Multiple Target Tracking in Experimental Multistatic MIMO mmWave Radar Sensor Networks

by

ENS George Thomas Denove
B.S., United States Naval Academy (2019)
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
June 2021

Author . . . .
Department of Aeronautics and Astronautics
May 18, 2021

Certified by . . . .
Professor Moe Z. Win
Professor, Aeronautics and Astronautics
Thesis Supervisor

Certified by . . . .
Dr. Sintayehu A. Dehnie
Laboratory Technical Staff, The Charles Stark Draper Laboratory, Inc.
Thesis Supervisor

Accepted by . . . .
Zoltan Spakovszky
Professor, Aeronautics and Astronautics
Chair, Graduate Program Committee

Multiple Target Tracking in Experimental Multistatic MIMO mmWave Radar Sensor Networks

by

ENS George Thomas Denove

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Location awareness of non-collaborative targets within 5G and beyond systems is becoming ever more prominent. This thesis presents a passive, hardware-agnostic multiple target tracking (MTT) system that addresses the shortcomings in current wireless position technology and is capable of seamlessly integrating with millimeter-wave (mmWave) communication infrastructure. The developed system is a radar sensor network (RSN) composed of distributed low-cost mmWave devices, which are designed to simultaneously transmit and receive for improved network throughput. We develop Doppler compensation and signal decoding algorithms integral to properly resolving target position information through multistatic sensing channels. Results indicate our system's advancements achieve performance improvements over existing systems in non-collaborative target detection and MTT in harsh wireless environments.

Thesis Supervisor: Professor Moe Z. Win
Title: Professor, Aeronautics and Astronautics

Thesis Supervisor: Dr. Sintayehu A. Dehnie
Title: Laboratory Technical Staff, The Charles Stark Draper Laboratory, Inc.

Acknowledgments

I would like to begin by thanking Draper for providing me with the incredible opportunity to attend MIT to pursue my masters degree and to conduct top notch research. Without the outstanding support of the Draper Fellows program, this thesis would not have been possible. Specifically, I would like to thank Dr. Chris Yu and Ms. Martha Porter for coordinating the entire Fellows program. Furthermore, I would like to extend my many thanks and gratitude to my Draper advisor, Dr. Sintayehu A. Dehnie. Sintayehu provided me with both technical expertise as well as constant encouragement throughout my entire stint in Cambridge. I would also like to thank all of the Draper professionals and technical experts who were so willing to meet with me to discuss everything from my thesis to Navy intelligence and everything in between, specifically Ramesh Annavajjala and Fred Kautz. Their wealth of knowledge is evident and I appreciate them sharing it with me.

In addition to Draper, I would like to thank Professor Moe Z. Win and his laboratory, the Wireless Information and Network Sciences Laboratory, for welcoming me into their group. Additionally, I would like to thank Professor Andrea Conti from the University of Ferrara in Italy for always providing comments and advice on my work, despite the large time difference between our physical locations. Many thanks and much gratitude for Dr. Augustin Alex Saucan. Our conversations, both in person and over Skype, truly transformed my thesis into what it is today and inspired me to keep seeking out solutions for some of the harder challenges I faced during this journey, even when no light was in sight. Also, thank you to my officemates, Zhenyu Liu and Spyros Tserkis, for always being there for me when I needed some support and advice.

To my friends, both near and far, I thank you for all of the support, encouragement, and late night phone calls. I especially thank two life long friends who I met here at MIT: Samuel Miller and Catalina Rico. Sam, I cannot thank you enough for your incredible friendship, tasty meals, and constant advice on my thesis, especially once you left MIT to start your Army training. And Cat, thank you for always being there

to lend an ear when needed, cooking Home Chef meals together, and sticking by one another through three different roommates.

Finally, to my family: I cannot thank you all enough. To my parents and brother Nick, thank you for always being just a phone call away. I always love seeing you guys; whether we are hanging out on the boat, going to the beach, or even just sharing a nice meal together, it is always a great time. I am looking forward to us experiencing my next journey to Hawaii together. Nick, good luck with the rest of your time at Case Western Reserve University and keep pounding the strike zone when you are on the mound! Also, thank you to my two Naval Academy sponsor families, the Reduzzi's and Bondura's. I hold our conversations and time together near and dear. I look forward to sharing many more experiences together for years to come.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of The Charles Stark Draper Laboratory, Inc., the United States Navy, the Department of Defense, or the U.S. Government.

