustic Source Localization, *Networked Digital Technologies Proceedings, Part II*, Prague, Czech Republic, Springer, pp. 289-295, July, 2010.

P. Volgyesi, J. Sallai, S. Szilvasi, P. Dutta and A. Ledeczi, Marmot: A Novel Low-Power Platform for WSNs, *Networked Digital Technologies*, Prague, Czech Republic, Springer LNCS, pp. 274–280, July, 2010.

S. Szilvasi, J. Sallai, I. Amundson, P. Volgyesi and A. Ledeczi: Configurable Hardware-Based Radio Interferometric Node Localization, *2010 IEEE Aerospace Conference*, Big Sky, MT, USA, March 6-13, 2010.

Awards and Achievements

S. Szilvasi, B. Babjak, P. Volgyesi and A. Ledeczi, 1st prize on the 23rd *Wireless @ Virginia Tech Symposium SDR Design Challenge*, Blacksburg, VA, May 29, 2013.

P. Volgyesi, M. Maroti, P. Horvath, S. Szilvasi and B. Babjak, 1st prize on the *2013 DARPA Spectrum Challenge Preliminary Tournament*, Washington, DC, September 12, 2013.

REFERENCES

[1] M. Ali, U. Saif, A. Dunkels, T. Voigt, K. Römer, K. Langendoen, J. Polastre, and Z. A. Uzmi, "Medium Access Control Issues In Sensor Networks," *Computer Communication Review*, vol. 36, no. 2, pp. 33–36, 2006.

[2] Crossbow, "Crossbow MICAz (MPR2400) Radio Module," http://www.xbow.com.

[3] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," *In Proc. of IPSN/SPOTS*, Apr. 2005.

[4] M. Maróti, P. Völgyesi, S. Dóra, B. Kusý, A. Nádas, A. Lédeczi, G. Balogh, and K. Molnár, "Radio interferometric geolocation," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, ser. SenSys '05. New York, NY, USA: ACM, 2005, pp. 1–12.

[5] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "SoftMAC - Flexible Wireless Research Platform," in *Proceedings of the Fourth Workshop on Hot Topics in Networks HotNetsIV*, 2005. [Online]. Available: http://www.cs.washington.edu/education/courses/cse590l/06wi/papers/grunwald.pdf

[6] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[7] K. Langendoen and G. Halkes, "Energy-Efficient Medium Access Control," 2005.

[8] P. J. M. Dil, B. J. Havinga, "Stochastic Radio Interferometric Positioning in the 2.4 GHz Range," in *SenSys11.* Seattle, WA, USA: ACM Press, 2011.

[9] M. Maróti, P. Völgyesi, S. Dóra, B. Kusý, A. Nádas, A. Lédeczi, G. Balogh, and K. Molnár, "Radio Interferometric Geolocation," in *Proceedings of the 3rd international conference on Embedded networked sensor systems - SenSys '05.* New York, New York, USA: ACM Press, Nov. 2005, p. 1.

[10] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, "Wireless Sensor Networks for Healthcare," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947–1960, Nov. 2010.

[11] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks For Habitat Monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications - WSNA '02.* New York, New York, USA: ACM Press, Sep. 2002, p. 88.

[12] R. Jurdak, P. Sommer, B. Kusy, N. Kottege, C. Crossman, A. Mckeown, and D. Westcott, "Enabling Multimodal Activity-based GPS Sampling," in *Proceedings of the 12th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '13. New York, NY, USA: ACM, 2013.

[13] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.* IEEE, 2005, pp. 108–120.

[14] J. Lloret, I. Bosch, S. Sendra, and A. Serrano, "A wireless sensor network for vineyard monitoring that uses image processing." *Sensors (Basel, Switzerland)*, vol. 11, no. 6, pp. 6165–96, Jan. 2011.

[15] J. Gutierrez, D. Durocher, and T. Habetler, "Applying Wireless Sensor Networks in Industrial Plant Energy Evaluation and Planning Systems," in *Conference Record of 2006 Annual Pulp and Paper Industry Technical Conference.* IEEE, 2006, pp. 1–7.

[16] A. Ledeczi, P. Volgyesi, M. Maroti, G. Simon, G. Balogh, A. Nadas, B. Kusy, S. Dora, and G. Pap, "Multiple Simultaneous Acoustic Source Localization in Urban Terrain," pp. 491–496, 2005.

