Object Tracking in mmWave Radar Networks

by

Samuel Miller

B.S., United States Military Academy (2018)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

Author .....
Department of Aeronautics and Astronautics
May 18, 2020

Certified by .....
Moe Z. Win
Professor
Thesis Supervisor

Accepted by .....
Sertac Karaman
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Object Tracking in mmWave Radar Networks

by

Samuel Miller

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Location-aware devices enable new services such as localization and tracking of objects within existing wireless communication networks like cellular mobile, Wi-Fi, and radio. To ensure these services are also available in the evolving millimeter wave (mmWave) communication infrastructure, it is important to develop algorithms that enable mmWave devices, like radars and 5G nodes, to localize and track objects. The main challenges that these algorithms must address is localizing objects that are not carrying sensing equipment, synchronizing devices exclusively via the mmWave band, and solving a data association uncertainty problem to reliably track objects of interest. Our development of the Multistatic Networking with mmWave Radar Arrays for Positioning (MiNiMAP) system solved these challenges by implementing mmWave processing in a multistatic network, scheduling, and radar synchronization algorithms. Through the use of these three algorithms in addition to Bayesian filtering, MiNiMAP is capable of tracking a single object with a network of mmWave radars. Indoor localization experiments validate MiNiMAP's overall system performance and the impact of each algorithm.

Thesis Supervisor: Moe Z. Win
Title: Professor

Acknowledgments

I would like to thank Draper for the outstanding support and freedom they have given me in pursuit of this degree. Specifically, Sheila Hemami, Martha Porter did a wonderful job coordinating the Draper Fellow program and I cannot thank my advisor Chris Yu enough for his mentorship and guidance. Also, I would like to thank Will Tomlinson for his help soldering those tiny components onto the early versions of MiNiMAP. The project would not have gotten to this point without every one of you.

To my advisor Professor Moe Z. Win, thank you for challenging me to always ask "why" and for demanding nothing short of excellence in everything from the setup of an experiment to the syntax of a sentence. Your dedication and commitment to research inspires me to work hard in pursuit of my goals. Professor Andrea Conti, thank you for always taking the time to provide me with advice and for making it feel like there was no time difference between Cambridge and Ferrara. My labmates and colleagues at MIT also deserve a sincere thank you. Zhenyu Liu, Florian Meyer, Alex Saucan, Zehao Yu, thank you for the helpful conversations. Emma Allen, Katie Burnham, George Denove, Carlos Gomez-Vega, Christian Montgomery, Connor Mullen, Jerrod Wigmore, and the members of the Black Knights IM squad also all helped support me along the journey.

Most importantly I would like to thank my family. Mom, Dad, Jacob, thank you for your encouragement. Kathleen, thank you for always being so loving and patient. I look forward to providing the same support to you as you finish Optometry school.

This research was supported, in part, by the Draper Fellow Program, by the Office of Naval Research under Grants N00014-16-1-2141, and by the Army Research Office through the MIT Institute for Soldier Nanotechnologies under Contract W911NF-13-D-0001.

Contents

List of Figures

Chapter 1

Introduction

Millimeter wave communication technology, known as 5G, is a steadily evolving technology that has many localization applications [1–5] in addition to providing its users faster communication speeds than 4G. For example, 5G shows potential to proliferate internet-of-things (IoT) applications [6–8] and make localization-of-things (LoT) [4] ubiquitous in areas such as autonomous driving [9–14], smart cities [15–17], health care [18], and industrial sensing [19–22]. The key component that enables these advantages for 5G is the use of the mmWave frequency band.

While GPS currently provides its users with their approximate location in a global reference frame [23–28], there is a lack of sub-meter positioning accuracy. Furthermore, GPS signals can be lost in use-cases that arise in challenging environments such as underground or indoors [29–34]. In these challenging environments, radar devices are used instead [35, 36, 36, 37]. However, radar devices cannot always estimate the state of objects in the environment due to unfavorable orientation or high levels of noise. As mmWave communication technology become ubiquitous, its use of the mmWave frequency band could provide users the additional service of localization and tracking of objects in these challenging environments to overcome the shortcomings of current wireless positioning technology.

In localization networks there are two types of nodes: anchors and agents [38–40]. Anchors have a known fixed position while agents have an unknown position that can be determined with inter and intra-node measurements. Inter-node measurements

refer to measurements between nodes such as positioning with radio-frequency signals via time-of-arrival [41–43], received signal strength [44–46], and angle-of-arrival [47–54]. Intra-node measurements refer to measurements obtained with respect to a single node such as inertial measurement unit (IMU) signals [55–59].

Various cooperative techniques can use these inter and intra-node measurements to achieve high-accuracy location awareness services in networks [38, 60–63]. For example, some researchers have considered the gain in positioning accuracy offered by a network of devices [1, 3, 35, 40, 64–66], but unlike the system in this project, these implementations do not create a multistatic network. Implementations from other works have created a multistatic network of devices [67–70] or devices that are capable of localizing cooperative objects in the environment with IMU and map information [55, 71–74]. Lastly, there has been research conducted on a network of mmWave radars [69, 75, 76], but this research does not solve the data association uncertainty problem.

Altogether, there is a need for a system that can be implemented on a network of low-cost mmWave devices in order to precisely and reliably track non-collaborative objects. The network localization and navigation (NLN) paradigm uses spatiotemporal cooperation techniques for position inference and could be leveraged by such a system [1, 4, 77–80]. The fundamental questions relevant to the problem are:

- How to develop a low-cost network of mmWave devices to provide an additional service of localization and tracking objects that are not carrying sensing equipment?
- How to avoid interference and synchronize mmWave radar devices that are close to one another without establishing a communication channel between each device on a non-mmWave band?
- How to solve the data association uncertainty that arises due to inherent clutter in the object's environment?

The answers to each of these questions led to the development of mmWave processing in a multistatic network configuration, scheduling and synchronization, and mod-

ified Bayesian filters. Through the development of these three NLN-based features, this project introduces an object tracking system called MiNiMAP. With a multistatic radar network, MiNiMAP is capable of resolving the states of non-collaborative objects. Furthermore, MiNiMAP implements synchronization and scheduling strategies [81–90] for enabling multiple radar devices within the network to share object position information in the multistatic setting. Lastly, despite the fact that radar systems produce both false-alarm and true detections, MiNiMAP is capable of reliably tracking one priority object with sub-meter accuracy by modifying Bayesian filtering algorithms [91–99] and other advanced statistical signal processing techniques [100–104].

Similar to experimentation in previous works [105–108], MiNiMAP verifies its theoretical basis in NLN through indoor localization experiments. Experimentation enabled MiNiMAP develop its algorithms to achieve precise and reliable object detection and tracking in a more robust manner than could have been achieved through simulation alone. The following three technical solutions for precise object tracking via low-cost mmWave networks are the key contributions of MiNiMAP:

- mmWave processing in a multistatic network configuration enables precise localization of objects due to the spatiotemporal cooperation between neighboring mmWave devices. Due to the nature of each mmWave radar's design, the network is capable of tracking objects that are not carrying any sensing equipment.

- Synchronizing and scheduling algorithms that allow devices in the network to share position information. The degree to which devices in MiNiMAP are synchronized lets each device share object state information in the physical layer. Furthermore, MiNiMAP's transmission schedule prevents interference from causing loss of object state information.

- A unique implementation of the Unscented Bernoulli filter (UBF) in addition to peak grouping and background removal algorithms enables object state inference in MiNiMAP.

Each of MiNiMAP's three key components above are grounded in theoretical principles of probability, statistics, and signal processing. The performance of NLN-based systems depend on factors such as the propagation conditions, signal bandwidth, and other fundamental limits of localization [2, 109–114]. To quantify the performance of NLN, previous research has used the Cramer-Rao lower bound [112, 113, 115, 116, 116–124]. MiNiMAP's theoretical components will be instead experimentally validated by localizing and track a moving object in an indoor environment. These experiments compare the object's mean squared positioning error in a system with MiNiMAP to the positioning error of a tracking system without MiNiMAP to emphasize the gain afforded by localizing with a network of mmWave radar devices. To further emphasize the gain afforded by localizing with a network of mmWave radar devices, the experiments compare the false-alarm and detection probability of MiNiMAP to those of a system without MiNiMAP.

The remaining sections are organized as follows: Section 2 describes the assumptions of a general multistatic object tracking system with respect to objects, radars, time, and other factors. Section 3 explains, in general, the algorithms needed for a mmWave object tracking system in a multistatic network such as radar synchronization, object detections, and state estimation over time. Section 4 explains how our specific experimental system, MiNiMAP, was set up considering the general system requirements and algorithms in section 2 and section 3. Section 5 shows how the experiments that were conducted as well as the results and implications of those experiments. Finally, Section 6 summarizes the main messages of the thesis.

Notation: Random variables are displayed in sans serif, upright fonts; their realizations in serif, italic fonts. Vectors and matrices are denoted by bold lowercase and uppercase letters, respectively. For example, a random variable and its realization are denoted by x and x; a random vector and its realization are denoted by $\mathbf{x}$ and $\boldsymbol{x}$; a random matrix and its realization are denoted by $\mathbf{X}$ and $\boldsymbol{X}$, respectively. Sets and random sets are denoted by upright sans serif and calligraphic font, respectively. For example, a random set and its realization are denoted by X and $\mathcal{X}$, respectively. The m-by-n matrix of zeros (resp. ones) is denoted by $\mathbf{0}_{m\times n}$ (resp. $\mathbf{1}_{m\times n}$); when $n = 1$,

the m-dimensional vector of zeros (resp. ones) is simply denoted by $\mathbf{0}_m$ (resp. $\mathbf{1}_m$). The m-by-m identity matrix is denoted by $\boldsymbol{I}_m$: the subscript is removed when the dimension of the matrix is clear from the context.

Chapter 2

System Description

Before exploring MiNiMAP's implementation of a general object tracking system in a multistatic radar network, it is important to first explain the key assumptions within most object tracking systems in a multistatic radar network. As a whole, the system can be broken down into two main ideas: what the system is trying to accomplish, and how the system is able to accomplish it. The system model subsection explains the assumptions inherent in the definition of what a multistatic radar tracking system is trying to accomplish, while the system architecture section describes the assumptions in the system's structure and behavior that enable it to accomplish the task at hand.

2.1 System Model

The goal of a multistatic radar network in a GPS-denied environment is to gather and share object state information in order to accurately estimate all objects' positions. In a general multistatic radar network in a GPS-denied environment, a network of radars $\mathcal{A} \in \mathbb{N}$ is considered over a discrete period of time $n = 0, 1, \ldots, T$ where T is the time period of observation and t_n is the duration of each time step. In order to be considered as a multistatic radar network, $|\mathcal{A}_\mathrm{r}| \geqslant 3$ where $|\mathcal{A}_\mathrm{r}| = 2$ is a bistatic network and $|\mathcal{A}_\mathrm{r}| = 1$ is a monostatic radar. The known state of radar $i \in \mathcal{A}_\mathrm{r}$ is $\boldsymbol{x}_i^{(n)}$ at time n and consists of its 2D position, 2D velocity, orientation, and rotational velocity. The position-related component of $\boldsymbol{x}_i^{(n)}$ at time n is denoted by $\boldsymbol{p}_i^{(n)}$ and

the velocity-related component is denoted by $\boldsymbol{v}_i^{(n)}$. The rotational velocity related component of $\boldsymbol{x}_i^{(n)}$ at time n is denoted by $\omega_i^{(n)}$. All state measurements are taken with respect to a global reference frame. The following three interdependent models are the key aspects of the system model and describe what the system is trying to accomplish.

2.1.1 Object motion model

Without using GPS, the objective of the multistatic radar network is to track a set of objects $\mathcal{A}_{\mathrm{o}} \in \mathbb{N}$ that are not carrying any sensing equipment. An object $j \in \mathcal{A}_{\mathrm{o}}$ has state $\mathbf{x}_j^{(n)}$ at time n that consists of 2D position and 2D velocity. The position-related component of $\mathbf{x}_j^{(n)}$ at time n is denoted by $\mathbf{p}_j^{(n)}$ and the velocity-related component is denoted by $\mathbf{v}_j^{(n)}$. In general, the state evolution of the object can be described by the following equation:

$$\mathbf{x}_j^{(n)} = \mathbf{f}_{\mathrm{o}}^{(n-1)}(\mathbf{x}_j^{(n-1)}) + \mathbf{u}_j^{(n-1)} \tag{2.1}$$

where $\mathbf{u}_j^{(n-1)}$ is the process noise of object j and $\mathbf{f}_{\mathrm{o}}^{(n-1)}(\cdot)$ is an object transition function dependent on how the tracked object's motion model. For example, the transition function can be a linear constant acceleration model given by:

$$\mathbf{x}_j^{(n)} = \boldsymbol{A}_{\mathrm{o}}\mathbf{x}_j^{(n-1)} + \boldsymbol{W}_{\mathrm{o}}\mathbf{u}_j^{(n-1)} \tag{2.2}$$

where $d = |\mathbf{x}_j^{(n)}|$, $\boldsymbol{A}_{\mathrm{o}} \in \mathbb{R}^{d\times d}$, and $\boldsymbol{W}_{\mathrm{o}} \in \mathbb{R}^{d\times 2}$ are given as in [125] with $\mathbf{u}_j^{(n)} = \mathcal{N}(0, \sigma_{u_j}\boldsymbol{I}_2)$. By extension, the object model can become nonlinear if $\omega_i^{(n)}$ is non-zero. Equation (2.1) can be seen as the first-order Markov process. Therefore, we assume that the prior state of all objects $j \in \mathcal{A}_{\mathrm{o}}$ at time $n = 0$ is Gaussian distributed, i.e. $f(\boldsymbol{x}_j^{(0)}) = \mathcal{N}(\boldsymbol{x}_j^{(0)}; \boldsymbol{\mu}_j^{(0)}, \boldsymbol{C}_j^{(0)})$ where the trace of $\boldsymbol{C}_j^{(0)}$ is large.

2.1.2 Radar measurement model

The purpose of every radar $i \in \mathcal{A}_{\mathrm{r}}$ is to gather observations $\mathbf{z}_i^{(n)}$ at time n from its field of view in order to detect and track objects in the environment. Depending on the radar's configuration, these observations can result from either an active setting or a passive setting. Similarly, the radar's modulation schemes affect the observations; for example, a frequency modulated continuous wave (FMCW) system produces detections from the frequency difference in the transmitted and received signal while a pulsed doppler system produces detections from the time difference between the received and transmitted signals. Regardless of the modulation scheme or configuration, the radar measurement model can be represented by the following equation:

$$\mathbf{z}_i^{(n)} = \mathbf{g}_n(\mathbf{x}_j^{(n)}) + \boldsymbol{R}_i \mathbf{v}_i^{(n)} \tag{2.3}$$

where $\mathbf{g}_n(\cdot)$ is a function mapping the state of all objects to the observations of the radar. The observation noise of radar i at time (n) is represented by the vector $\mathbf{v}_i^{(n)}$ and the variability in radar i's measurements are represented by $\boldsymbol{R}_i$. For example, in a model where the conversion between object state $\mathbf{x}_j^{(n)}$ and radar measurements $\mathbf{z}_i^{(n)}$ is linear (2.3) simplifies to the following:

$$\mathbf{z}_i^{(n)} = \boldsymbol{G}\mathbf{x}_j^{(n)} + \boldsymbol{R}_i \mathbf{v}_i^{(n)} \tag{2.4}$$

where $\boldsymbol{G}$ is the matrix representing the conversion between the object state and measurement domain. $\boldsymbol{R}_i$ could be the range resolution, velocity resolution, or angle-of-arrival resolution of the radar.

2.1.3 Tracking network model

The goal of the tracking network model is to estimate the state of the objects in the environment. By assuming these objects behave in a dynamic and stochastic manner, Bayesian tracking network models recursively predict object states and rely

on measurements from the multistatic radar network to update the prediction. A non-Bayesian tracking network models assumes the objects behave in a deterministic, time-varying manner and attempt to estimate this state via maximum likelihood estimation. Regardless of the tracking approach, the measurements gathered from the multistatic radar network are not perfect; often measurement noise, clutter, missed detection, and false alarms obscure object detection.

The object motion model and radar measurement model will also affect the implementation of the tracking network model. Whether the approach is Bayesian or non-Bayesian, the number of expected objects, and the number of radar detections that a object is capable of producing affect which models are better for minimizing the object state estimation error.