This research was supported, in part, by the Draper Fellow Program, by the Office of Naval Research under Grants N00014-16-1-21Add, and by the Army Research Office through the MIT Institute for Soldier Nanotechnologies under Contract W911NF-13-D-0001.

Contents

1 Introduction **17**

2 Multistatic RSN Development **23**

2.1 Preliminaries . . . 23
2.2 System Architecture . . . 24

3 Distributed MIMO RSN **27**

3.1 Principles of FMCW Radar . . . 27
3.2 Simultaneous Radar Transmission . . . 30
  3.2.1 Radar Transmission Scheme . . . 30
  3.2.2 Doppler Compensation . . . 31
  3.2.3 Received IF Signal Decoding . . . 34

4 Target Detection **37**

4.1 Received Signal Spectral Analysis . . . 37
4.2 Monostatic Target Detection . . . 39
4.3 Bistatic Target Detection . . . 40

5 Multiple Target Tracking **43**

5.1 MTT Challenges . . . 43
5.2 MTT Background . . . 45
5.3 MTT Assumptions . . . 46
5.4 MTT Filtering . . . 47
  5.4.1 Prediction . . . 48

5.4.2 Update 49
5.4.3 Multiple Target Inference 53

6 Network Experimentation **55**
6.1 Implementation 55
6.2 Scenario 56
6.3 Network Setup 58
6.4 Performance Evaluation - Target Detection 59
6.5 Performance Evaluation - Target Tracking 60
6.5.1 True Track Development 61
6.5.2 Performance Metrics 62
6.5.3 Multiple Target Tracking Results 63

7 Conclusion **67**

A System Architecture for Multistatic Distributed MIMO RSN **69**
A.1 Synchronization 70
A.2 Communication 71
A.3 Data Collection and Aggregation 71
A.4 Data Analysis 72

B Distributed MIMO Framework **75**

C Hardware Implementation of mmWave RSN **77**
C.1 Synchronization 78
C.2 Communication 79
C.2.1 Dissemination of Shared Chirp Parameters 79
C.2.2 AWR1642 Constraints 80
C.2.3 Firmware Changes 80
C.2.4 Transmission Scheme 81
C.3 Data Collection and Aggregation 81
C.3.1 DCA1000 82

C.3.2 Processors 82

C.3.3 Automation 83

C.4 Data Analysis 84

D Softare Implementation of mmWave RSN 85

D.1 Multistatic Detection 85

D.2 Target Tracking Algorithm 86

D.2.1 Peak Grouping 87

D.2.2 Background Removal 88

D.2.3 LMB Particle Filter 88

E Expansion on OSPA Performance Metric 91

F Single Target Tracking Comparison 93

G Topics for Further Exploration 97

G.1 Software Synchronization 97

G.2 Mobile Node Deployment 99

G.3 Real-Time Processing 99

List of Figures

1-1 Multistatic RSN 18

3-1 Function Block Diagram for FMCW Radar 28
3-2 CDMA Transmission Sequence based on Hadamard Matrix 31

4-1 Spectral Analysis on Received Frame of FMCW Signals 38
4-2 Bistatic RSN Geometry 40

5-1 Measurement Origin Uncertainty 44
5-2 LMB Filter: Prediction 48
5-3 LMB Filter: Update 50

6-1 Implemented mmWave Sensing Node 56
6-2 Indoor Localization Environment 57
6-3 Estimated Target Tracks versus Ground True Tracks 61
6-4 Two Target OSPA Error 64
6-5 Three Target OSPA Error 65
6-6 Three Target OSPA Error (Includes Target Deaths) 66

A-1 Relationship between System Model and System Architecture 70

D-1 Detection Preprocessing: Peak Grouping and Background Removal . 87

F-1 One Target OSPA Error 94
F-2 Estimated Target Tracks versus Ground True Tracks (Single Target) . 95

List of Tables

6.1 RSN Configuration for Network Experimentation 59
6.2 Detection Gain from RSN Spatial Cooperation 60
6.3 Multiple Target Tracking Performance 64