[17] A. Ledeczi, T. Hay, P. Volgyesi, D. Hay, A. Nadas, and S. Jayaraman, "Wireless acoustic emission sensor network for structural monitoring," *Sensors Journal, IEEE*, vol. 9, no. 11, pp. 1370–1377, Nov. 2009.

[18] A. Warrier, S. Park, J. Min, and I. Rhee, "How much energy saving does topology control offer for wireless sensor networks? A practical study," *Computer Communications*, vol. 30, no. 14-15, pp. 2867–2879, Oct. 2007.

[19] F. Chraim and S. Karaki, "Fuel Cell applications in Wireless Sensor Networks," in *2010 IEEE Instrumentation & Measurement Technology Conference Proceedings.* IEEE, 2010, pp. 1320–1325.

[20] F. Simjee and P. H. Chou, "Everlast: Long-life, Supercapacitor-operated Wireless Sensor Node," in *ISLPED'06 Proceedings of the 2006 International Symposium on Low Power Electronics and Design.* IEEE, Oct. 2006, pp. 197–202.

[21] T. W. Davis, X. Liang, M. Navarro, D. Bhatnagar, and Y. Liang, "An Experimental Study of WSN Power Efficiency: MICAz Networks with XMesh," *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 1–14, Feb. 2012.

[22] C. Knight, J. Davidson, and S. Behrens, "Energy Options for Wireless Sensor Nodes," *Sensors*, vol. 8, no. 12, pp. 8037–8066, 2008.

[23] J. Paradiso and T. Starner, "Energy Scavenging for Mobile and Wireless Electronics," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, Jan. 2005.

[24] A. Holmes, G. Hong, K. Pullen, and K. Buffard, "Axial-flow microturbine with electromagnetic generator: design, CFD simulation, and prototype demonstration," in *17th IEEE International Conference on Micro Electro Mechanical Systems. Maastricht MEMS 2004 Technical Digest.* IEEE, pp. 568–571.

[25] J. Polastre and D. Culler, "Perpetual environmentally powered sensor networks," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE, pp. 463–468.

[26] P. Mitcheson, T. Green, E. Yeatman, and A. Holmes, "Architectures for Vibration-Driven Micropower Generators," *Journal of Microelectromechanical Systems*, vol. 13, no. 3, pp. 429–440, Jun. 2004.

[27] C. Park and P. Chou, "AmbiMax: Autonomous Energy Harvesting Platform for Multi-Supply Wireless Sensor Nodes," in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks.* IEEE, 2006, pp. 168–177.

[28] T. Le, K. Mayaram, and T. Fiez, "Efficient Far-Field Radio Frequency Energy Harvesting for Passively Powered Sensor Networks," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 5, pp. 1287–1302, May 2008.

[29] Texas Instruments, "MSP430F16x mixed signal microcontroller datasheet," http://www.ti.com/litv/pdf/slas368g, 2011.

[30] Analog Devices, "ADSP-2188 datasheet," http://www.analog.com/static/imported-files/data_sheets/ADSP-218XN_SERIES.pdf, 2006.

[31] Xilinx, "Xilinx DS160 Spartan-6 Family Overview," www.xilinx.com/support/documentation/data_sheets/ds160.pdf, 2011.

[32] Microsemi, "IGLOO Low Power Flash FPGAs with Flash*Freeze Technology Datasheet," www.actel.com/documents/IGLOO_DS.pdf, 2012.

[33] H. Belhadj, V. Aggrawal, A. Pradhan, and A. Zerrouki, "Power-Aware FPGA Design," Tech. Rep., 2009.

[34] Lattice Semiconductor, "LatticeXP2 Family Handbook," http://www.latticesemi.com/dynamic/view_document.cfm?document_id=24315, 2012.

[35] Cypress Semiconductor, "PSoC 5: CY8C52 Family Data Sheet," http://www.cypress.com/?docID=34814 , 2012.

[36] Microsemi, "SmartFusion Customizable System-on-Chip (cSoC) Datasheet," www.actel.com/documents/SmartFusion_DS.PDF, 2013.

[37] Altera, "Cyclone V Device Datasheet," www.altera.com/literature/hb/cyclone-v/cv_51002.pdf, 2013.

[38] Xilinx, "Xilinx DS190 Zynq-7000 All Programmable SoC Overview," http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf, 2013.