We focus on Bayesian tracking models in this thesis. The most general tracking network model is the Kalman Filter [93]. The general equations are restated here:

$$\bar{\mathbf{x}}_j^{(n)} = \boldsymbol{A}_{\mathrm{o}}\mathbf{x}_j^{(n-1)} \tag{2.5a}$$

$$\bar{\mathbf{P}}^{(n)} = \boldsymbol{A}_{\mathrm{o}}\mathbf{P}^{(n-1)}\boldsymbol{A}_{\mathrm{o}}^T + \boldsymbol{W}_{\mathrm{o}}\mathbf{u}_j^{(n-1)}\boldsymbol{W}_{\mathrm{o}}^T \tag{2.5b}$$

$$\mathbf{K}^{(n)} = \bar{\mathbf{P}}^{(n)}\boldsymbol{G}^T(\boldsymbol{G}\bar{\mathbf{P}}^{(n)}\boldsymbol{G}^T + \boldsymbol{R}_i\mathbf{v}_i^{(n)}\boldsymbol{R}_i^T)^{-1} \tag{2.5c}$$

$$\mathbf{x}_{i,j}^{(n)} = \bar{\mathbf{x}}_j^{(n)} + \mathbf{K}^{(n)}(\mathbf{z}_i^{(n)} - \boldsymbol{G}\bar{\mathbf{x}}_j^{(n)}) \tag{2.5d}$$

$$\mathbf{P}^{(n)} = (\boldsymbol{I}_{|\mathbf{z}_i^{(n)}|} - \mathbf{K}^{(n)}\boldsymbol{G})\bar{\mathbf{P}}^{(n)} \tag{2.5e}$$

where $\mathbf{P}^{(n)}$ is the covariance of the object state at time n.

The Kalman Filter produces estimates of the object's state that are less noisy than the observations, but there are still more issues that a tracking network model should address besides noisy object state observations. Nonlinear state transition and state transform matrices, clutter observations, centralized or decentralized processing, multiple observations for one object, and multiple objects each require more intricate tracking network models to produce accurate and reliable object state estimates. Depending on the radar measurement model and object motion model, the multistatic radar network implementation can add these capabilities on top of the basic Kalman

Filter to produce more reliable and accurate object state estimates.

2.2 System Architecture

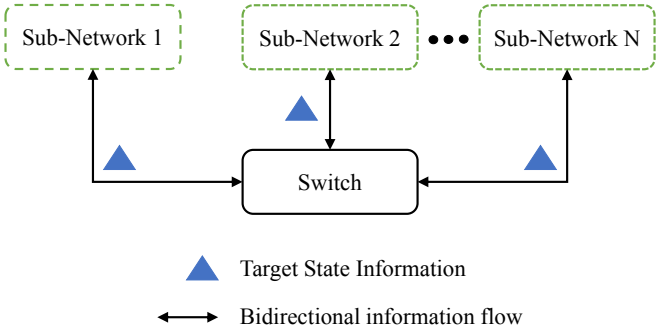
In order for a multistatic radar network in a GPS-denied environment to gather and share object state information for accurate object state estimation, a functional hardware and software structure must exist. First, all components must be able to synchronize and communicate with each other to share object information. Second, there must be a system to gather object information from the environment. Lastly, there must be a system to analyze and interpret object information gathered from the environment.

2.2.1 Synchronization and communication

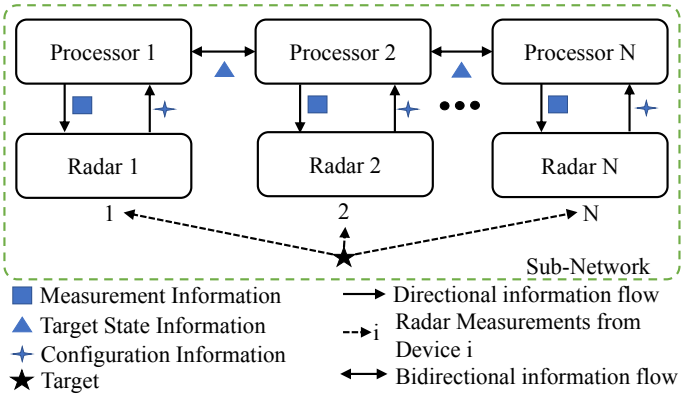
Before the multistatic network gathers any information from the object environment, it is necessary for all of the radars observing the object environment to establish a means of synchronization and communication.

Synchronization enables the radars to establish a common time-based reference. A common time-based reference allows multiple radars to share information at any of the first three levels of the operating system interconnection model: physical, data link, or network layer as shown in Fig. 2.2.1. The synchronization at these levels can be done either in through either hardware-based or software-based means. Hardware-based synchronization involves each device's physical connection to each other and a common clock. Software-based synchronization allows the devices to remain untethered to each other by synchronizing wirelessly. Without the common time-based reference, the inter-node information would be incoherent and calculating relevant object state information would become much more challenging. Furthermore, it is important to note that each information sharing layer requires a different scale of synchronization; it is much easier to synchronize on the network layer scale than it is on the physical layer scale.

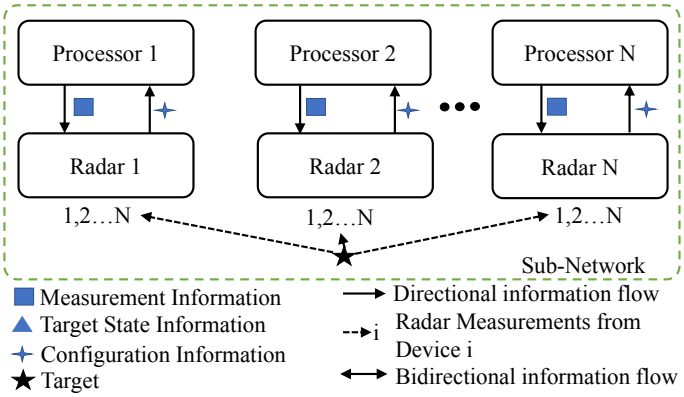
Communication enables the radars to establish other common parameters such



(a) Network layer information sharing



(b) Data link layer information sharing



(c) Physical layer information sharing

Figure 2-1: Three different levels of sharing object state information in a multistatic radar network. Information sharing is fastest at the physical layer and slowest at the network layer.

as transmit and receive power, start frequency, bandwidth, and other application-dependent parameters. Even if all of the radars in the network are perfectly synchronized, a failure to communicate these parameters can lead to dropped packets, missed detections, and false alarms. Within the network, communication can either be implicit or explicit. Implicit communication relies on preprogrammed or preconfigured parameters while explicit communication requires the devices to exchange messages in order to configure these parameters before testing.

2.2.2 Data gathering

Once the radars have a means of synchronizing and communicating object state information, they must establish a means of gathering this object state information. The challenges of data gathering are described in the following paragraphs at a high level because the system model has a large effect on how to implement solutions to the following issues.

The first issue is the configuration of radar parameters such as interference avoidance. Whether the multistatic radar network avoids interference via time-division multiplexing, phase multiplexing, or another multiplexing strategy, it is important to minimize the loss of object state information from interference.

The second issue addresses the medium of sharing the object state information between devices. Devices can share information at either the physical, data link, or network layer with each layer having distinct advantages and disadvantages.

The third and final issue is that devices can share information at each of those layers in an online or offline manner. Sharing information in an online manner would require real-time data analysis so that the information can be exported from the device to neighboring devices. Offline data sharing removes this time-based data processing requirement.

2.2.3 Data analysis

After gathering observations from the object environment, the multistatic radar network must analyze and interpret the observations. Once again, the system model largely determines the means of analyzing the observations according to both the radar measurement model and object motion model. The following issues address the common features that any multistatic radar network architecture architecture must address in order to enable proper data analysis.

The data analysis requirements change for a multistatic radar network that conducting online data processing rather than offline data processing. Due to its time-sensitive nature, online data processing is generally more restrictive in the time and memory needed to analyze radar data. All of the information sharing strategies in Fig. 2.2.1 are still viable from the data analysis perspective, but the physical layer information sharing strategy is better for a time-constrained multistatic radar network.

Lastly, the data analysis can be done in either a centralized manner, decentralized manner, or combination of the two. Centralized data analysis involves the collection of the radar network's data on one node that produces object state estimates while decentralized analysis relies on each individual node to produce its own estimates.

Chapter 3

Algorithms

The previous section explained the key assumptions within most object tracking system in a multistatic radar network, and this section continues by explaining the key algorithms needed to track objects in a multistatic radar network. Due to the fact that the algorithms are largely affected by the problem definition in the system model, this section constrains the object motion model, radar measurement model, and tracking network model to develop appropriate algorithms. The system architecture is left unconstrained so that these algorithms can be implemented with a variety of hardware and software solutions for a multistatic radar system. The three general algorithms needed for most tracking systems in a multistatic radar network are radar synchronization, multistatic detection, and modifications to traditional object tracking filters.

3.1 System Constraints Model

Before developing the synchronization, multistatic detection, and tracking network algorithms, it is necessary to constrain the system model. The system model affects which algorithms are the most effective upon implementation; the multistatic detection strategy that works for continuous wave radar systems will not work for pulse radar systems.

Throughout the rest of the thesis, the object tracking model is constrained to a

single object, i.e. $|\mathcal{A}_o| = 1$, that is not carrying any sensing equipment. The linear constant acceleration model in (2.2) applies to this object. The radar measurement model is constrained to a mmWave FMCW system that can observe object range, doppler velocity, and angle-of-arrival. The tracking network model assumes that the object can produce one or more detections on the radar and that the object can either exist or not exist within the radar network's field of view.

3.2 Multistatic Detection

The key advantage of a multistatic radar network is the gain in spatial diversity of the radars in the network. A multistatic radar network that shares object state information at the physical layer should have performance gains over the same network that only fuses monostatic detections at either the data link or network layer. Previous work in multistatic radar networks address physical layer object state information sharing [69, 126] and even with FMCW [127]. Neither of these approaches leverage the mmWave frequency band, so this section will cover how to produce range, velocity, and angle-of-arrival detections in a mmWave FMCW multistatic radar network.

The process of transmitting and receiving a mmWave FMCW signal on one radar within the multistatic network is explained in [128–130].

$$\boldsymbol{x_{1,i}} = A_1 \cos((S\boldsymbol{t_i} + 2\pi f) \circ \boldsymbol{t_i} + \phi_1) \tag{3.1a}$$

$$\boldsymbol{x_{2,k}} = x_{1,k} \tag{3.1b}$$

$$\boldsymbol{x_{2,i}} = 0 \quad \forall i \neq k \tag{3.1c}$$

$$\boldsymbol{x_{3,i}} = A_3 \cos((S(\boldsymbol{t_i} - \boldsymbol{\tau_k}) + 2\pi f) \circ \boldsymbol{t_i} + \phi_3) \tag{3.1d}$$

$$\begin{aligned} \boldsymbol{x_{4,i}} = \gamma(A_1, A_3)[&\cos(S\boldsymbol{\tau_k} \circ \boldsymbol{t_i} + \phi_1 - \phi_3) \\ &+ \cos(2S\boldsymbol{t_i} - S\boldsymbol{\tau_k} + 4\pi f \boldsymbol{t_i} + \phi_1 + \phi_3)] \end{aligned} \tag{3.1e}$$

The $\odot$ notation represents the Hadamard product. (3.1a) is the FMCW signal of radar i at the frequency synthesizer while (3.1b) and (3.1c) is the signal at the receive antenna. The received FMCW signal is represented by (3.1d) and (3.1e) is the result of

the convolution of (3.1a) and (3.1d) at the mixer known as the intermediate frequency (IF) signal. The radar index k is the transceiver while all other radars are passive receivers. S represents the chirp slope of the FMCW signal on all radars, $\boldsymbol{\tau_k}$ represents the time difference between the start of the chirp transmission and time at which the chirp was received on each radar i, and f represents the starting frequency of the chirp on all radars. $\boldsymbol{t_i}$ is the vector of all times at which the signal is sampled during the test, ϕ_1 is the designed phase shift while ϕ_3 is a function of the signal's propagation path. Following the mixer, the IF signal then passes through a low-pass filter with a cutoff frequency f_c and is sampled by the analog to digital converter (ADC). To prevent aliasing, $f_\mathrm{c} \leqslant f_\mathrm{s}/2$ where f_s is the sampling frequency of the ADC.

Given the filtered and sampled IF signal, [129–131] each walk through how to detect the range, velocity, and angle-of-arrival of an object in the monostatic case. Here are the key equations for object detection in monostatic radar:

$$\delta d_\mathrm{m} = \frac{c}{2ST_\mathrm{c}} \tag{3.2a}$$

$$\delta v_\mathrm{m} = \frac{\lambda}{2C_\mathrm{f}T_\mathrm{c}} \tag{3.2b}$$

$$\delta \theta_\mathrm{m} \cong \frac{2}{N_\mathrm{rx}N_\mathrm{tx}} \tag{3.2c}$$

where δd_m, δv_m, and $\delta \theta_\mathrm{m}$ are the radar's resolution of the range, velocity, and angle-of-arrival, respectively, in the monostatic case. T_c is the time period of a chirp, λ is the carrier wavelength, C_f is the number of chirps in a frame, N_rx and N_tx are the number of receive and transmit antennas on each radar, respectively. A *frame* is simply a fixed series of chirps.

Once the entire frame is sampled, each radar has a $C_\mathrm{f} \times (f_\mathrm{s}T_\mathrm{c}) \times (N_\mathrm{rx}N_\mathrm{tx})$ matrix $\boldsymbol{M}$ of I/Q samples. The discrete fast Fourier transform (FFT) on each dimension of the $\boldsymbol{M}$ matrix results in a frequency domain signal. An FFT of the I/Q samples in each chirp results in the frequency of the IF signal which is linearly related to the object's range. An FFT of the $s \in \{1, 2, \cdots, f_\mathrm{s}T_\mathrm{c}\}$ I/Q sample over all chirps in the frame results in the temporal phase shift which is linearly related to the velocity of the

object. Finally, the FFT of $s \in \{1, 2, \cdots, N_{\text{tx}}N_{\text{rx}}\}$ I/Q sample results in the spatial phase shift which is nonlinearly related to the angle-of-arrival of the object. There are numerous means of detecting whether or not an object signature in range, velocity, or angle-of-arrival exist [132]. Once the index of the detected objects is calculated by these algorithms, multiplying the range, velocity, and angle-of-arrival detection index by the corresponding equation (3.2a),(3.2b), and (3.2c) yields the range, velocity, and angle-of-arrival detections.

The detection of range, velocity, and angle-of-arrival with FMCW radar is similar in the multistatic case. Instead of processing the reflection from a co-located transmitter, the receiver processes the reflection of the spatially separated transmitter. Due to the fact that the chirp does not originate on the receiving device, there is a synchronization problem addressed in the next section. The discussion in this section assumes that the transmitting and receiving devices are synchronized to a sufficient enough degree to allow the reception of a line-of-sight (LOS) peak on the receiving device.

The LOS peak represents the direct path between the transceiver and passive receiver. The LOS peak is always the first signal received from the transceiver because it travels the shortest path between the two devices. Also, the LOS peak is always the signal with the greatest received power due to two factors: it travels the shortest distance so the power doesn't attenuate as much as a multipath reception due to the Friis equation, and all multipath receptions will lose power after reflecting off of objects in the environment.

Using the fact that the LOS peak is always the first received signal from the transceiver and that it has much greater power than other received signal, the LOS peak becomes easy to detect. However, it is not as straightforward as simply finding the maximum received power from the frequency domain of the signal. This is because it is possible that the transceiver is far enough away to attenuate the transmitter's power enough so that it has less power than the noise from close by sources. Therefore, a robust approach to detecting the LOS peak is using a detection algorithm similar to CFAR-CA [132]. The advantage of CFAR-CA is that instead of setting a constant

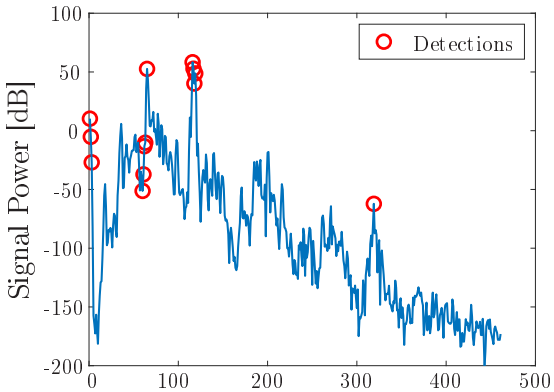
power detection threshold, it sets a constant false-alarm detection threshold. This allows the algorithm to adapt the threshold power at a particular received frequency relative to neighboring frequencies. Other techniques for detection threshold design have soft-decision thresholds rather than hard-decision thresholds which can improve detection accuracy [54,133]. The downside of these techniques is that they are highly complex and may require significant processing power.