F.1 Single Target Tracking Performance 94

List of Algorithms

1 Doppler Compensation for CDMA-based Signals 34

Chapter 1

Introduction

MILLIMETER-WAVE radar sensor networks (RSNs) promote the further development of the Localization-of-Things (LoT) [1], a key enabler for context awareness in the Internet-of-Things (IoT), the 5G ecosystem, and the global navigation satellite system (GNSS)-denied environments. With 5G unleashing the interconnection of a massive number of small, low-cost radio access technology (RAT)-independent and RAT-dependent devices [2], novel, scalable sensor network localization techniques must be developed and deployed in order to utilize the passive, device-free localization potential these devices offer [3–5]. Realizing these devices' localization potential enables new location-based applications that directly impact the industrial, commercial, and military sectors, prominently autonomous driving [6], healthcare [7], smart cities [8], industrial sensing [9], and rescue operations [10].

However, to form such a RSN, intricate technical challenges must be addressed and accounted for. Strict time, frequency, and phase synchronization among all nodes in the multistatic RSN is paramount for properly communication and resolution of target position information. Furthermore, properly sharing the wireless sensing channel through effective network operation strategies is imperative for interference mitigation. Finally, properly associating the received data to the correct targets of interests within highly cluttered, congested wireless environments is essential for high-performing multiple target tracking (MTT) in millimeter-wave (mmWave) RSNs.

RSNs reap many benefits, such as spatial resilience for target detection and track-

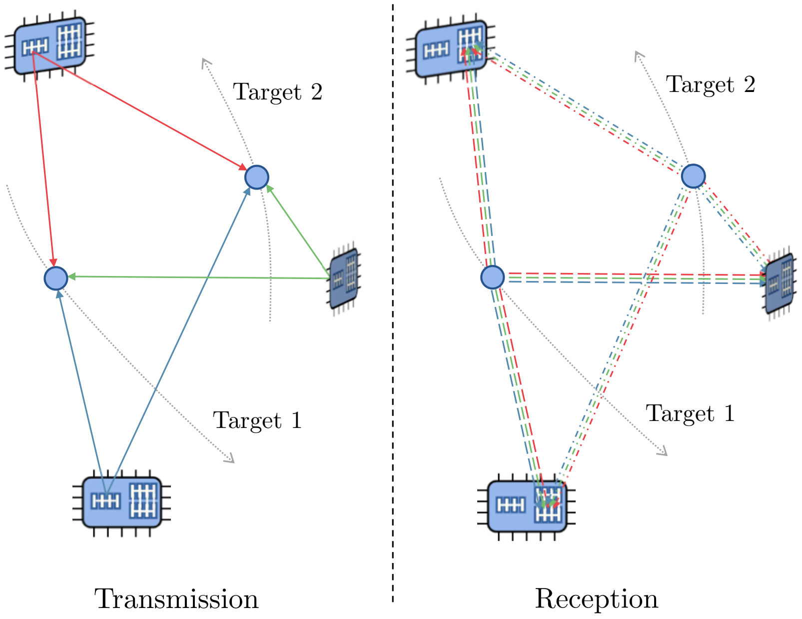


Figure 1-1: A CDMA-based multistatic RSN that resolves all simultaneously transmitted orthogonal signals for location awareness within the sensing environment.

ing and mitigation of inherent radar defects [11–13]. Multistatic RSNs, which employ multiple, spatially diverse radar nodes with a shared area of coverage, offer additional information to detect and track targets as well as mitigate obscuration, fading, and clutter because of increased sensor geometry patterns [14].

Within a RSN, a target's position information may be measured by monostatic radar nodes and bistatic radar nodes. A monostatic radar is a radar in which the transmitter and receiver are collocated and there is a single stable reference oscillator that controls the system's timing and frequency sources [15]. A bistatic radar system is one with spatially separated transmitting and receiving nodes, often with overlapping coverage areas [16] and differing reference oscillators. When a system includes both monostatic and bistatic target detections, the system is a cooperative multistatic RSN [17]. Through this inter-node cooperation, more measurements are collected per time step. For successful data collection and processing, efficient channel sharing strategies must be implemented. Both frequency division multiple access (FDMA) [18] and time division multiple access (TDMA) [19] scheduling strategies have been proven effective. Though, more efficient scheduling strategies are possible. In particular, code division multiple access (CDMA) unleashes the capability for many multistatic radars to simultaneously transmit, even as the RSN increases in size; this approach to channel sharing allows for an exponential increase in the number of spatially diverse detections collected per time step. For an even higher degree of localization awareness, the RSN's algorithms must address both node cooperation [20–22] as well as the data association uncertainty problem [23]. The multi-sensor multi-Bernoulli filter and its related variations is a leader in addressing the data association challenge and achieving MTT [24–27].