[39] J. L.-C. Hill, "System Architecture for Wireless Sensor Networks," Ph.D. dissertation, Jan. 2003.

[40] J. Hill and D. Culler, "Mica: A Wireless Platform for Deeply Embedded Networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, Nov. 2002.

[41] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE, 2005, pp. 497–502.

[42] P. Volgyesi, G. Balogh, A. Nadas, C. B. Nash, and A. Ledeczi, "Shooter localization and weapon classification with soldier-wearable networked sensors," in *Proceedings of the 5th international conference on Mobile systems, applications and services - MobiSys '07.* New York, New York, USA: ACM Press, Jun. 2007, p. 113.

[43] J. Beutel, "Fast-prototyping Using the BTnode Platform," in *Proceedings of the Design Automation and Test in Europe Conference.* IEEE, 2006, pp. 1–6.

[44] C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris, "WiseNET: an ultralow-power wireless sensor network solution," *Computer*, vol. 37, no. 8, pp. 62–70, Aug. 2004.

[45] Libelium, "Libelium Comunicaciones Distribuidas S.L," http://www.libelium.com.

[46] S. Szilvási and P. Völgyesi, "An experimental wireless platform for acoustic source localization," in *Networked Digital Technologies Proceedings, Part II*, Springer. Prague, Czech Republic: Springer, Jul. 2010, p. 289–295.

[47] R. Lackey and D. Upmal, "Speakeasy: the military software radio," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 56–61, May 1995.

[48] J. Mitola, "SDR architecture refinement for JTRS," in *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No.00CH37155)*, vol. 1. IEEE, pp. 214–218.

[49] K. Tan, H. Liu, J. Zhang, Y. Zhang, J. Fang, and G. M. Voelker, "Sora: High-Performance Software Radio Using General Purpose Multi-Core Processors," *Communications of the ACM*, vol. 54, no. 1, p. 99, Jan. 2011.

[50] Ettus Research, “USRP N210,” Tech. Rep. [Online]. Available: https://www.ettus.com/product/details/UN210-KIT

[51] G. J. Minden, J. B. Evans, L. Searl, D. DePardo, V. R. Petty, R. Rajbanshi, T. Newman, Q. Chen, F. Weidling, J. Guffey, D. Datla, B. Barker, M. Peck, B. Cordill, A. M. Wyglinski, and A. Agah, “KUAR: A Flexible Software-Defined Radio Development Platform,” in *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*. IEEE, Apr. 2007, pp. 428–439.

[52] J. E. Volder, “The CORDIC Trigonometric Computing Technique,” *IEEE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, Sep. 1959.

[53] “GNU Radio website,” http://gnuradio.org, 2008.

[54] J. Chapin and V. Bose, “Vanu Software Radio Waveform Research System.”

[55] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, “SODA: A Low-power Architecture For Software Radio,” *ACM SIGARCH Computer Architecture News*, vol. 34, no. 2, pp. 89–101, May 2006.

[56] H. Lee, C. Chakrabarti, and T. Mudge, “A Low-Power DSP for Wireless Communications,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 9, pp. 1310–1322, Sep. 2010.

[57] A. Duller, D. Towner, G. Panesar, A. Gray, and W. Robbins, “picoArray Technology: The Tool’s Story,” in *Design, Automation and Test in Europe*. IEEE, Mar. 2005, pp. 106–111.

[58] B. A. Dalio and K. A. Shelby, “The Implementation of OFDM Waveforms on an SDR Development Platform Supporting a Massively Parallel Processor,” in *Proceedings of the SDR ’09 Technical Conference and Product Exposition*, 2009.

[59] J. Glossner, K. Chirca, M. Schulte, H. Wang, N. Nasimzada, D. Har, S. Wang, J. A. Hoane, G. Nacer, M. Moudgill, and S. Vassiliadis, “Sandblaster Low-Power Multithreaded SDR Baseband Processor,” *IEEE Custom Integr Circ Conf CICC*, pp. 575–581, 2004.

[60] A. Lodi, A. Cappelli, M. Bocchi, C. Mucci, M. Innocenti, C. DeBartolomeis, L. Ciccarelli, R. Giansante, A. Deledda, F. Campi, M. Toma, and R. Guerrieri, “XiSystem: A XiRisc-Based SoC With Reconfigurable IO Module,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 85–96, Jan. 2006.