Detecting the frequency of the LOS peak allows the user to accommodate for signal propagation delay and remaining synchronization offset. Fig. 3-1(c) shows a typical mmWave network processing scenario for a single chirp within a frame. Fig. 3-1(a) shows the transceiver's monostatic reception of the chirp and Fig. 3-1(b) shows a spatially separated passive receiver's multistatic reception of the same chirp. The monostatic chirp is inherently synchronized with itself due to the design of the mixer circuit, but there is a slight synchronization offset in addition to propagation delay that causes the LOS peak in Fig. 3-1(b) to appear at approximately 2 MHz.

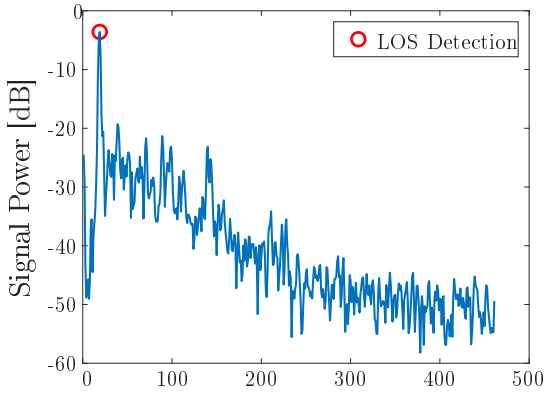
$$\tau_{\mathrm{s}} = \frac{f_{\mathrm{los}}}{S} - \frac{\|\boldsymbol{p}_k - \boldsymbol{p}_i\|}{c} \quad \text{s.t.} \quad i \neq k \tag{3.3}$$

The detected LOS frequency is represented by f_{los}. Using (3.3), the user can calculate the time that should be subtracted from all subsequent multistatic detections. Either the measure of time or distance can be used because they are linearly related quantities, so it depends on whichever quantity the user is more comfortable with implementing. The user can obtain the distance separating the receiver and transmitter by either explicitly calculating it via radar detection, internal measurement unit estimation, or by assuming perfect knowledge of all radar positions. All of the multistatic detections of interest are after the LOS peak and should only be detected after the shift in time or distance is complete. This is because the power of the LOS peak can mask bistatic range detections close to the bistatic range, so once the LOS peak frequency is calculated, the power level of the remaining samples can be adjusted to avoid missed detections.

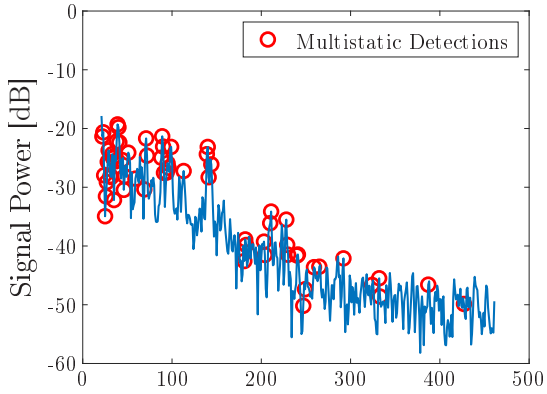
After shifting the frequency domain signal such that the LOS peak is at 0 Hz, the



(a) Monostatic chirp



(b) Multistatic chirp



(c) Shifted multistatic chirp

Figure 3-1: Figure (a) shows typical range detections for a monostatic chirp, figure (b) shows the LOS detection for a multistatic chirp and figure (c) shows the shifted LOS peak and range detctions for a multistatic chirp.

mmWave processing is the similar to the monostatic case. The calculation for the angle-of-arrival is the same as in monostatic, but the range and velocity are slightly different:

$$\delta d_{\mathrm{b}} = \frac{c}{ST_{\mathrm{c}}} \tag{3.4a}$$

$$\delta v_{\mathrm{b}} = \frac{\lambda}{C_{\mathrm{f}} T_{\mathrm{c}}} \tag{3.4b}$$

In the multistatic case, we calculate bistatic range which is the distance from the transmitter to the object to the receiver and the velocity corresponds to the rate of change of the bistatic range rather than the monostatic range. If an object traveled in a perfect circle around a monostatic radar, there would be no velocity. Similarly, if an object traveled in a perfect elliptical around a bistatic radar, there would be no velocity. Combining these velocity measurements helps ensure that an object that is truly not moving, but this will be discussed further in the tracking network model section. The velocity calculation is also much more difficult in the bistatic case than in the monostatic case due to the degree of synchronization required. Due to the fact that the bistatic velocity of the object is detected via constant phase rotation across chirps in a frame, the transmitting and receiving devices must be synchronized enough to prevent a constant phase rotation from clock drift. This will be discussed further in the synchronization section.

3.3 Radar Synchronization

The unique challenge of radar synchronization for multistatic radar networks in GPS-denied environments is the ability to synchronize multiple radars exclusively via mmWave frequencies while avoiding interference. Previous multistatic radar systems [134, 135] rely on other frequency bands like GPS for synchronization of devices, and other works [69, 126] use a hardware triggering means to synchronize the radar network. The hardware triggering technique is more specific to the multistatic radar network's architecture, so instead this section will address two wireless mmWave syn-

chronization techniques. The first wireless technique is how to shift the LOS peak frequency to be less than f_c given the multistatic detection procedure. The second wireless technique is, as explained in the multistatic detection section, correcting the final synchronization offset. There are certain requirements for the final synchronization offset correction to be successful upon LOS peak detection by the receiving device. Finally, this section will address the need for a scheduling or collision detection algorithm to avoid interference when sharing information between multiple devices in the network.

Previous wireless mmWave synchronization techniques are the reference broadcast time synchronization [136,137] and time synchronization protocol for sensor networks [138]. The challenge in implementing these previous solutions to the problem of shifting the LOS peak frequency to be less than f_c is that these synchronization techniques rely on message exchange. The mmWave multistatic radar network architecture in 2 does not include a means of passing messages via the mmWave band; it is only capable of transmitting and receiving FMCW chirps. Therefore, our synchronization solution must exclusively use the presence of the LOS peak or lack thereof to determine the degree of synchronization.

Our solution to wireless synchronization in a passive multistatic radar network is to designate one device in the network as the master and the rest as slaves. The responsibility of the slaves is to synchronize with the master's transmission. We assume that the master and slaves are both configured with the same parameters as far as frame period, chirp start frequency, chirp slope, and chirp period. By making this assumption, it makes it possible for the slave devices to receive and process the master's transmission. To find the LOS peak of the master, the slaves "scan" transmission times by adjusting the starting times of each chirp within the frame until the entire frequency band has been "scanned" for the presence of the LOS peak. Once the LOS peak is detected in one of the chirps, the slave will be able to calculate the timing offset between itself and the master. The amount of time it will take in order to discover the LOS peak depends on the configuration, but the worst-case scan time is as follows:

$$T_{\mathrm{s}} = \frac{T_{\mathrm{f}}^2 S}{f_{\mathrm{c}}} \tag{3.5}$$

where $B_{\mathrm{adc}} = \frac{S}{f_{\mathrm{c}}}$ is the bandwidth of the radar's ADC. This method can be costly if the frame period is too long, chirp slope too great, or ADC bandwidth too low.

Alternatively, the user can configure both the master and slave radars with a synchronization frame. The synchronization frame should have the same bandwidth as the ADC and a longer period than detection frames so that the slave only needs to transmit one synchronization frame to detect the master. The downside of this approach is the lower information throughput because no detections are produced during the synchronization frame.

The technique of adjusting for the final offset synchronization once the LOS peak is detected is explained in the previous section, but there are requirements to the success of these synchronization strategies. Due to the clock offset of each radar in the network, there is some synchronization noise. This variability means that software-based synchronization must be done frequently in order to ensure that the slave can still detect the master device. Our experiments do not address software-based synchronization, so future studies will reveal this synchronization refresh rate.

In the multistatic scenario, the peak detections do not always have the same arrival time at the device due to timing inconsistencies between radars. Therefore, the radar needs to have a range resolution much less than half of its operating wavelength to detect velocity in the bistatic case. This fine range resolution enables the device to precisely time the arrival of detections such that the phase shift of those detections over subsequent chirps can still be determined even if the time of arrival varies across chirps. In an imprecise timing system, the phase shift of the subsequent detection peaks would be greater than $\frac{\pi}{2}$ which means the velocity of the object cannot be reliably detected. Even if a system does not meet the requirements for detecting velocity in the multistatic case, the object's velocity can still be estimated. This will be covered in more detail in the object tracking section.

The last aspect of wireless radar synchronization in a multistatic radar network is the radar's need for an accurate means of starting transmissions at a specific time. For example, if the device determines that it should wait 5ns in order to synchronize with the master, it must be capable of starting exactly 5ns later. This timing resolution needs to be just as exact as the timing detection in the LOS peak if you want to detect velocity, and it needs to be just as exact as the timing detection in synchronizing with the master if you want to do software synchronization. Therefore, The scan-time step, T_{scan}, of the device must be greater than the device timing accuracy. $T_s = f_c/S$.

Whether the devices in the multistatic radar network synchronize using wireless means as described in this section or hardware-based means, the devices within the network still must be able to avoid interference. The two primary strategies for avoiding interference in the multistatic radar network is time division multiple access (TDMA) or phase division multiple access (PhDMA) schedules. The implementation is case-dependent, but there are advantages and disadvantages to each strategy. PhDMA guarantees the highest throughput in the network because multiple devices can transmit at the same time, but if the devices do not have a high likelihood of distinguishing multiple devices' transmissions, there will be high packet loss. TDMA ensures a lower packet loss rate at the expense of a lower throughput.

In conclusion, synchronization is a necessary means to enable TDMA or PhDMA scheduling to avoid interference and also to ensure appropriate detections in the multistatic object case.

3.4 Object Tracking Modifications

Once the multistatic radar network has achieved both synchronization and multistatic detection, a reliable tracking algorithm should produce object state estimates from the noisy, cluttered set of detections. Given the constraints in the first subsection of 3, there are still a variety of algorithms that the user can implement to produce object state estimates. This section will address those algorithms and the general modifications that must be made to those algorithms in a multistatic radar network. Then,

the section will describe methods for implementing a less complex object tracking filter.

Given a single moving object that will be tracked with a mmWave multistatic radar network, there are multiple approaches to tracking an object. Generally, there is a tradeoff between the most robust approach that can track an object in any scenario and reduction in complexity that requires fewer data points or processing time. In general, tracking algorithms address the problems of noise in both the object and measurement models, nonlinearities in the object and measurement models, clutter, existence, extent tracking, and number of object tracking. While the most robust algorithm incorporates each of these aspects, it will also be highly complex to implement. For example, the unscented Bernoulli filter (UBF) [94–96] is one type of Bayesian filters [97,98,102–104,139] that addresses nonlinear point object tracking for a single object that has an existence probability. Other approaches like probabilistic multiple hypothesis tracking (PMHT) [140] and joint probabilistic data association (PDA) [141] solve the problem of multiple object tracking while others [96, 142, 143] discuss extended rather than point object tracking.

Regardless of the algorithm chosen above, they are Bayesian and largely based on the Kalman Filter [99]. The Kalman Filter consists of iterating between predict and update steps. The object state is first predicted according to the previous states and the object motion model. The mean and covariance of this prediction is then updated with radar measurements. This process is repeated for every time step, but there are slight modifications to this predict-update structure that are necessary for implementation in the multistatic radar network. First, multiple radar measurements of the object occur at every given time step. Not only does the transceiver produce a monostatic measurement, but all passive multistatic receivers also produce a multistatic measurement. Therefore, at every time step there are $|\mathcal{A}|$ radar measurements. This alters both the predict and update steps in the filter.

First, the predict step is changed because the object will only move after $|\mathcal{A}_\mathrm{r}|$ measurements have been incorporated. Thus, the first predict step will be a traditional Kalman Filter prediction based on the object motion model $\boldsymbol{A}_\mathrm{o}$ according to (2.5a)

and (2.5b); however, subsequent predict steps will not occur until after $|\mathcal{A}_\mathrm{r}|$ updates. Alg. 1 describes this process. The following equations are the restated versions of (2.5a) and (2.5b) for the single object tracking case.

$$\bar{\mathbf{x}}_1^{(n)} = \boldsymbol{A}_\mathrm{o}\mathbf{x}_1^{(n-1)} \tag{3.6a}$$

$$\bar{\mathbf{P}}^{(n)} = \boldsymbol{A}_\mathrm{o}\mathbf{P}^{(n-1)}\boldsymbol{A}_\mathrm{o}^T + \boldsymbol{W}_\mathrm{o}\mathbf{u}_1^{(n-1)}\boldsymbol{W}_\mathrm{o}^T \tag{3.6b}$$

These update steps take the observation in either the monostatic or multistatic mode and update the predicted mean and covariance depending on the set of observations. Therefore, the update step is changed because the measurement model changes for a monostatic and multistatic radar. The monostatic measurements include the range, doppler velocity, and angle-of-arrival while the multistatic measurements include the bistatic range, doppler bistatic velocity, and angle-of-arrival. Due to these differences, the transformation from the object state to the observation state is different for a monostatic and multistatic device. If this conversion is not done for the right sensors or in the right way, the wrong detections could be associated with the object, causing the filter to produce poor object state estimates.

The following equations describe how to transform the object state into the observation domain in the monostatic case:

$$\mathbf{z}_i^{(n)} = \begin{bmatrix} z_{i,1}^{(n)} \\ z_{i,2}^{(n)} \\ z_{i,3}^{(n)} \end{bmatrix} \quad \text{s.t.} \quad i = k$$

$$z_{i,1}^{(n)} = \|\mathbf{p}_1^{(n)} - \boldsymbol{p}_i^{(n)}\| \tag{3.7a}$$

$$z_{i,2}^{(n)} = \frac{(\mathbf{v}_1^{(n)})^\mathrm{T}(\mathbf{p}_1^{(n)} - \boldsymbol{p}_i^{(n)})}{\|\mathbf{p}_1^{(n)} - \boldsymbol{p}_i^{(n)}\|} \tag{3.7b}$$

$$z_{i,3}^{(n)} = \tan^{-1}\left(\frac{\mathsf{x}_{1,1}^{(n)} - x_{i,1}^{(n)}}{\mathsf{x}_{1,2}^{(n)} - x_{i,2}^{(n)}}\right) \tag{3.7c}$$

where $z_{i,1}$,$z_{i,2}$, and $z_{i,3}$ are the range, velocity, and angle-of-arrival components of the

measurement vector $\mathbf{z}_i^{(n)}$ from radar i.

Due to the nature of the multistatic case, the range and velocity conversion is different while the angle of arrival equation remains the exact same:

$$\mathbf{z}_i^{(n)} = \begin{bmatrix} z_{i,1}^{(n)} \\ z_{i,2}^{(n)} \\ z_{i,3}^{(n)} \end{bmatrix} \quad \text{s.t.} \quad i \neq k$$

$$z_{i,1}^{(n)} = \|\mathbf{p}_1^{(n)} - \boldsymbol{p}_k^{(n)}\| + \|\mathbf{p}_1^{(n)} - \boldsymbol{p}_i^{(n)}\| \tag{3.8a}$$

$$z_{i,2}^{(n)} = \frac{(\mathbf{v}_1^{(n)})^{\mathrm{T}}(\mathbf{p}_1^{(n)} - \boldsymbol{p}_k^{(n)})}{\|\mathbf{p}_1^{(n)} - \boldsymbol{p}_k^{(n)}\|} + \frac{(\mathbf{v}_1^{(n)})^{\mathrm{T}}(\mathbf{p}_1^{(n)} - \boldsymbol{p}_i^{(n)})}{\|\mathbf{p}_1^{(n)} - \boldsymbol{p}_i^{(n)}\|} \tag{3.8b}$$

$$z_{i,3}^{(n)} = \tan^{-1}\Big(\frac{\mathsf{x}_{1,1}^{(n)} - x_{i,1}^{(n)}}{\mathsf{x}_{1,2}^{(n)} - x_{i,2}^{(n)}}\Big) \tag{3.8c}$$

as you can see, the only difference between the monostatic and multistatic models is that the multistatic model must address both the path from the transmitting device to the object as well as the path from the object to the receiving device. In the monostatic case, these two paths are identical and greatly simplify the model. Due to the fact that these paths are not identical in the multistatic case, the first value in the measurement vector is the bistatic range instead of the receiver-object range. Previous work addresses the conversion from bistatic range and angle-of-arrival to receiver-object range [144], but the filter does not need to make this conversion because it is just comparing the transformed object state to the observations.