Accurate MTT is achieved by exploiting multiple spatially diverse detections per time step. A previously developed system reliably validated non-collaborative target tracking for a single target in a network of low-cost frequency-modulated continuous-wave (FMCW) mmWave devices by employing TDMA [28]. This implementation, though, has two main limitations: the TDMA system was only capable of collecting a fraction of the spatially diverse detections that a CDMA system would collect and the system did not consider the presence of multiple targets. Therefore, this thesis developed an advanced system that utilizes CDMA in a multistatic RSN setting to achieve unparalleled MTT. Hence, our developed system mitigates the previously mentioned limitations and provides a more practical non-collaborative target localization and tracking application afforded to the end user by the same network of low-cost mmWave devices. The fundamental questions relevant to the problem presented in this thesis are:

- How to design an efficient mmWave RSN for localization and tracking of multiple device-free targets operating in a harsh wireless environment?
- How to solve the data association uncertainty problem when tracking multiple targets in a highly cluttered environment?

The answers to these questions led to the design of a new multistatic, multiple-input multiple-output (MIMO) RSN localization system for MTT. Multistatic RSNs that implement CDMA scheduling strategies [29], multi-sensor multi-Bernoulli filtering algorithms [30–32], and advanced statistical signal processing techniques [33] are able to track multiple targets with high precision in harsh wireless environments even though the radar system may produce both false-alarm and missed detections. The developed system employs CDMA to include more radar sensors into the network to transmit and thus improve the efficiency of the multistatic transmission. This system also employs advanced Bayesian filtering techniques to transform the detection data into multiple true target state estimates and to solve the data association uncertainty problem for MTT even within highly cluttered environments. Similar to experimentation in previous works [34–37], our mmWave MTT system verifies its theoretical basis in network localization and navigation (NLN) through indoor localization experiments. This experimentation verifies the accuracy and robustness of the developed system.

This thesis develops a low-cost mmWave MIMO RSN capable of precisely and reliably tracking multiple non-collaborative targets. The key contributions of this thesis are as follows:

- design and implement a slow-time CDMA algorithm that accounts for Doppler shift to mitigate interference and improve the scalability of RSNs;
- develop multi-Bernoulli filtering algorithms that both resolve the data association problem and track multiple targets in a cluttered environment; and
- perform network experimentation in indoor environments to characterize the multi-target detection and tracking performance of the developed system.

The remaining chapters are organized as follows. Chapter 2 presents the preliminaries and an overview of the system architecture necessary for location aware RSNs. Chapter 3 presents the CDMA-based RSN transmission approach. Chapter 4 and Chapter 5 introduce the algorithms for target detection and tracking, respectively. Chapter 6 provides an overview of the system implementation and the performance results from various indoor experiments. Chapter 7 summarizes 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 $\boldsymbol{\mathsf{x}}$ and $\boldsymbol{x}$; a random matrix and its realization are denoted by $\boldsymbol{\mathsf{X}}$ 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. $\boldsymbol{\mathsf{X}} \sim \mathcal{N}(\mu, \sigma^2)$ denotes that $\boldsymbol{\mathsf{X}}$ obeys a normal distribution with mean μ and variance σ^2. The cardinality of $\mathcal{Z}$ is denoted by $|\mathcal{Z}|$. The inner product is denoted with $\langle \cdot\, , \cdot \rangle$. A set of integers $\{i, i+1, \ldots, j-1, j\}$ is represented as $\{i : j\}$.

Chapter 2

Multistatic RSN Development

2.1 Preliminaries

A distributed network of individual radar nodes with index set $\mathcal{A}_\mathrm{r} \in \mathbb{N}$ is considered. Time is discretized and the time samples are indexed by $n \in \{1, 2, \ldots, N_\mathrm{f}\}$, where N_f is the total number of frames during an observation period. The n-th frame duration is denoted by t_n. The number of radars operating in the RSN dictates the specific network configuration. A monostatic radar system, where $|\mathcal{A}_\mathrm{r}| = 1$, is a system comprised of a transceiver, where both the transmitting and receiving components are collocated in a single node. A bistatic radar system, where $|\mathcal{A}_\mathrm{r}| = 2$, is a system comprised of spatially separated nodes, where one is transmitting and another is receiving. A multistatic radar system, where $|\mathcal{A}_\mathrm{r}| \geqslant 2$, is a system built from multiple spatially diverse monostatic and bistatic radar components.