[61] A. Khattab, J. Camp, C. Hunter, P. Murphy, A. Sabharwal, and E. W. Knightly, “WARP A Flexible Platform for Clean-Slate Wireless Medium Access Protocol Design,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, no. 1, p. 56, Jan. 2008.

[62] M. C. Ng, K. E. Fleming, M. Vutukuru, S. Gross, and H. Balakrishnan, “Airblue: A system for cross-layer wireless protocol development,” in *Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems - ANCS ’10*. New York, New York, USA: ACM Press, Oct. 2010, p. 1.

[63] Pentek, Inc, “Pentek, inc,” http://www.pentek.com, 2008.

[64] “FlexRadio Systems,” 2013. [Online]. Available: http://www.flex-radio.com/

[65] “Nutaq Incorporated,” 2013. [Online]. Available: http://nutaq.com/en/applications/wireless/software-defined-radio

[66] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, “An Architecture for Software Defined Cognitive Radio,” in *Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems - ANCS ’10*. New York, New York, USA: ACM Press, Oct. 2010, p. 1.

[67] “USRP E100,” Ettus Research, Tech. Rep. [Online]. Available: https://www.ettus.com/product/details/UE100-KIT

[68] S. Szilvási, J. Sallai, I. Amundson, P. Völgyesi, and Á. Lédeczi, “Configurable hardware-based radio interferometric node localization,” in *2010 IEEE Aerospace Conference*, IEEE. Big Sky, Montana, USA: IEEE, Mar. 2010.

[69] P. Volgyesi, S. Szilvasi, J. Sallai, and A. Ledeczi, “External smart microphone for mobile phones,” Palmerston North, 12/2011 2011, pp. 171–176.

[70] Microsemi, “Total System Power,” http://www.microsemi.com/document-portal/doc_view/131305-total-system-power-product-information-brochure, 2008.

[71] J. Valverde, A. Otero, M. Lopez, J. Portilla, E. de la Torre, and T. Riesgo, “Using SRAM Based FPGAs for Power-Aware High Performance Wireless Sensor Networks,” *Sensors*, vol. 12, no. 3, pp. 2667–2692, 2012.

[72] Texas Instruments, “CC1000: Single Chip Very Low Power RF Transceiver,” http://www.ti.com/lit/gpn/cc1000, 2007.

[73] A. Warrier, M. Aia, and M. Sichitiu, “Z-MAC: A Hybrid MAC for Wireless Sensor Networks,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511–524, Jun. 2008.

[74] W. Ye, J. Heidemann, and D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” *ProceedingsTwentyFirst Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1567–1576, 2002.

[75] B. Krishnamachari and C. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks,” in *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.* IEEE, pp. 224–231.

[76] T. van Dam and K. Langendoen, “An Adaptive Energy-Efficient MAC Protocol For Wireless Sensor Networks,” in *Proceedings of the first international conference on Embedded networked sensor systems - SenSys ’03.* New York, New York, USA: ACM Press, Nov. 2003, p. 171.

[77] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys ’04.* New York, New York, USA: ACM Press, Nov. 2004, p. 95.

[78] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys ’06. New York, NY, USA: ACM, 2006, pp. 307–320.

[79] D. Moss and P. Levis, “BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking,” Stanford, Tech. Rep., 2008.

[80] Y. Sun, O. Gurewitz, and D. B. Johnson, “Ri-mac: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks,” in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, ser. SenSys ’08. New York, NY, USA: ACM, 2008, pp. 1–14.

[81] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, “Design and Evaluation of a Versatile and Efficient Receiver-initiated Link Layer For Low-power Wireless,” in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’10. New York, NY, USA: ACM, 2010, pp. 1–14.

[82] S. Kulkarni, “TDMA service for sensor networks,” in *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.* IEEE, 2004, pp. 604–609.

[83] I. Demirkol, C. Ersoy, and F. Alagoz, “MAC Protocols For Wireless Sensor Networks: A Survey,” *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, Apr. 2006.

[84] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC Essentials for Wireless Sensor Networks,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[85] A. El-Hoiydi and J. D. Decotignie, “WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks,” *Proceedings ISCC 2004 Ninth International Symposium on Computers And Communications IEEE Cat No04TH8769*, vol. 1, no. June 2004, pp. 244–251, 2004.