The update step is also changed because when the model is updating with observations from different devices because the geometrical perspective is different for each device. Therefore, the update step must reorient the object's state according to the geometrical perspective of the radar producing the observation at that time step. This is important because each radar device collects measurements with reference to itself, so comparing the observations to a global position would not work. In order to incorporate observations from different devices into the tracking network model, we must first designate any radar as the perspective radar $\mathbf{x}_p \quad \text{s.t.} \quad p \in \mathcal{A}$. Designating a radar as the perspective radar means that the network model will produce object

state estimates from the perspective of that radar. Should the user wish to view the object state estimates from all radars, the following methods can also be applied to the output to obtain the object state estimate from any radar's perspective.

$$\hat{\mathbf{x}}_1^{(n)} = \begin{bmatrix} \hat{\mathsf{x}}_{1,1}^{(n)} \\ \hat{\mathsf{x}}_{1,2}^{(n)} \\ \hat{\mathsf{x}}_{1,3}^{(n)} \\ \hat{\mathsf{x}}_{1,4}^{(n)} \end{bmatrix} \quad \text{s.t} \quad i \neq p$$

$$\hat{\mathsf{x}}_{1,1}^{(n)} = \sin(\psi_1)\|\mathbf{p}_1^{(n)} - \boldsymbol{p}_i^{(n)}\| \tag{3.9a}$$

$$\hat{\mathsf{x}}_{1,2}^{(n)} = \cos(\psi_1)\|\mathbf{p}_1^{(n)} - \boldsymbol{p}_i^{(n)}\| \tag{3.9b}$$

$$\hat{\mathsf{x}}_{1,3}^{(n)} = \sin(\psi_2)\|\mathbf{v}_1^{(n)}\| \tag{3.9c}$$

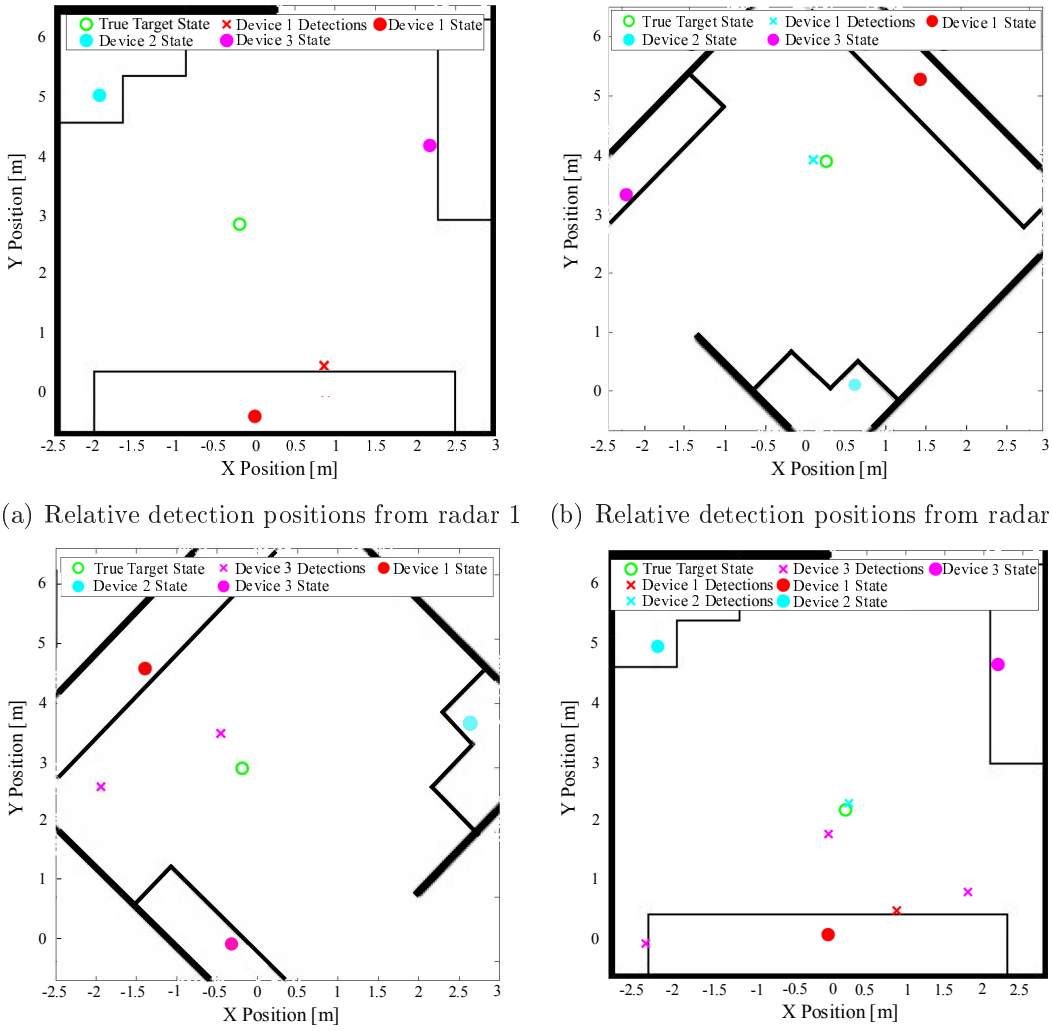
$$\hat{\mathsf{x}}_{1,4}^{(n)} = \cos(\psi_2)\|\mathbf{v}_1^{(n)}\| \tag{3.9d}$$

$$\psi_1 = \tan^{-1}\left(\frac{\mathsf{x}_{1,2}^{(n)} - x_{p,2}^{(n)}}{\mathsf{x}_{1,1}^{(n)} - x_{p,1}^{(n)}}\right) + |\omega_p^{(n)}| + \Delta \tag{3.9e}$$

$$\psi_2 = \tan^{-1}\left(\frac{\mathsf{x}_{1,3}^{(n)}}{\mathsf{x}_{1,4}^{(n)}}\right) - \omega_p^{(n)} \tag{3.9f}$$

where Δ is the degree adjustment necessary depending on the location of the non-perspective device relative to the perspective device. Once the perspective is shifted, the tracking network model proceeds with the standard predict-update steps and applying the monostatic and multistatic models to their respective observations.

To avoid the complexities of implementing a filter that solves every tracking challenge, the user can implement a less complex filter with other signal processing functions that simplify the object environment. For example, rather than implementing the complex belief propagation (BP) algorithm [104] to solve nonlinearities in the object or measurement model, the user may instead implement sigma point belief propagation (SPBP) [139]. However, there are trade-offs in performance and complexity with these simpler solutions; the optimal algorithm depends on the system requirements and implementation. Even if the devices are not capable of observing a



(a) Relative detection positions from radar 1

(b) Relative detection positions from radar 2

(c) Relative detection positions from radar 3

(d) Reoriented network detections

Figure 3-2: Examples demonstrating the necessity of reorienting the network's detections to a single perspective device

specific object state like velocity, the filter is still capable of estimating the object's velocity from it's change in position over time.

Algorithm 1 – Object State Estimation in a Multistatic Network

1: **for** $n \geqslant 0$ **do**
2: Acquire observations from every sensor in the network $\{\boldsymbol{z}_i^{(n)}\}_{i\in\mathcal{A}}$ at time n according to the multistatic detection procedure.
3: Predict the object state based on the previous time step and the object motion model (3.6a).
4: **for** $i \in \mathcal{A}_\mathrm{r}$ **do**
5: If the measurement $\boldsymbol{z}_i^{(n)}$ is not from the perspective sensor, propagate the sigma points through the reorientation equations (3.9a).
6: If the measurement $\boldsymbol{z}_i^{(n)}$ is monostatic, propagate the sigma points through the monostatic measurement model (3.7a). Otherwise, propagate the sigma points through the multistatic measurement model (3.8a).
7: Assess the likelihoods of the observations and update the object state accordingly.
8: **end for**
9: **end for**

Chapter 4

Implementation

So far, section 2 described the key assumptions within any object tracking system in a multistatic radar network and section 3 explained specific algorithms for a mmWave FMCW multistatic radar network that tracks a single object without sensing equipment. This section will now fully constrain both the system model and system architecture from section 2 and introduce MiNiMAP. The subsection on the radar network architecture will explain the hardware and software setup that enables MiNiMAP to track an object that is not carrying any sensing equipment. Following the explanation of MiNiMAP's architecture, the information processing subsection will describe how MiNiMAP implements the algorithms proposed in 3 in addition to other processes to reliably track the object in the mmWave FMCW multistatic radar network.

4.1 MiNiMAP Architecture

This subsection explains the hardware and software components of MiNiMAP that enable it to track an object in a mmWave FMCW radar multistatic network. MiNiMAP is a hardware-independent, passive system that is capable of leveraging a network of local mmWave sensors to provide enhanced localization and tracking information for users. In this project, we used three AWR1642BOOST (AWR) radars to model a network of mmWave devices. The three main components that comprise MiNiMAP are exactly as described in section 2: synchronization, data gathering, and data analysis.

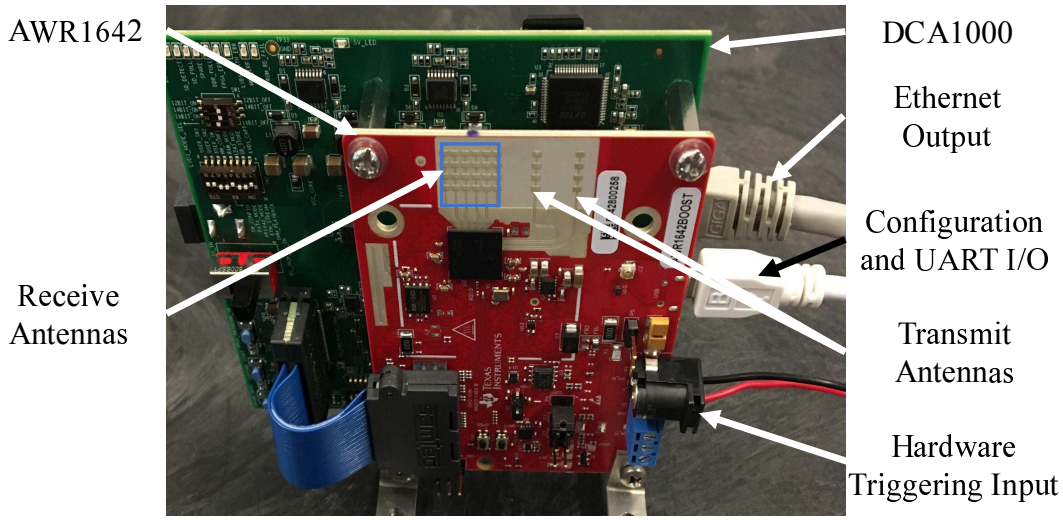


Figure 4-1: Diagram of the AWR and DCA devices as well as their inputs and outputs.

To build MiNiMAP, we used Texas Instruments' (TI) AWR automotive radar shown in Fig. 4-1 [145]. The function of the AWR is to transmit and receive frequency modulated continuous wave (FMCW) chirps [70, 134]. The AWR is an ideal choice for this implementation due to its low cost (roughly $300) and small form factor (the size of a standard smartphone). In addition to its low power requirements of approximately 5 Watts, the form factor and low-cost makes the AWR a versatile and scalable radar network solution. Furthermore, the AWR operates in the 77-81 GHz extremely high frequency (EHF) mmWave band, enabling short range tracking applications and mmWave multistatic radar network experimentation. However, the AWR is not perfect. The following sections will address how MiNiMAP circumvents some of the AWR's design shortcomings in order to implement a low-cost mmWave FMCW multistatic radar network that tracks objects which do not carry sensing equipment.

4.1.1 Synchronization and communication

The first shortcoming in the design of the AWR is that it is inherently a monostatic radar in the physical layer. TI did not design the AWR to exchange object state

information with neighboring AWR's via the mmWave band out-of-box, much less exchange synchronization-related messages with neighboring devices. Therefore, the design of MiNiMAP required a synchronization solution because even if the AWR's could share object state information, a lack of synchronization would prevent them from doing so.

To synchronize multiple AWR's, we had the option of pursuing software-based or hardware-based synchronization. Given that the AWR's cannot exchange timestamp messages to wirelessly synchronize, we implement the solution in section 3 to designate one device as the master and the rest as slaves. The specific processing is discussed further in the information processing section, but in order to achieve software-based synchronization we need a data processing and radar control mechanism. Each AWR is equipped with an ARM Cortex-R4F-Based radio control system and C674x digital signal processor (DSP), so the devices should be capable of conducting software-based synchronization independently of any external processors or microcontrollers. However, the goal of the first implementation of MiNiMAP was to provide a minimum viable product (MVP) from which further improvements could be made. Coding and troubleshooting a software-based synchronization on an embedded microcontroller was more time-intensive than adding an external processor to the system for both control and data processing. Specifically, MiNiMAP has the capability to integrate Windows, Mac, and Linux operating systems (OS) as the external processor and controller. These external processors interface with the AWR via the universal asynchronous receiver/transmitter (UART) protocol over the universal serial bus (USB) interface.

MiNiMAP's second synchronization option was a hardware-based triggering mechanism. Rather than rely on each device in the multistatic radar network to synchronize with the master, the hardware-based synchronization would provide a more consistent and instantaneous means of synchronization. Due to the additional overhead of a software-based synchronization approach, we decided that a hardware-based means of synchronization would expedite the development of the MVP. To achieve hardware-based synchronization, we needed to slightly modify the AWR and also design an

external hardware triggering circuit.

The AWR has two 20-pin BoosterPack connectors that enable the user to interface directly with the AWR's microcontroller [146]. Pin 9 on the J6 connector pin allows the user to interface with ball P4 on the AWR's microcontroller that is responsible for synchronization inputs. Examining the AWR schematic on page 10, we found that a 0 ohm resistor needed to be soldered on the AWR to allow a synchronization signal to pass from pin 9 to ball P4 [147].

With a synchronization pin available on all devices, MiNiMAP needed an external hardware triggering circuit. All input/output signals on the AWR require 3.3V and 50mA of current to operate, so the triggering circuit needs to be capable of supplying $|\mathcal{A}_{\mathrm{r}}| * 50\mathrm{mA}$ of current. To achieve this, we designed a basic hardware triggering circuit. Equal lengths of 22AWG wire were cut from the emitter of the circuit to each AWR device in order to reduce the synchronization offset between devices. The input to the base is controlled by the general purpose input/output (GPIO) pin on a Raspberry Pi 3 Model B (RPi). With each rising edge of a 3.3V synchronization signal, each AWR transmits a frame from the synthesizer. The process of triggering these synchronization pulses from the triggering circuit is discussed further in the MiNiMAP information processing section.

With the hardware triggering circuit capable of synchronizing all of the AWR's in a network, the next challenge to solve from an architecture perspective is the means of communication to avoid interference and whether the devices process data online or offline. Once again, the AWR devices are not capable of exchanging messages with each other, so the means of communication to avoid interference needed to be implicit rather than explicit. Before any object data captures begin, the devices must all share common configuration parameters for calculating the multistatic detections such as the starting frequency, chirp bandwidth, chirp slope, and frame period. To avoid interference, the devices can implement either a TDMA or PhDMA approach and each device would be pre-configured with a specific time slot or phase to avoid interference with neighboring devices. MiNiMAP achieved this by sharing a configuration file between all external processors that would be uploaded to the AWR by

each respective processor before testing. The configuration file contained common transmission parameters and unique TDMA time slots describing when each device transmitted and received in order to prevent interference.

Lastly, MiNiMAP processes data offline to accelerate MVP development. The development of an online data processing solution first requires robust data processing algorithms, so troubleshooting the development of these algorithms is easier in an offline rather than an online fashion. Furthermore, all of the testing data can be collected an stored in large quantities to verify the robustness of the data processing algorithms on multiple data sets instead of just the current test. Once the data processing algorithms of MiNiMAP are validated with experimental data sets, it is much easier to move to an online processing approach.