The location of each transmitting and receiving radar is known (i.e., calculated during the observation window or measured prior to the RSN operation) in a multistatic RSN in order to properly resolve the target position information. Accordingly, the known state of radar $i \in \mathcal{A}_\mathrm{r}$ at time n is $\boldsymbol{x}_{i,n}$, which consists of the radar's position, velocity, orientation, and rotational velocity. The position-related component of $\boldsymbol{x}_{i,n}$ is denoted by $\boldsymbol{p}_{i,n}$, the velocity-related component is denoted by $\boldsymbol{v}_{i,n}$, and the rotational velocity-related component is denoted by $\omega_{i,n}$. The state measurements of each radar are with respect to the system's global reference frame.

An index set of non-collaborative targets $\mathcal{A}_\mathrm{t} \in \mathbb{N}$ in a GNSS-denied environment is considered, where each target $j \in \mathcal{A}_\mathrm{t}$ at time n has a d dimensional state vector $\mathsf{x}_{j,n} \in \mathbb{R}^d$ in relation to the system's global reference frame. The state vector is normally constructed with the target's position and velocity components, but other kinematic information may also be considered. Furthermore, the target transition function is entirely dependent on the specific system's implementation and does not drive the algorithm development in this thesis. At time step $n = 0$, it is assumed that the prior distribution of all targets $j \in \mathcal{A}_\mathrm{t}$ are Gaussian distributed, i.e., $\boldsymbol{x}_{j,0} \sim \mathcal{N}(\boldsymbol{x}_{j,0}; \boldsymbol{\mu}_{j,0}, \boldsymbol{C}_{j,0})$. The trace of $\boldsymbol{C}_{j,0}$ is large for all targets as the initial state of target j is uncertain.

At time n, each radar $i \in \mathcal{A}_\mathrm{r}$ in the RSN gathers observations $\mathsf{z}_{i,n}$ relative to $\mathsf{x}_{i,n}$ in its field of view (FOV). These observations may be comprised of the j-th target's range $\mathsf{r}_{j,n}$, radial Doppler velocity $\mathsf{d}_{j,n}$, and bearing $\mathsf{b}_{j,n}$.

2.2 System Architecture

Our developed multistatic, distributed MIMO RSN relies on its well-defined, well-developed system architecture. The system architecture is composed of both hardware and software subcomponents necessary for a scalable, robust, non-collaborative target tracking RSN; these components include node synchronization, node communication, data collection and aggregation, and data analysis. Independent of the specific sensing hardware, node synchronization is imperative for sharing target state information throughout the network. Specifically for the distributed, multistatic network configuration, the radar system must implement both frequency and phase synchronization (FPS) and time syncrhonization (TS) to properly share target state information over the physical layer [38].

Once the nodes achieve FPS and TS, the system must implement an effective and efficient communication strategy among the nodes to mitigate interference in the sensing channel —TDMA, FDMA, and CDMA channel sharing protocols, to name a few. Some channel sharing protocols may also increase the system's efficiency by increasing the network throughput, as our system achieves by implementing CDMA. Finally, the system must support seamless data collection and aggregation between the nodes for data analysis and target tracking. An in-depth discussion on the multistatic, distributed MIMO RSN system architecture is presented in Appendix A.

Chapter 3

Distributed MIMO RSN

The fundamental algorithms that enable multiple target localization and tracking in harsh wireless environments via a mmWave FMCW radar node, which leverages CDMA to achieve distributed MIMO, are presented in this chapter. Appendix B discusses the key differences between distributed and collocated MIMO in relation to radar nodes. Each FMCW radar node presented in this thesis is capable of resolving a target's range, Doppler velocity, and angle of arrival (AoA).

3.1 Principles of FMCW Radar

Linear FMCW is an extremely effective modulation technique in extracting target state information from a cluttered sensing channel [39]. Figure 3-1 presents a functional block diagram of an FMCW radar system that adopts a complex baseband architecture, which is implemented in this thesis.