[86] S. Singh and C. S. Raghavendra, “PAMAS - power aware multi-access protocol with signalling for ad hoc networks,” *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, Jul. 1998.

[87] Y. Sun, O. Gurewitz, S. Du, L. Tang, and D. B. Johnson, “ADB: An Efficient Multihop Broadcast Protocol Based On Asynchronous Duty-Cycling In Wireless Sensor Networks,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’09. New York, NY, USA: ACM, 2009, pp. 43–56.

[88] L. Nachman, R. Kling, R. Adler, J. Huang, and V. Hummel, “The Intel Mote platform: a Bluetooth-based sensor network for industrial monitoring,” p. 61, Apr. 2005.

[89] Dynastream Innovations Inc., “The ANT+ Protocol,” http://www.thisisant.com, 2013.

[90] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, “IMOTE2: Serious Computation at the Edge,” in *2008 International Wireless Communications and Mobile Computing Conference*. IEEE, Aug. 2008, pp. 1118–1123.

[91] M. K. Simon, J. K. Omura, R. A. Sholtz, and B. K. Levitt, *Spread Spectrum Communications Handbook*, revised ed., ser. Electronic communications, M. K. Simon, Ed. McGraw-Hill Professional Publishing, 1994.

[92] I. Getting, “Perspective/navigation-The Global Positioning System,” *IEEE Spectrum*, vol. 30, no. 12, pp. 36–38, Dec. 1993.

[93] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. McGraw-Hill, 2008.

[94] R. Gold, “Optimal binary sequences for spread spectrum multiplexing (Corresp.),” *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 619–621, Oct. 1967.

[95] MEMSIC Inc., “IRIS Wireless Measurement System Datasheet,” http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf.

[96] D. Wang, L. Song, X. Kong, and Z. Zhang, “Near-Ground Path Loss Measurements and Modeling for Wireless Sensor Networks at 2.4 GHz.” *International Journal of Distributed Sensor Networks (IJDSN)*, vol. 2012, 2012.

[97] G. Mao, B. Fidan, and B. Anderson, “Wireless Sensor Network Localization Techniques,” *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, Jul. 2007.

[98] J. Hightower, C. Vakili, G. Borriello, and R. Want, “Design and Calibration of the SpotON Ad-Hoc Location Sensing System,” Tech. Rep., 2001.

[99] K. Whitehouse and D. Culler, “Calibration as Parameter Estimation in Sensor Networks,” in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications - WSNA ’02*. New York, New York, USA: ACM Press, Sep. 2002, p. 59.

[100] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket location-support system,” in *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom ’00*. New York, New York, USA: ACM Press, Aug. 2000, pp. 32–43.

[101] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, Jan. 1992.

[102] L. Ni and A. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).* IEEE Comput. Soc, 2003, pp. 407–415.

[103] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004.

[104] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Journal of Telecommunication Systems*, vol. 22, pp. 267–280, 2003.

[105] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01, vol. 01, ACM New York, NY, USA. ACM, 2001, pp. 166–179.

[106] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2. IEEE, 2000, pp. 775–784.

[107] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '99.* New York, New York, USA: ACM Press, Aug. 1999, pp. 59–68.

[108] S. Lanzisera, D. T. Lin, and K. S. J. Pister, "RF Time of Flight Ranging for Wireless Sensor Network Localization," in *2006 International Workshop on Intelligent Solutions in Embedded Systems.* IEEE, Jun. 2006, pp. 1–12.

[109] "Ubisense," 2013. [Online]. Available: http://www.ubisense.net

[110] H.-l. Chang, J.-b. Tian, T.-T. Lai, H.-H. Chu, and P. Huang, *Spinning beacons for precise indoor localization.* New York, New York, USA: ACM Press, Nov. 2008.

[111] J. Sallai, P. Volgyesi, and A. Ledeczi, "Radio interferometric Quasi Doppler bearing estimation," in *IPSN '09 Proceedings of the 2009 International Conference on Information Processing in Sensor Networks.* Washington, DC, USA: IEEE Computer Society, Apr. 2009, pp. 325–336.

[112] I. Amundson, J. Sallai, X. Koutsoukos, and A. Ledeczi, "Radio Interferometric Angle of Arrival Estimation," in *7th European Conference on Wireless Sensor Networks.* Coimbra, Portugal: Springer, 2010. [Online]. Available: http://www.isis.vanderbilt.edu/node/4139

[113] W. Li, X. Wang, and B. Moran, "Resolving RIPS measurement ambiguity in maximum likelihood estimation," in *2011 Proceedings of the 14th International Conference on Information Fusion*, 2011.