In summary, MiNiMAP's synchronization architecture implements a hardware-triggering approach of multiple AWR radars in the network and collects data offline. While this type of system is not immediately practical for real-life applications, this architecture enabled MiNiMAP to develop a MVP information processing system that achieves reliable and precise object estimates. The MVP performance is elaborated more in section 5, but now that MiNiMAP has developed a robust information processing system, further improvements can be made to its synchronization architecture to make it more viable for real-life applications. Specifically, future implementations of MiNiMAP should incorporate software-based synchronization to enable mobile applications and online data processing to provide near real-time object state estimates to the user.

4.1.2 Data gathering

Another shortcoming of the AWR's design that MiNiMAP needed to overcome was the fact that TI intended for the AWR to be exclusively an active radar. This is a problem because the design of a mmWave multistatic radar network requires devices to switch between passive receivers and active transceivers. Furthermore, this complicates the processing of multistatic data because the AWR's DSP chain was built exclusively for processing monostatic data. Therefore, MiNiMAP needed to rewrite part of the

AWR's firmware and also develop a means of collecting data before the AWR's DSP chain processed it.

Out-of-box, the AWR comes flashed with a firmware that does not allow the user to configure the AWR to produce the synthesizer signal without transmitting it. Most of the AWR's firmware is available for modification by TI's customers, so MiNiMAP rewrote part of the firmware to enable passive receivers and active transceivers. The AWR as a whole consists of three subsystems: the radar subsystem (RSS) that is responsible for producing the synthesizer signal based on the user's configuration, the DSP subsystem (DSS) that is responsible for processing the received signal, and the master subsystem (MSS) that is responsible for message passing between the user, RSS, and DSS. The RSS is not accessible for reprogramming, so MiNiMAP focused on altering the MSS and DSS code which is accessible from TI's software development kit (SDK) user guide [148].

According to the AWR user guide published by TI [146], the RSS could be configured to disable transmission of the synthesizer signal while still providing the signal as input to the mixer. The issue in TI's original firmware was that the DSS was configured to only process monostatic signals, so the MSS returned an error to a user attempting to configure the RSS for passive receiving. MiNiMAP rewrote part of the MSS code to avoid an error state for a passive receiving configuration, send the passive receiving configuration to the RSS, and inform the DSS that it should process the received signal as if the device was a transmitter in order to avoid any divide-by-zero errors from interrupting the AWR. Before multistatic experimentation, each AWR in the network must have MiNiMAP's firmware. The process for loading firmware on to the AWR can be found in the SDK user guide [148].

With the user able to configure each AWR device to be either a passive receiver or active transmitter, the next issue that the MiNiMAP's data gathering architecture must solve is how to process the received signal. The previous paragraph mentioned that the MSS informs the DSS to process all signals, whether they are monostatic or multistatic, as a monostatic signal. While TI's DSS would output the correct object detections for the monostatic case, every multistatic detection output would

be wrong and unusable. It is possible to rewrite the AWR's DSS to correctly process multistatic signals, but it is much more difficult to troubleshoot and develop the mutlistatic detection procedure described in section 3 than it is to troubleshoot a more developer-friendly program such as Python or MATLAB. Therefore, MiNiMAP incorporates TI's DCA1000 (DCA) [149] to route binary post-ADC signals from the AWR to the user.

The DCA's throughput on RGMII is 325Mbps, so any external processor that is receiving packets must be able to support this throughput or risk dropping packets, thus losing ADC data. In order to develop MiNiMAP's MVP, we used laptop computers with Windows, Linux, and Mac operating systems to act as the external processor to configure the AWR and to receive ADC data from the DCA. These systems are capable of achieving no packet loss because they support throughput greater than 325Mbps and have sufficient memory for data storage. However, the downside of using laptop computers is they do not have a small form factor.

In summary, MiNiMAP has its own firmware to enable each AWR to switch between passive receiver and active transceiver roles and MiNiMAP also uses the DCA to save ADC data for offline processing on laptop computers. While the large form factor of the laptop computers obviates the need for the AWR's small form factor, future implementations could incorporate small form factor single-board computers with sufficient ethernet throughput and memory, such as the Odroid XU4 or Rock64. Future implementations of MiNiMAP may not even require the use of the DCA because both monostatic and multistatic data processing would be done with the DSS. Small form factor radar devices enable applications, such as autonomous vehicles, that wish to reduce the weight and size of radar sensors.

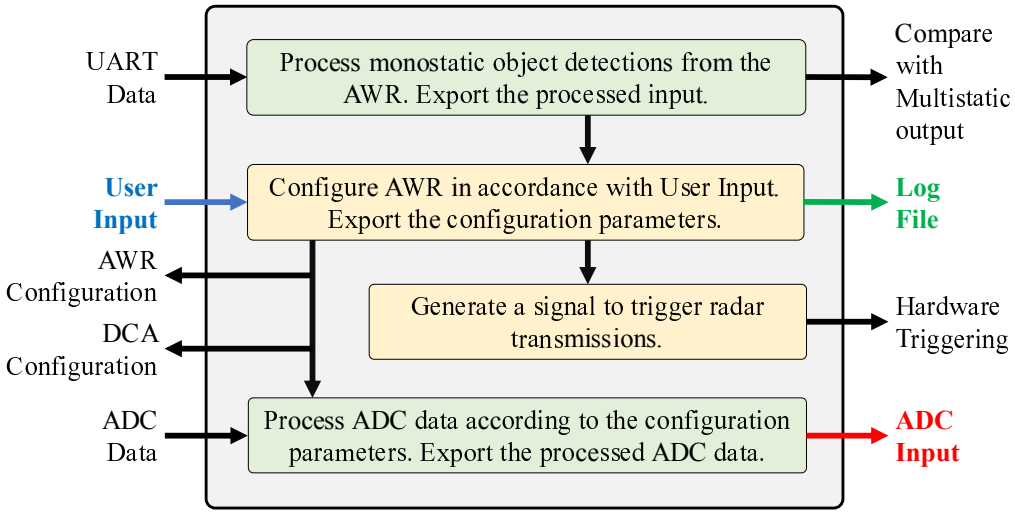
4.1.3 Data analysis

The final shortcoming of the AWR that MiNiMAP needed to improve was the simple tracking algorithms and lack of algorithms for multistatic detection. TI includes an algorithm called Group Tracking (GTRACK) in their SDK which uses an extended Kalman Filter (EKF) to track groups of objects [148]. However, the algorithm as-

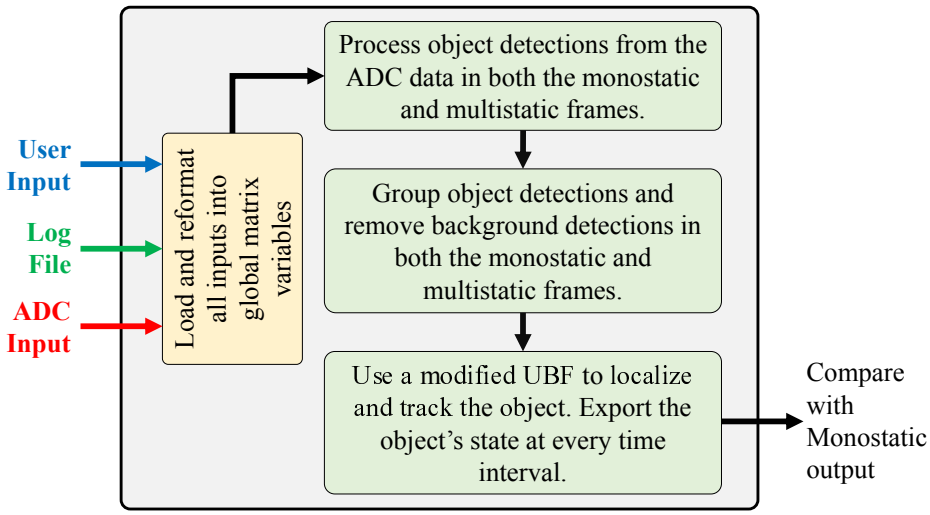
sumes that all radar detections are true object detections and that the object always exists in the radar's field of view. To overcome these design shortcomings, MiNiMAP develops more complex tracking algorithms and multistatic detection algorithms on an external processor to improve the probability of detecting an object and the accuracy of the object state estimate. The actual tracking algorithms and multistatic detection algorithms are explained further in the information processing section.

In order to develop more complex tracking algorithms, MiNiMAP uses an external processor for both ease in troubleshooting as well as for increased computing power. By itself, the AWR's DSS and MSS would not have enough computing power to support producing object state estimates in near real time from a tracking algorithm more complex than GTRACK. Therefore, MiNiMAP leverages the MATLAB programming environment to implement complex tracking algorithms and produce object state estimates from ADC data. In the current implementation, the binary ADC data from the DCA is preprocessed by Python scripts to create comma separated value (CSV) ADC files that serve as an input to MATLAB scripts. Due to the overhead of the external processor doing the work of the preprocessor, DSS, and object tracking, the object state estimates are not produced in real time. Therefore, the current implementation of MiNiMAP's object tracking system is in offline processing.

Similar to the implementation of the object tracking algorithms, MiNiMAP also implements the multistatic detection algorithms on the external processor. The AWR's DSS and MSS are capable of performing these computations, but to accelerate the development of MiNiMAP's MVP, the initial implementation was written in MATLAB. The program flow diagram is shown in Fig. 4.1.3 which gives an idea of the tasks the external processor is responsible for. In future implementations of MiNiMAP, a majority of these tasks can be removed from the external processor and assigned to the AWR. Sharing these tasks will help the computation of the object state estimate occur much closer to real time.



(a) Configuration block diagram



(b) Processing block diagram

Figure 4-2: The code block diagram for both the configuration and processing components of the external processor.

4.2 MiNiMAP Information Processing

Where the previous subsection on MiNiMAP architecture explained the hardware and software setup that enables tracking of objects in a multistatic radar network, this subsection applies the algorithms described in section 3 to track a person moving through a room. It is important to note that MiNiMAP is a case study of the algorithms explained in section 3; thus, MiNiMAP verifies synchronization, multistatic detection, and object tracking algorithms in a multistatic radar network.

4.2.1 Multistatic detection

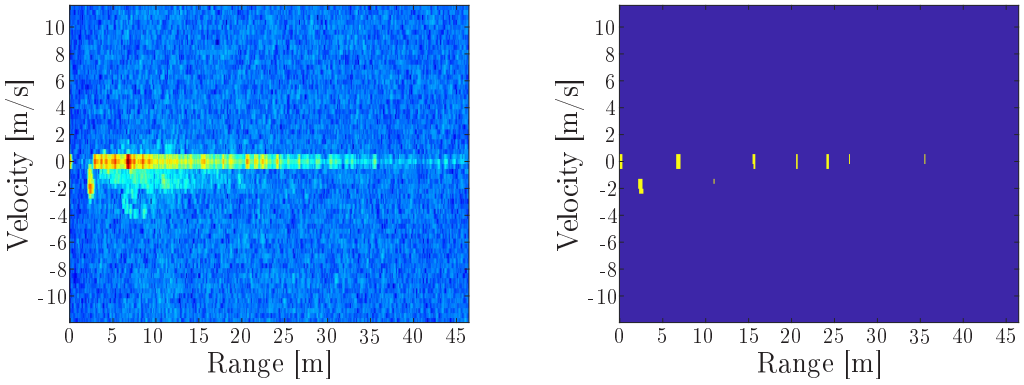
This section discusses the configuration of the AWR devices, and everything from the initiation of testing to the tracking algorithm. First, we explain the application's impact on the radar configuration parameters. Next, we explain how ADC data is collected and processed into object detections that serve as the input to object tracking algorithms.

The radar configuration file is written by the user in order to set application-specific parameters on each AWR before testing. For multistatic testing, it is important for MiNiMAP to assign the same chirp and frame parameters so that each device may process the backscattered signals from other devices. Each AWR in the network uses the full 4GHz bandwidth available in order to achieve the maximum range resolution, according to (3.2a). As long as the product of the chirp slope and chirp period equal the 4GHz bandwidth, MiNiMAP can achieve the maximum range resolution but the velocity resolution is affected by the chirp period according to (3.2b). MiNiMAP's chirp period is set at 160μs in order to set a maximum measurable doppler velocity at $\pm$5.78m/s. This speed is slightly faster than jogging for most humans and does not cover sprinting speed. The angle resolution and maximum observable angle is set by the design of the AWR. For timing purposes, the period of the frame is set to 200ms to allow 5 object measurements per second and to reduce the driving covariance in the object tracking model. The highest sampling rate was chosen to avoid aliasing.

To eliminate the possibility of multiple devices interfering with each other, MiNiMAP employs a time division multiplexing approach to device transmission. Each AWR is configured with 3 subframes: one subframe for receiving backscattered signals from all devices scheduled to transmit before device i, one subframe for device i's transmission (i.e. device $i = k$ in 3.1b), and one subframe for receiving backscattered signals from all devices scheduled to transmit after device i. Each device $i \in \mathcal{A}_{\mathrm{r}}$ receives a unique time index in which to transmit such that no devices should interfere with one another. Because MiNiMAP only uses 3 AWR in its network, the time between successive monostatic transmissions is 400ms, i.e. twice the frame period. As more devices are added to the network, a more intelligent scheduling algorithm should be employed to reduce the time between monostatic transmissions. The maximum number of devices that MiNiMAP can support in this scheme is 256 due to the AWR's design limitation of a maximum number of 255 subframe loops.

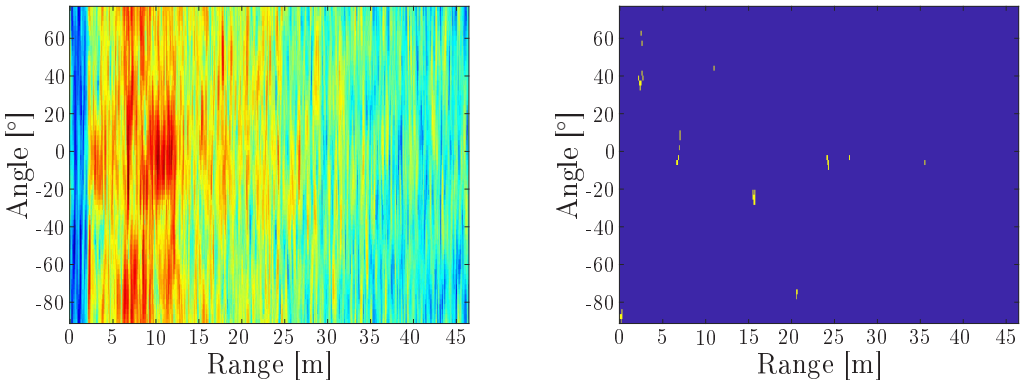
Once each AWR in the network is configured by its own external processor, radar synchronization takes over to trigger each AWR simultaneously for transmission or receiving. Then, the DCA begins to transfer ADC data from the AWR to the external processor. The DCA is shown in Fig. 4-1 and processes low-voltage differential signal (LVDS) data from the AWR's ADC, converts the data to binary, and transmits packaged binary data the user via the reduced gigabit media-independent interface (RGMII). The data format, command format, and configuration information for this interface is provided in the DCA user guide [149]. Out-of-box, TI restricts their customers to the use of proprietary software to collect and analyze ADC data from the AWR. The proprietary software was too restrictive and did not allow the user to process signals with their own software, so MiNiMAP needed to develop its own means of configuring and collecting raw ADC data from the DCA to enable data gathering during a multistatic network test. By using Wireshark to monitor the packet exchange between the DCA and a local computer running TI's propriety software, we were able to determine the list of commands that the computer needed to send to the DCA in order to initiate data transfer between the DCA and the local computer.

After the external processor receives the ADC data via UDP packets, a Python



(a) Range-Velocity FFT map

(b) Range-Velocity detection map



(c) Range-Angle FFT map

(d) Range-Angle detection map

Figure 4-3: A demonstration of the conversion between raw signal and detections in the range, velocity, and angle dimensions.

script both stores the binary ADC data to a local file and converts the ADC data to a comma separated value (CSV) file of complex I/Q samples. To perform this conversion, the Python file was written according to TI's ADC capture guide [150]. Once testing is complete, each external processor has a local CSV file containing all of the AWR's ADC data as well as the updated configuration file used for the test. The updated configuration file used for the test contains the radar configuration parameters as well as ADC data capture statistics such as dropped packets. The CSV file from each external processor can then be analyzed by MATLAB offline either individually or together with the other CSV files, depending on the user's desired implementation.