The transmitted waveform, commonly referred to as a chirp, is a continuous signal with bandwidth B whose frequency is linearly modulated for the chirp duration T_c. Therefore, radar i's transmitted signal $s_i(t)$ can be modeled as [40]:

$$s_i(t) = \cos\left(2\pi \int_0^t f(\xi)\; d\xi\right) \tag{3.1}$$

where $0 \leqslant t \leqslant T_\mathrm{c}$, $f(t) = f_\mathrm{c} + \frac{B}{T_\mathrm{c}}t$, f_c is the chirp's start frequency, and $\frac{B}{T_\mathrm{c}}$ is the chirp

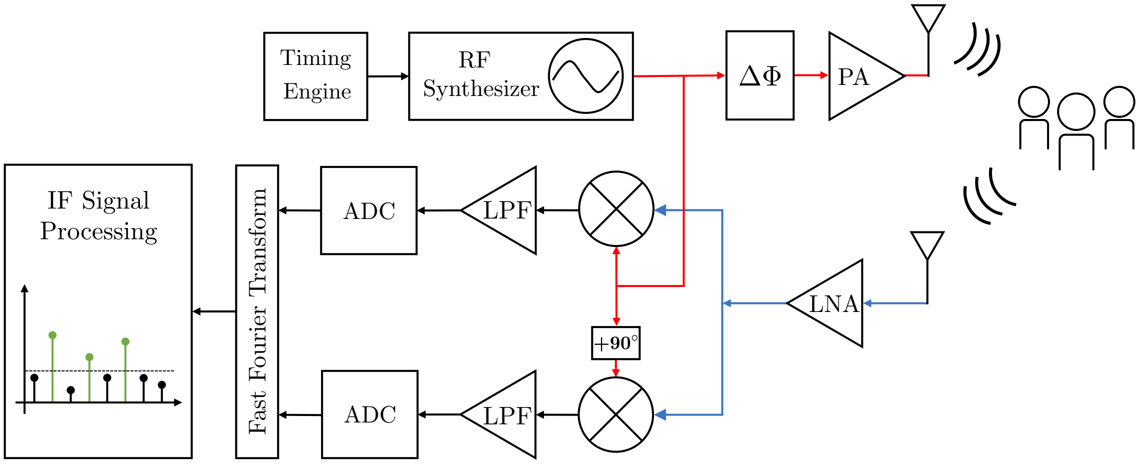


Figure 3-1: The functional block diagram for an FMCW radar is presented, which depicts a single transmitting subsystem and a single receiving subsystem. Properly resolving the target state information relies on the in-phase and quadrature channels.

slope, S.

Assuming that no phase jumps occur during signal reflection, radar i's m-th received signal is the superposition of many damped, delayed versions of the m-th transmitted signal for each target echo and can be characterized by:

$$\mathsf{r}_{i,m}(t) = \sum_{j=1}^{|\mathcal{A}_\mathrm{t}|} \gamma_j s_{i,m}\big(t - \tau_{m,j} + (t + mT_\mathrm{c})\nu_{m,j}\big) + \mathsf{w}_m(t) \tag{3.2}$$

where γ_j is the radar channel gain to include the effects of path loss, antenna gain, and j's radar cross section (RCS), $\tau_{m,j}$ is the range-dependent propagation delay, $\nu_{m,j}t$ is the Doppler shift within the m-th chirp transmission, $\nu_{m,j}mT_\mathrm{c}$ is the Doppler shift across consecutive chirps, and $\mathsf{w}_m(t)$ is an additive white gaussian noise (AWGN) signal. Under the stop-and-hop assumption [39], the range-Doppler coupling term $\nu_{m,j}t$ in (3.2) is negligible and commonly ignored because S is steep enough such that the Doppler shift induced by a target's motion during T_c is insignificant relative to the target's range [41]. For a monostatic chirp, τ and ν are:

$$\tau = \frac{2r_\mathrm{m}}{c} \quad \text{and} \quad \nu = \frac{2v_\mathrm{r}}{c} \tag{3.3}$$

where r_m is the target's monostatic range, c is the speed of light, and v_r is the target's radial velocity relative to the receiving radar.