[114] W. Li, X. Wang, X. Wang, and B. Moran, "Distance Estimation Using Wrapped Phase Measurements in Noise," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1676–1688, Apr. 2013.

[115] X. Wang, B. Moran, and M. Brazil, "Hyperbolic Positioning Using RIPS Measurements for Wireless Sensor Networks," in *2007 15th IEEE International Conference on Networks.* IEEE, Nov. 2007, pp. 425–430.

[116] "IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks," pp. 1–252, 2012.

[117] "IEEE 802.11-2007, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 2007." Jun. 2007.

[118] "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (3GPP TS 36.211 version 10.0.0 Release 10)," Jan. 2011.

[119] S. Tretter, "Estimating The Frequency of a Noisy Sinusoid by Linear Regression," *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 832–835, Nov. 1985.

[120] S. Kay, "A Fast and Accurate Single Frequency Estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 1987–1990, 1989.

[121] M. Fitz, "Further Results in the Fast Estimation of a Single Frequency," *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, pp. 862–864, Feb. 1994.

[122] M. Narasimha and D. Cox, "An Improved Single Frequency Estimator," *IEEE Signal Processing Letters*, vol. 3, no. 7, pp. 212–214, Jul. 1996.

[123] L. Palmer, "Coarse Frequency Estimation Using The Discrete Fourier Transform," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 104–109, Jan. 1974.

[124] D. Rife and R. Boorstyn, "Single-Tone Parameter Estimation from Discrete-Time Observations," *IEEE Transactions on Information Theory*, vol. 20, no. 5, pp. 591–598, Sep. 1974.

[125] I. Clarkson, "Frequency estimation, phase unwrapping and the nearest lattice point problem," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258).* IEEE, 1999, pp. 1609–1612 vol.3.

[126] R. G. McKilliam, B. G. Quinn, I. V. L. Clarkson, and B. Moran, "Frequency Estimation by Phase Unwrapping," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 2953–2963, Jun. 2010.

[127] S. M. Kay, *Modern Spectral Estimation: Theory and Application.* Englewood Cliffs, NJ: Prentice Hall, 1987.

[128] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1996.

[129] P. Liu and Y. Bar-Ness, "Comparing the effect of Carrier Frequency Offset on OFDM and Single-Carrier Block Transmission in AWGN Channels," in *IEEE Globecom 2006.* IEEE, Nov. 2006, pp. 1–5.

[130] J.-J. van de Beek, M. Sandell, M. Isaksson, and P. Ola Borjesson, "Low-Complex Frame Synchronization in OFDM Systems," in *Proceedings of ICUPC '95 - 4th IEEE International Conference on Universal Personal Communications.* IEEE, 1995, pp. 982–986.

[131] T. Schmidl and D. Cox, "Low-Overhead, Low-Complexity [Burst] Synchronization for OFDM," in *Proceedings of ICC/SUPERCOMM '96 - International Conference on Communications*, vol. 3. IEEE, 1996, pp. 1301–1306.

[132] J. van de Beek, M. Sandell, and P. Borjesson, "ML Estimation of Time and Frequency Offset in OFDM Systems," *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1800–1805, Jul. 1997.

[133] K. Shi and E. Serpedin, "Coarse Frame and Carrier Synchronization of OFDM Systems: A New Metric and Comparison," *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1271–1284, Jul. 2004.

[134] J.-W. Choi, J. Lee, Q. Zhao, and H.-L. Lou, "Joint ML Estimation of Frame Timing and Carrier Frequency Offset for OFDM Systems Employing Time-Domain Repeated Preamble," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 311–317, Jan. 2010.

[135] M. Morelli, C.-C. J. Kuo, and M.-O. Pun, "Synchronization Techniques for Orthogonal Frequency Division Multiple Access (OFDMA): A Tutorial Review," *Proceedings of the IEEE*, vol. 95, no. 7, pp. 1394–1427, Jul. 2007.

[136] C. Wang, Q. Yin, and W. Wang, "An Efficient Ranging Method for Wireless Sensor Networks," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 2846–2849.

[137] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, A. V. Oppenheim, Ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1993, vol. I.