MiNiMAP's MATLAB scripts take the CSV files containing complex I/Q samples from each AWR device, in addition to the radar configuration file, as an input. The digital signal processing (DSP) chain then conducts either monostatic or multistatic processing depending on the AWR's configuration file and outputs a set of object detections for each frame. In the monostatic frames, the DSP chain outputs three dimensional detections in range, doppler velocity, and angle-of-arrival just like TI's DSS would. In the multistatic frames, the DSP chain outputs two dimensional detections in bistatic range and angle-of-arrival as described in section 3. Using the constant false-alarm rate cell averaging (CFAR) detection algorithm from [132], the CFAR threshold for both the monostatic case and the bistatic case was altered based on trial experiments. Ultimately, the thresholds were set at 13.5dB for monostatic and 3dB for multistatic. Bistatic doppler velocity is not possible for the AWR and this is explained in the next subsection on synchronization. Finally, the DSP chain outputs both monostatic and multistatic detection set to the tracking algorithm.

4.2.2 Radar Synchronization

This section discusses everything after configuration of the AWR devices to the initiation of testing. First, we explain the shortcomings of the AWR's design for software synchronization and then explain how the hardware triggering signal is controlled for hardware synchronization in MiNiMAP.

The two limiting factors of the AWR's design that makes software synchronization more difficult than hardware synchronization is its low range resolution and timing accuracy. While the AWR only has 5MHz ADC bandwidth, it would still be possible to configure a subframe with a longer period and lower bandwidth than the other subframes. This would allow any device to detect the transmitter, no matter how asynchronous the two devices begin. By detecting which chirp within the synchronization subframe contains the LOS peak, the receiving device could delay its subsequent transmissions so that it could synchronize with the master. With this scheme, the software synchronization would work just as well as hardware synchronization, but the downside is a lower data throughput since most of the transmission time would be occupied with maintaining synchronization. In future implementations of MiNiMAP, we will implement the software synchronization approach to test mobile tracking algorithms.

The true limitation of the AWR device comes with the inability for bistatic velocity detection. This is due to a combination of its low range resolution and timing accuracy. According to section 3, the device needs a range resolution much less than 0.5 the wavelength of the device. In the case of the AWR, the maximum range resolution is 4cm which is much greater than the 3mm operating wavelength. Even if the range resolution was precise enough to determine the time delay necessary for bistatic velocity synchronization, the timing accuracy of the AWR is not precise enough for this synchronization.

To achieve synchronization, MiNiMAP instead relies on the hardware triggering circuit described in the MiNiMAP hardware architecture section. The final component that was not explained in that section is the triggering mechanism at the gate of the BJT in the hardware triggering circuit. The gate voltage is controlled by the GPIO pin of a RPi, which is activated by a Python script. After all of the AWR devices in the network are configured and ready for testing, the host computer uploads the radar configuration file used for the test to the Raspberry Pi. The Python script on the Pi then parses the configuration file to find the frame timing. The AWR transmits a frame on the rising edge of the 3.3V pulse, so the Python script triggers

Figure 4-4: Physical test setup in an indoor localization environment

the BJT once at the beginning of each frame period for as many frames during the test as the configuration file specifies.

4.2.3 Object tracking modifications

This section covers everything from the output of multistatic detection to object state estimation. First, we explain the preprocessing to the object detections that were necessary before implementation in the object tracking filter. Next, we explain the object tracking filter that was used and the modifications that were made for the AWR multistatic radar network.

MiNiMAP employs the unscented single Bernoulli Filter (UBF) to track a person moving through the room [96]. The assumptions of the UBF is that all objects are point objects and that there can be a maximum of one object in the detection environment at all times. Due to the nature of the testing environment and the configuration of the AWR, the person produces multiple detections instead of the point detection that the UBF expects. Furthermore, the AWR also detects static

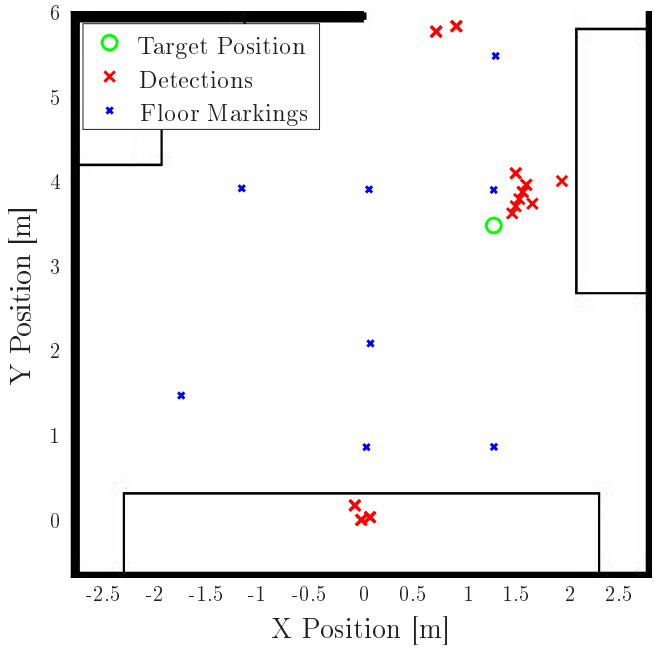
objects in the environment such as the walls, but the UBF is only expecting the single moving object. Therefore, MiNiMAP preprocesses the object detections with peak grouping and background removal algorithms to provide the UBF with its expected input. In future implementations of MiNiMAP, we will implement an extended object tracking algorithm to take advantage of the cluster of points that an object produces.

Peak grouping

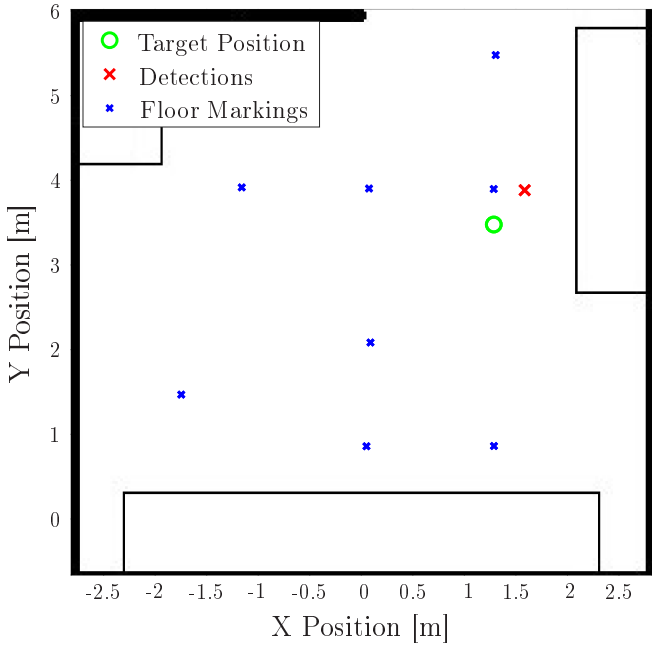
Peak grouping and background removal are necessary steps to distill the multiple detections that each device produces within any given frame as well as to avoid detecting undesirable objects so that the object in the environment can be reliably tracked. Due to the small resolution of range, velocity, and angle-of-arrival of mmWave devices given in (3.2a), (3.2c), objects produce more than one detection. However, object detections tend to cluster around each other. Peak grouping averages multidimensional set of neighboring detections and outputs a new, continuous-valued multidimensional detection. Peak grouping does this for every set of neighbors in every frame on every device. This aids in the data association uncertainty problem that the UBF will solve later in this section.

Background removal

Background removal takes the continuous-valued peak detections as an input and filters out the non-desired objects in the object environment. During a period of time in which the user is not surrounded by objects that they wish to track, background removal collects the set of detections and creates a multi-dimensional Gaussian PDF where the mean μ_d of each component of the PDF corresponds to the continuous-valued detection input from peak grouping. Background removal then uses this PDF to filter the non-desired objects during every time period in which the user is surrounded by objects. For every detection during the observation period of interest, if the detection is greater than a fixed threshold, the detection is considered to be a *background* detection. This detection is filtered out, but all remaining detections are provided as input to the UBF. Background removal is unable to filter clutter



(a) Original AWR1642 detections



(b) Detections after peak grouping and background removal

Figure 4-5: Figures (a) and (b) depict the indoor localization environment in Fig. 4-4. Figure (a) shows the detections produced by the AWR1642BOOST after the mmWave network processing step and figure (b) shows those same detections after peak grouping and background removal

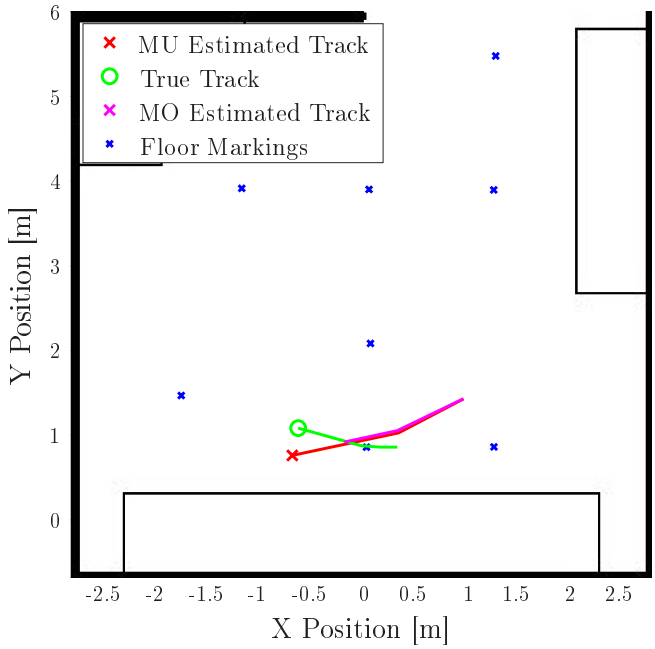
detections, but this problem is addressed by the UBF.

UBF

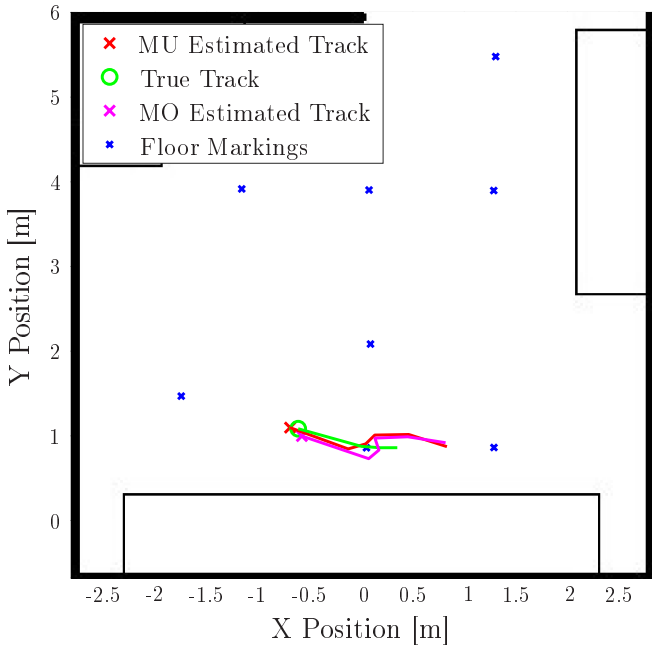
The primary components of the UBF is the spatial PDF and the existence random variable. In the context of the UBF, *existence* is defined as the ability of the mmWave network to detect the object. If the object lies outside of the detectable range of the device, the object does not exist. When the object is within the detectable range of the device, the object exists. *Survival* is defined as the object existing both in the previous time step as well as the current time step.

To solve the data association problem, the UBF accounts for the fact that a true object is more likely behave in a predictable fashion while the clutter detections from mirroring and non-informative multipath will occur in a more random, unpredictable fashion. Therefore, the UBF leverages temporal cooperation by using the information from the history of observations to inform its current prediction of the object's location. If the existence probability is below a certain threshold, no object position is output because the algorithm deems that no object exists. The more observations that the UBF has of the object, the better its estimate of the object position.

The modifications that MiNiMAP made to the UBF is addressed in section 3. To address the nonlinearities in the conversion between the monostatic and multistatic radar measurements, MiNiMAP employs sigma point belief propagation [139] to transform the desired object state domain to the observation domain.



(a) Comparison of tracks produced from one device in a multistatic (MU) and monostatic (MO) mode



(b) Comparison of tracks produced from three devices in multistatic (MU) and monostatic (MO) modes

Figure 4-6: Figures (a) and (b) depict the indoor localization environment in Fig. 4-4. Figure (a) shows the absence of a blue cross which signifies that the monostatic case failed to produce a detection and reliably track the target. Figure (b) shows the gain in tracking accuracy by adding multiple devices.

Chapter 5

Experimentation

Given the implementation of MiNiMAP in the previous section, this section covers the experimental setup and results for tracking a single object with a multistatic radar network. First, we discuss how each test for MiNiMAP was set up. Then, we discuss the improvements that MiNiMAP has for both one radar within a multistatic network and the fusion of all radars in the multistatic network.

5.1 Testing Setup

All together, there were three independent multistatic experiments conducted to validate the performance of MiNiMAP over monostatic radar. Each experiment consisted of three hardware trigger enabled AWR's connected to their own DCA and external processor as shown in Fig. 4-1. Due to the design of the AWR, the field of view is ± 90 according to the following equation:

$$\theta_{\max} = \sin^{-1}\left(\frac{\lambda}{2l}\right) \tag{5.1a}$$

where l is the separation distance of the receiving antennas. In the case of the AWR, this distance is $\frac{\lambda}{2}$ which maximizes the field of view. According to the AWR user guide [151], the radiation pattern has the highest gain in the E-plane and the

lowest gain in the H-plane. Therefore, the test setup keeps all AWR's at the same elevation with zero pitch to maximize receiver gain. Also, it is important to set up the AWR's such that they are within the field of view of all other AWR's in order to detect the LOS peak and produce multistatic detections.

The object of interest in each experiment was an adult human who walked around the room in a random pattern. Each test was conducted over a period of two minutes in which the person could have chosen to remain in the room or leave at their own will. The person had free choice over their movement speed as long as it did not exceed a jogging pace.

In order to assess the performance of MiNiMAP's ability to track the person moving through the room, it is necessary to measure the position and orientation of each radar as well as some reference positions within the room. In Fig. 4-5, you can see the reference positions within the room marked by white crosses. A laser range finder was used to measure the distance from each white cross to two of the surrounding walls in order to find the relative position of each white cross. To find the position of each radar, the same laser range finder was used to find the distance to to of the surrounding walls with the center of the receiver array acting as the reference point. The orientation was calculated by measuring the sides of a right triangle of the radar with respect to the wall that it was closest to.

Upon initiation of the test by the user, the triggering circuit activates a red LED that can be seen by the camera. This is used in order to obtain the relative time stamp that the person arrives and departs each of the white crosses. By assuming constant velocity between reference points, a rough true track of the object can be created by the user after testing is complete. In future implementations of MiNiMAP, it would be ideal to conduct the experiments in a motion capture laboratory to ensure that all error results from the imperfection of the tracking algorithm rather than from calibration instruments. Furthermore, a motion capture laboratory would enable the measurement of the person's velocity rather than just their position.

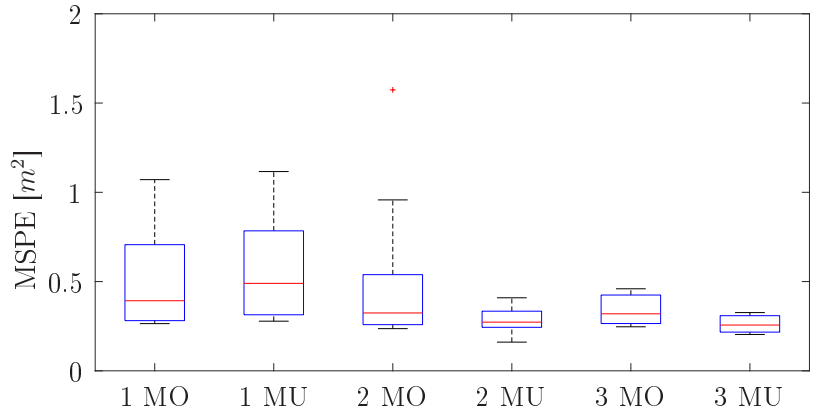
For the ease of configuring multiple AWR devices with their own radar configuration file, MiNiMAP set up a ad-hoc wireless network on the 2.4GHz band to allow

the host computer to SSH into all external processors on the network to configure the radars. While this wireless network is not in the mmWave band, it is exclusively used for the convenience of testing setup and not the facilitation of multistatic object tracking, synchronization, or data gathering. There is automatic Python scripts for configuring each radar device and saving the collected data to the local computer. Videos recording test results were also produced [152].

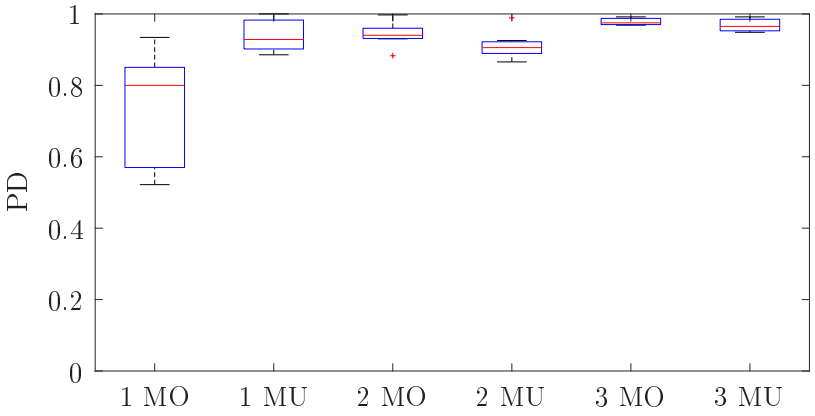
5.2 Single Radar Multistatic Results

The single radar case addresses MiNiMAP's performance when analyzing data collected by just one radar within a multistatic network. This case would be relevant in a non-collaborative scenario where the radar is not capable of sharing information in the data-link layer or network layer with other devices due to any constraint. While data is being collected by just one radar, all three radars are still transmitting according to a TDMA schedule. As the results in Fig. 5-1 show, the multistatic implementation experienced a higher probability of detection (PD) than the monostatic case, and a comparable probability of false-alarm (PFA) and localization mean squared error (MSE).

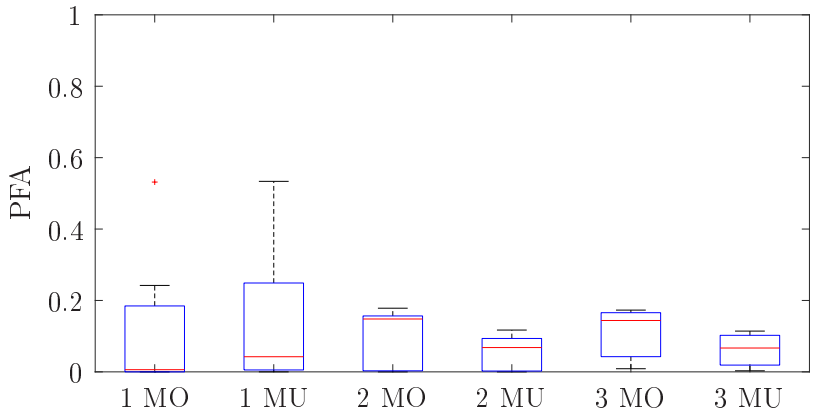
Assuming a fixed time schedule in order to avoid interference, the monostatic case only updates every $|\mathcal{A}_\mathrm{r}|$ time steps. Therefore, the UBF is able to converge its belief state to the true object state faster with multiple devices in the network than just one device because the update step happens more frequently. By passively receiving when other devices are transmitting, MiNiMAP improves the PD. While the localization MSE seems higher for the multistatic case, it is still very accurate as the cross-sectional area of a person is roughly $0.5\mathrm{m}^2$. Fig. 4-6 shows the multistatic case's improvement over the monostatic case. The PFA is observed to be mostly constant across all tests, indicative that the CFAR algorithm is doing its job.



(a) Comparing localization MSE for multistatic (MU) and monostatic (MO) radar networks



(b) Comparing PD for multistatic (MU) and monostatic (MO) radar networks



(c) Comparing PFA for multistatic (MU) and monostatic (MO) radar networks

Figure 5-1: The above figures show the key results from MiNiMAP's indoor localization experiments.

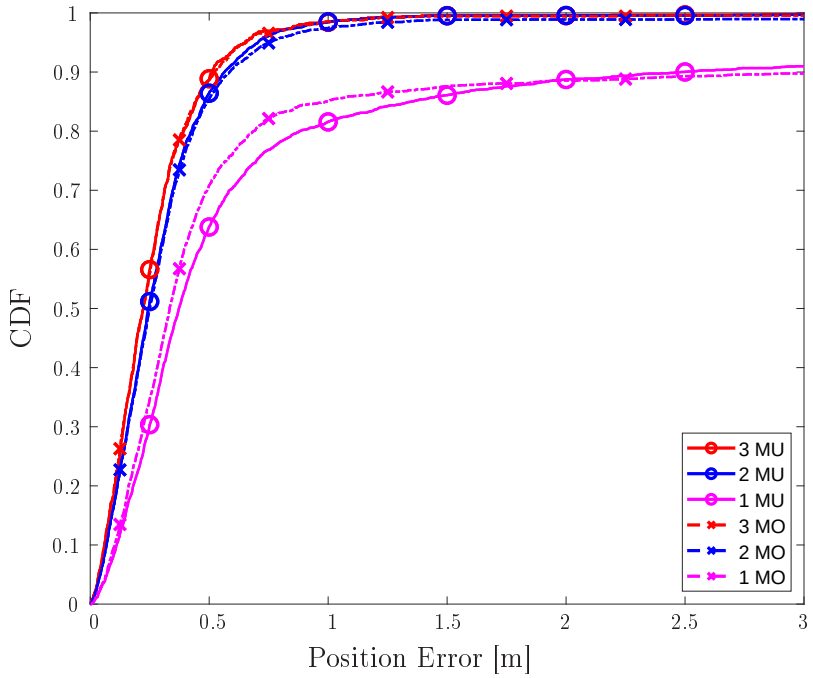


Figure 5-2: CDF of MiNiMAP's position estimate error for different multistatic (MU) and monostatic (MO) radar configurations

5.3 Multiple Radars Multistatic Results

The multiple radars case addresses MiNiMAP's performance when analyzing data collected by two or more radars within the multistatic network. This case is relevant when the radars are capable of sharing information with other radars in the network via either the data-link or network layer. As you can see in Fig. 5-1, the PD and PFA are comparable to the monostatic multiple device case, but the localization MSE is lower in the multistatic multiple device case.

Similar to the single device case, we assume there is a fixed time schedule in order to avoid interference. Therefore, the monostatic case only updates once every time step. Even though MiNiMAP is capable of updating $|\mathcal{A}_{\mathrm{r}}|$ times per time step, the PD is not greater than that of the monostatic multiple device case. A possible explanation for this is the relatively low detection probability for multistatic versus monostatic detections. The most significant improvement of the multistatic implementation seems to be that its PD is consistent across all number of devices in the

network. As the network scales up, it makes sense to only analyze the meaningful observations while in a limited access network like multistatic single device, it helps to analyze any measurements at the device's disposal in order to maintain PD.

Chapter 6

Conclusion

While the global positioning system (GPS) solves the problem of large-scale localization, it is more challenging to provide users with a precise and reliable location of objects that are within a close proximity. MiNiMAP is a novel localization and tracking system that localizes non-collaborative objects, synchronizes radars exclusively via the mmWave band, and solves an object detection association uncertainty problem in order to reliably track objects of interest.

By integrating these solutions into a coherent system, we demonstrate the functionality of MiNiMAP on mmWave radars for an indoor localization application. The indoor localization experiment showed that using a multistatic rather than monostatic radar network configuration affords gains in both PD and MPSE. Moving forward, there are many ways to improve the applicability of MiNiMAP to include software synchronization, more complex tracking algorithms, and increasing the number of devices in the network. Expanding the project in any of these directions will reveal new insights about applications of multistatic radar networks.

Bibliography

[1] M. Z. Win, A. Conti, S. Mazuelas, Y. Shen, W. M. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[2] M. Z. Win, Y. Shen, and W. Dai, "A theoretical foundation of network localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1136–1165, Jul. 2018, special issue on *Foundations and Trends in Localization Technologies*.

[3] M. Z. Win, W. Dai, Y. Shen, G. Chrisikos, and H. V. Poor, "Network operation strategies for efficient localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1224–1254, Jul. 2018, special issue on *Foundations and Trends in Localization Technologies*.

[4] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft information for localization-of-things," *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Nov. 2019.

[5] T. Wang, Y. Shen, A. Conti, and M. Z. Win, "Network navigation with scheduling: Error evolution," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7509–7534, Nov. 2017.

[6] M. Z. Win, F. Meyer, Z. Liu, W. Dai, S. Bartoletti, and A. Conti, "Efficient multi-sensor localization for the Internet-of-Things," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 153–167, Sep. 2018.

[7] N. Javaid, A. Sher, H. Nasir, and N. Guizani, "Intelligence in IoT-based 5G networks: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 94–100, Oct. 2018.

[8] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[9] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for vehicular communications," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 111–117, Jan. 2018.

[10] F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera, "Robust multirate on-road vehicle localization for autonomous highway driving vehicles," *IEEE Trans. Serv. Comput.*, vol. PP, no. 99, 2016.

[11] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[12] T. Rappaport, J. Reed, and B. Woerner, "Position location using wireless communications on highways of the future," *IEEE Commun. Mag.*, vol. 34, no. 10, pp. 33–41, 1996.

[13] S. B. Cruz, T. E. Abrudan, Z. Xiao, N. Trigoni, and J. Barros, "Neighbor-aided localization in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. PP, no. 99, pp. 1–10, 2017.

[14] J. Yao, A. T. Balaei, M. Hassan, N. Alam, and A. G. Dempster, "Improving cooperative positioning for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2810–2823, Jul. 2011.

[15] L.-J. Kau and C.-S. Chen, "A smart phone-based pocket fall accident detection, positioning, and rescue system," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 44–56, Jan. 2015.

[16] G. Cardone, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola, "Fostering participaction in smart cities: A geo-social crowdsensing platform," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 112–119, Jun. 2013.

[17] X. Shen, J. W. Mark, and J. Ye, "Mobile location estimation in CDMA cellular networks by using fuzzy logic," *Wireless Personal Commun.*, vol. 22, no. 1, pp. 57–70, 2002.

[18] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, and M. Z. Win, "High-accuracy localization for assisted living," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 59–70, Mar. 2016.

[19] T. Latif, E. Whitmire, T. Novak, and A. Bozkurt, "Sound localization sensors for search and rescue biobots," *IEEE Sensors J.*, vol. 16, no. 10, pp. 3444–3453, May 2016.

[20] J. Guivant, E. Nebot, and S. Baiker, "Autonomous navigation and map building using laser range sensors in outdoor applications," *J. of Robotic Systems*, vol. 17, no. 10, pp. 565–583, Oct. 2000.

[21] T. Budinger, "Biomonitoring with wireless communications," *Annual Review of Biomedical Engineering*, vol. 5, no. 1, pp. 383–412, 2003.

[22] S. Bartoletti, A. Conti, and M. Z. Win, "Device-free counting via wideband signals," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1163–1174, May 2017.

[23] B. W. Parkinson, J. J. Spilker Jr., P. Axelrad, and P. Enge, *Global Positioning System: Theory and Applications, Volume I*, ser. Progress in Astronautics and Aeronautics. Washington D.C., USA: AIAA, 1996.

[24] B. W. Parkinson and S. W. Gilbert, "NAVSTAR: Global positioning system - Ten years later," *Proc. IEEE*, vol. 71, no. 10, pp. 1177–1186, Oct 1983.

[25] E. D. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*, 3rd ed. Norwood, MA, USA: Artech House, 2017.

[26] B. S. Pervan, D. G. Lawrence, and B. W. Parkinson, "Autonomous fault detection and removal using gps carrier phase," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 3, pp. 897–906, Jul 1998.

[27] C. Hegarty and E. Chatre, "Evolution of the global navigation satellite system (GNSS)," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 1902–1917, January 2008.

[28] R. W. Klukas, G. Lachapelle, C. Ma, and G.-I. Jee, "GPS signal fading model for urban centers," *IEE Proc. Microw., Antennas Propagat.*, vol. 150, no. 4, pp. 245–252, Aug. 2003.

[29] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.

[30] D. Schneider, "New indoor navigation technologies work where GPS can't," *IEEE Spectr.*, Nov. 2013.

[31] C.-C. Chong, D. I. Laurenson, and S. McLaughlin, "A wideband dynamic spatio-temporal Markov channel model for typical indoorpropagation environments," in *Proc. IEEE Semiannual Veh. Technol. Conf.*, vol. 1, Jeju, SOUTH KOREA, Apr. 2003, pp. 6–10.

[32] R. Casas, A. Marco, J. J. Guerrero, and J. Falcó, "Robust estimator for non-line-of-sight error mitigation in indoor localization," *EURASIP J. Appl. Signal Process.*, no. 1, pp. 156–156, 2006.

[33] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Lightweight map matching for indoor localisation using conditional random fields," in *Proc. IEEE Inform. Processing in Sensor Networks*, Berlin, Germany, Apr. 2014, pp. 131–142.

[34] Z. Yang, X. Feng, and Q. Zhang, "Adometer: Push the limit of pedestrian indoor localization through cooperation," *IEEE Trans. Mobile Comput.*, vol. 13, no. 11, pp. 2473–2483, Nov. 2014.

[35] Z. Liu, W. Dai, and M. Z. Win, "Mercury: An infrastructure-free system for network localization and navigation," *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 1119–1133, May 2018.

[36] S. Bartoletti, A. Giorgetti, M. Z. Win, and A. Conti, "Blind selection of representative observations for sensor radar networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1388–1400, Apr. 2015.

[37] S. Bartoletti, A. Conti, A. Giorgetti, and M. Z. Win, "Sensor radar networks for indoor tracking," *IEEE Wireless Commun. Lett.*, vol. 3, no. 2, pp. 157–160, Apr. 2014.

[38] U. A. Khan, S. Kar, and J. M. F. Moura, "DILAND: An algorithm for distributed sensor localization with noisy distance measurements," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1940–1947, Mar. 2010.

[39] Y. Shen and M. Z. Win, "On the accuracy of localization systems using wideband antenna arrays," *IEEE Trans. Commun.*, vol. 58, no. 1, pp. 270–280, Jan. 2010.

[40] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: Challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 24–40, Jul. 2005.

[41] C. Falsi, D. Dardari, L. Mucchi, and M. Z. Win, "Time of arrival estimation for UWB localizers in realistic environments," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 1–13, Article ID 32 082, 2006, special issue on *Wireless Location Technologies and Applications*.

[42] A. Rabbachin, I. Oppermann, and B. Denis, "GML ToA estimation based on low complexity UWB energy detection," in *Proc. PIMRC '06*, Helsinki, Finland, Sep. 2006.

[43] D. Dardari and M. Z. Win, "Threshold-based time-of-arrival estimators in UWB dense multipath channels," in *Proc. IEEE Int. Conf. Commun.*, vol. 10, Istanbul, Turkey, Jun. 2006, pp. 4723–4728.

[44] H. S. Ahn and W. Yu, "Environmental-adaptive RSSI-based indoor localization," *IEEE Transactions on Automation Science and Engineering*, 2009, in press.

[45] N. Patwari and A. O. Hero, III, "Using proximity and quantized RSS for sensor localization in wireless networks," *IEEE/ACM 2nd Workshop on Wireless Sensor Nets. & Applications*, pp. 20–29, 2003.

[46] J. P. Beaudeau, M. F. Bugallo, and P. M. Djurić, "RSSI-based multi-target tracking by cooperative agents using fusion of cross-target information," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5033–5044, Oct. 2015.

[47] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 4, pp. 806–811, Aug. 1985.

[48] B. Van Veen and K. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE Signal Process. Mag.*, vol. 5, no. 2, pp. 4–24, 1988.

[49] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, 1996.

[50] R. Engelbrecht, "Passive source localization from spatially correlated angle-of-arrival data," *IEEE Trans. Signal Process.*, vol. 31, no. 4, pp. 842–846, 1983.

[51] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," *Proc. IEEE Int. Conf. on Computer Commun.*, vol. 3, pp. 1734–1743, Mar./Apr. 2003.

[52] L. Xiong, "A selective model to suppress NLOS signals in angle-of-arrival (AOA) location estimation," in *Proc. 9th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications*, vol. 1, Boston, MA, 1998, pp. 461–465.

[53] H.-J. Shao, X.-P. Zhang, and Z. Wang, "Efficient closed-form algorithms for AOA based self-localization of sensor nodes using auxiliary variables," *IEEE Trans. Signal Process.*, vol. 62, no. 10, pp. 2580–2594, May 2014.

[54] S. Bartoletti, W. Dai, A. Conti, and M. Z. Win, "A mathematical model for wideband ranging," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 2, pp. 216–228, Mar. 2015.

[55] J. Prieto, S. Mazuelas, and M. Z. Win, "Context-aided inertial navigation via belief condensation," *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 3250–3261, Jun. 2016.

[56] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems.* Artech House, 2008.

[57] A. R. Jiménez, F. Seco, J. C. Prieto, and J. Guevara, "Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 1, pp. 178 – 189, Jan. 2012.

[58] S. Beauregard, "A helmet-mounted pedestrian dead reckoning system," in *4th International Forum on Applied Wearable Computing*, 2006.

[59] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Comput. Graph. Appl.*, vol. 25, no. 6, pp. 38–46, Nov.-Dec. 2005.

[60] P. Closas, C. Fernández-Prades, and J. A. Fernández-Rubio, "Direct position estimation approach outperforms conventional two-steps positioning," in *Proc. of European Signal Processing Conference (EUSIPCO)*, Aug. 2009, pp. 1958–1962.

[61] S. Safavi, U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed localization: A linear theory," *Proc. IEEE*, vol. 106, no. 7, pp. 1204–1223, Jul. 2018.

[62] T. V. Nguyen, Y. Jeong, H. Shin, and M. Z. Win, "Machine learning for wideband localization," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1357–1380, Jul. 2015.

[63] J. Warrior, E. McHenry, and K. McGee, "They know where you are – location detection," *IEEE Spectr.*, vol. 40, no. 7, pp. 20–25, 2003.

[64] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, Jul. 2005.

[65] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 961–974, Nov. 2004.

[66] M. Li and Y. Liu, "Rendered path: Range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 320–332, Feb. 2010.

[67] A. M. Haimovich, R. S. Blum, and L. J. Cimini, "MIMO radar with widely separated antennas," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 116–129, Jan. 2008.

[68] H. Godrich, A. M. Haimovich, and R. S. Blum, "Target localization accuracy gain in MIMO radar-based systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2783–2803, Jun. 2010.

[69] T. Derham, S. Doughty, K. Woodbridge, and C. Baker, "Design and evaluation of a low-cost multistatic netted radar system," *IET Radar, Sonar and Navigation*, vol. 1, no. 5, pp. 362–368, Sep. 2007.

[70] E. G. Alivizatos, M. N. Petsios, and N. K. Uzunoglu, "Architecture of a multistatic FMCW direction-finding radar," *Aerospace Science and Technology*, vol. 12, no. 2, pp. 169–176, Mar. 2008.

[71] B. Teague, Z. Liu, F. Meyer, and M. Z. Win, "Peregrine: 3-D network localization and navigation," in *Proc. IEEE Latin-American Conf. Commun.*, Guatemala City, Guatemala, Nov. 2017, pp. 1–6.

[72] M. Anisetti, C. A. Ardagna, V. Bellandi, E. Damiani, and S. Reale, "Map-based location and tracking in multipath outdoor mobile networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 3, pp. 814–824, 2011.

[73] F. Montorsi, S. Mazuelas, G. M. Vitetta, and M. Z. Win, "On the performance limits of map-aware localization," *IEEE Trans. Inf. Theory*, vol. 59, no. 8, pp. 5023–5038, Aug. 2013.

[74] K. Zhang, Y. Shen, and M. Z. Win, "On the performance of map-aware cooperative localization," in *Proc. IEEE Int. Conf. Commun.*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[75] M. Vari and D. Cassioli, "mmWaves RSSI indoor network localization," in *Proc. IEEE Int. Conf. Commun.*, Sydney, NSW, Australia, Jun. 2014, pp. 1–6.

[76] J. Palacios, P. Casari, and J. Widmer, "JADE: Zero-knowledge device localization and environment mapping for millimeter wave systems," in *Proc. IEEE Int. Conf. on Computer Commun.*, Atlanta, GA, USA, May 2017, pp. 1–9.

[77] W. Chen and X. Meng, "A cooperative localization scheme for Zigbee-based wireless sensor networks," *Networks, 2006. ICON'06. 14th IEEE International Conference on*, vol. 2, 2006.

[78] N. Alsindi and K. Pahlavan, "Cooperative localization bounds for indoor ultra-wideband wireless sensor networks," *EURASIP J. Adv. in Signal Process.*, vol. 8, no. 2, pp. 1–13, Jan. 2008.

[79] X. Tan and J. Li, "Cooperative positioning in underwater sensor networks," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5860–5871, Nov. 2010.

[80] S. Mazuelas, A. Conti, J. C. Allen, and M. Z. Win, "Soft range information for network localization," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3155–3168, Jun. 2018.

[81] W. C. Lindsey, F. Ghazvinian, W. C. Hagmann, and K. Dessouky, "Network synchronization," *Proc. IEEE*, vol. 73, no. 10, pp. 1445–1467, 1985.

[82] S. Bregni, *Synchronization of Digital Telecommunications Network*. West Sussex, England: John Wiley & Sons, Ltd., 2002.

[83] S. Bregni, "Clock stability characterization and measurement in telecommunications," *IEEE Trans. Instrum. Meas.*, vol. 46, no. 6, pp. 1284 – 1294, Dec. 1997.

[84] S. Bregni, "A historical perspective on telecommunications network synchronization," *IEEE Commun. Mag.*, vol. 36, no. 6, pp. 158–166, Jun. 1998.

[85] J. Kim, J. Chun, and I. Choi, "Time and frequency synchronization of bistatic FMCW radar," in *Proceedings of the IET International Radar Conference*, Hangzhou, China, Oct. 2015, pp. 1–5.

[86] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1992.

[87] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[88] M. J. Neely, "Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 5, pp. 1188–1199, Oct. 2008.

[89] T. Wang, A. Conti, and M. Z. Win, "On the design of scheduling algorithms for wireless navigation networks," in *Proc. IEEE Global Telecomm. Conf.*, Washington, DC, Dec. 2016, pp. 1–6.

[90] Y. Liu, Y. Shen, D. Guo, and M. Z. Win, "Network localization and synchronization using full-duplex radios," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 714–728, Feb. 2018.

[91] A. Conti, D. Dardari, M. Guerra, L. Mucchi, and M. Z. Win, "Experimental characterization of diversity navigation," *IEEE Syst. J.*, vol. 8, no. 1, pp. 115–124, Mar. 2014.

[92] D. M. Bevly and B. Parkinson, "Cascaded kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 2, pp. 199–208, March 2007.

[93] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, 1960.

[94] H. G. Hoang and B. T. Vo, "Sensor management for multi-target tracking via multi-bernoulli filtering," *Automatica*, vol. 50, no. 4, pp. 1135 – 1142, 2014.

[95] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409–423, Feb. 2009.

[96] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina, "A tutorial on bernoulli filters: Theory, implementation and applications," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3406–3430, Jul. 2013.

[97] E. A. Wan and R. Van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. New York, NY, USA: Wiley, 2001, ch. 7, pp. 221–280.

[98] S. Haykin, "Kalman filters," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. New York, NY, USA: Wiley, 2001, ch. 1, pp. 1–21.

[99] R. E. Kalman, P. L. Falb, and M. A. Arbib, *Topics in Mathematical System Theory*. New York: McGraw-Hill, 1969.

[100] Y. Cheng and T. Singh, "Efficient particle filtering for road-constrained target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1454–1469, 2007.

[101] M. Z. Win, "Belief condensation filtering: framework and algorithms," Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, May 2012.

[102] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[103] S. Mazuelas, Y. Shen, and M. Z. Win, "Belief condensation filtering," *IEEE Trans. Signal Process.*, vol. 61, no. 18, pp. 4403–4415, Sep. 2013.

[104] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo: Morgan Kaufmann, 1988.

[105] Z. Liu, W. Dai, and M. Z. Win, "Node placement for localization networks," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, May 2017, pp. 1–6.

[106] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experimentation for cooperative localization," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 467–475, Feb. 2012.

[107] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.

[108] D. Dardari, A. Conti, J. Lien, and M. Z. Win, "The effect of cooperation on localization systems using UWB experimental data," *EURASIP J. Appl. Signal Process.*, vol. 2008, pp. 1–11, Article ID 513873, 2008, special issue on *Cooperative Localization in Wireless Ad Hoc and Sensor Networks*.

[109] Y. Shen, H. Wymeersch, and M. Z. Win, "Fundamental limits of wideband localization – Part II: Cooperative networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4981–5000, Oct. 2010.

[110] M. Z. Win, Y. Shen, and H. Wymeersch, "On the position error bound in cooperative networks: A geometric approach," in *Proc. IEEE Int. Symp. on Spread Spectrum Tech. & Applicat.*, Bologna, Italy, Aug. 2008, pp. 637–643.

[111] Y. Shen and M. Z. Win, "Fundamental limits of wideband localization – Part I: A general framework," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4956–4980, Oct. 2010.

[112] Y. Shen, H. Wymeersch, and M. Z. Win, "Fundamental limits of wideband cooperative localization via Fisher information," in *Proc. IEEE Wireless Commun. and Networking Conf.*, Kowloon, Hong Kong, Mar. 2007, pp. 3951–3955.

[113] Y. Shen and M. Z. Win, "Fundamental limits of wideband localization accuracy via Fisher information," in *Proc. IEEE Wireless Commun. and Networking Conf.*, Kowloon, Hong Kong, Mar. 2007, pp. 3046–3051.

[114] W. Dai and M. Z. Win, "A theoretical foundation for location secrecy," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, May 2017, pp. 1–6.

[115] H. L. V. Trees, *Detection, Estimation, and Modulation Theory*, 1st ed. New York, NY 10158-0012: John Wiley & Sons, Inc., 1968.

[116] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer-Verlag, 1994.

[117] A. D'Andrea, U. Mengali, and R. Reggiannini, "The modified Cramer-Rao bound and its application to synchronization problems," *IEEE Trans. Commun.*, vol. 42, no. 2,3,4, pp. 1391–1399, Feb.-Apr. 1994.

[118] T. A. Cover and J. A. Thomas, *Elements of Information Theory*, 1st ed. New York, NY, 10158: John Wiley & Sons, Inc., 1991.

[119] Y. Shen, S. Mazuelas, and M. Z. Win, "Network navigation: Theory and interpretation," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 9, pp. 1823–1834, Oct. 2012.

[120] M. Leng, W. P. Tay, C. M. S. See, S. G. Razul, and M. Z. Win, "Modified CRLB for cooperative geolocation of two devices using signals of opportunity," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3636–3649, Jul. 2014.

[121] T. Wang, Y. Shen, S. Mazuelas, H. Shin, and M. Z. Win, "On OFDM ranging accuracy in multipath channels," *IEEE Syst. J.*, vol. 8, no. 1, pp. 104–114, Mar. 2014.

[122] P. Tichavský, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[123] C. Chang and A. Sahai, "Cramér-Rao-type bounds for localization," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. Article ID 94 287, 13 pages, 2006.

[124] H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part 1.* New York, NY: Wiley, 1968.

[125] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation.* New York, NY, USA: Wiley, 2002.

[126] H. Griffiths and M. Inggs, "Multistatic radar: System requirements and experimental validation," *International Radar Conference*, Oct. 2014.

[127] E. Alivizatos, M. Petsois, and N. Uzunoglu, "Towards a range-doppler uhf multistatic radar for the detection of non-cooperative targets with low rcs," *Journal of Electromagn. Waves and Appl.*, vol. 19, no. 15, pp. 2015–2031, 2005.

[128] A. Stove, "Linear FMCW radr techniques," *IEEE Proceedings F - Radar and Signal Processing*, vol. 139, no. 5, pp. 343–350, Oct. 1992.

[129] C. Iovescu and S. Rao, "The fundamentals of millimeter wave sensors."

[130] H. Griffiths, "New ideas in fm radar," *Electronics and Communication Engineering Journal*, vol. 2, no. 5, pp. 185–194, Oct. 1990.

[131] G. Booker, "Understanding millimetre wave fmcw radars," *International Conference on Sensing Technology*, vol. 1, Nov. 2005.

[132] M. Richards, *Fundamentals of Radar Signal Processing.* New York: McGraw Hill, 2005.

[133] S. Bartoletti, A. Conti, W. Dai, and M. Z. Win, "Threshold profiling for wideband ranging," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 873–877, Jun. 2018.

[134] M. Ash, M. Ritchie, K. Chetty, and P. V. Brennan, "A new multistatic FMCW radar architecture by over-the-air deramping," *IEEE Sensors J.*, vol. 15, no. 12, pp. 7045–7053, Dec. 2015.

[135] M. Weib, "Synchronisation of bistatic radar systems," *International Geoscience and Remote Sensing Symposium*, pp. 1750–1753, Sep. 2004.

[136] I. Sari, E. Serpedin, K. NOh, Q. Chaudhari, and B. Suter, "On the joint synchronization of clock offset and skew in rbs-protocol," *IEEE Trans. Commun.*, vol. 56, no. 5, pp. 700–703, May 2008.

[137] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *The Fifth Symposium on Operating Systems Design and Implementation*, pp. 147–163, Dec. 2002.

[138] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," *Proc. SenSys*, vol. 3, pp. 138–149, Nov. 2003.

[139] F. Meyer, O. Hlinka, and F. Hlawatsch, "Sigma point belief propagation," *IEEE Signal Processing*, vol. 21, pp. 145–149, Feb. 2014.

[140] P. Willett and R. S. Y. Ruan, "Pmht: Problems and some solutions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 738–754, Jul. 2002.

[141] Y. Bar-Shalom and J. H. F. Daum, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[142] O. Drummond, "Tracking clusters and extended objects with multiple sensors," *Signal and Data Processing of Small Targets*, vol. 1305, Oct. 1990.

[143] K. Granstrom and S. R. M. Baum, "Extended object tracking: Introduction, overview, and applications," *Journal of Advances in Information Fusion*, vol. 12, no. 2, pp. 139–174, Dec. 2016.

[144] S. Coraluppi and D. Grimmett, "Multistatic sonar tracking," *Signal Processing, Sensor Fusion, and Target Recognition*, vol. 5096, Aug. 2005.

[145] Texas Instruments, "AWR1642 single-chip 76-GHz to 81-GHz automotive radar sensor." [Online]. Available: http://www.ti.com/product/AWR1642/technicaldocuments

[146] Texas Instruments, "AWR1642 evaluation module (awr1642boost) single-chip mmWave sensing solution," Apr. 2018. [Online]. Available: http://www.ti.com/lit/ug/swru508b/swru508b.pdf

[147] Texas Instruments, "Awr1642boost schematic, assembly, and bom (rev. a)." [Online]. Available: http://www.ti.com/tool/AWR1642BOOST#technicaldocuments

[148] Texas Instruments, "mmWave software development kit (sdk)." [Online]. Available: http://www.ti.com/tool/MMWAVE-SDK

[149] Texas Instruments, "DCA1000EVM data capture card." [Online]. Available: http://www.ti.com/tool/DCA1000EVM

[150] Texas Instruments, "mmwave radar device adc raw data capture." [Online]. Available: http://www.ti.com/lit/an/swra581b/swra581b.pdf

[151] Texas Instruments, "Awr1642 obstacle detection sensor (awr1642boost-ods) single-chip mmwave sensing solution." [Online]. Available: http://www.ti.com/lit/ug/spruik1/spruik1.pdf

[152] S. Miller, Z. Liu, Z. Yu, and G. Denove, "Minimap: Localization and tracking in a multistatic millimeter wave MIMO radar network," Nov. 2019. [Online]. Available: https://youtu.be/Ff8oj5HhLFM