To resolve target state information, radar i utilizes its receiving subsystem's complex baseband architecture as seen in Figure 3-1, which includes both in-phase and quadrature channels, to produce a complex intermediate frequency (IF) signal. Each channel demodulates the received signal at the radar's quadrature mixers with a reference signal, commonly originating from radar i's transmitting local oscillator (LO). The IF signal produced by radar i's m-th chirp $\mathsf{y}_{i,m}(t)$ is expressed by the following equation [42]:

$$\mathsf{y}_{i,m}(t) = \sum_{j=1}^{|\mathcal{A}_{\mathrm{t}}|} \gamma_j e^{-\imath 2\pi [S\tau_{m,j} t]} e^{\imath 2\pi [f_{\mathrm{c}} \nu_{m,j} m T_{\mathrm{c}}]} + \mathsf{w}_m^{\mathrm{IF}}(t) \tag{3.4}$$

where $\imath$ represents $\sqrt{-1}$ and $\mathsf{w}_m^{\mathrm{IF}}(t)$ is the noise after demodulation and low-pass filtering. In a multistatic configuration, demodulation of the received signal, which includes transmitted signal components originating from spatially separated nodes, is mixed with the receiver's LO.

The frequency difference between the transmitted and received signal reflected off target j is proportional to target j's range and is known as the beat or IF frequency, $f_{\mathrm{IF},j}$ [43]:

$$f_{\mathrm{IF},j} = S\tau_j = S\frac{2r_j}{c}\,. \tag{3.5}$$

Following the mixing process, the IF signal is sampled by the analog-to-digital converter (ADC). The discrete-time IF signal model for multiple targets can be expressed as [44]:

$$\boldsymbol{\mathsf{y}}_{i,m,k} = \sum_{j=1}^{|\mathcal{A}_{\mathrm{t}}|} \gamma e^{-\imath 2\pi [f_{\mathrm{IF},j} k T_{\mathrm{s}}]} e^{\imath 2\pi [f_{\mathrm{c}} \nu_{m,j} m T_{\mathrm{c}}]} + \mathsf{w}_{m,k}^{\mathrm{IF}} \tag{3.6}$$

where T_{s} is the sampling period, $k \in \{1, 2, \ldots, N_{\mathrm{s}}\}$ is the fast-time indices of the N_{s} total samples, and $m \in \{1, 2, \ldots, N_{\mathrm{c}}\}$ is the slow-time indices of the N_{c} total chirps.

3.2 Simultaneous Radar Transmission

Fast-time CDMA and slow-time CDMA are both multiple access techniques that enables joint sensing and communication among networked FMCW nodes while avoiding performance-degrading interference. Fast-time CDMA applies varying phase shifts for the duration of the m-th chirp, whereas slow-time CDMA applies varying phase shifts across the N_c chirps in a frame, such that the m-th chirp has the same induced phase shift based on the implemented CDMA symbol.

Due to particular limitations, such as a node's internal hardware architecture, intra-pulse modulation is not always possible; the readers are directed to [45] for more information. This thesis focuses on slow-time CDMA via binary phase-shift keying (BPSK), such that radar i's transmitted signal for the m-th chirp is as follows:

$$s_i^{\mathrm{BPSK}}(t) = b_{i,m} s_{i,m}(t) \tag{3.7}$$

where $s_i^{\mathrm{BPSK}}(t)$ is the transmitted signal with BPSK and $b_{i,m} \in \{1, -1\}$ is the binary shift. The spatial symbol $b_{i,m}$ is applied for the duration of transmitted chirp m. The transmitted signals are approximately orthogonal if their cross-correlation is zero, such that [40]:

$$\langle s_i(t), s_i^{\mathrm{BPSK}}(t) \rangle = \int_0^{T_\mathrm{c}} s_i(t) s_i^{*\mathrm{BPSK}}(t)\; dt = 0\,. \tag{3.8}$$

To note, the receiving node demodulates the received signals with a non-BPSK reference signal, as the transmitted signal model is presented in (3.1), to ensure the initial CDMA sequences are preserved at the receiver.

3.2.1 Radar Transmission Scheme

Each transmitting radar's binary signature sequence is derived from the Hadamard matrix. The Hadamard matrix is an $\kappa \times \kappa$ square matrix $\boldsymbol{H}_\kappa$ with elements $\{+1, -1\}$, such that $\boldsymbol{H}_\kappa \boldsymbol{H}_\kappa^\top = \kappa \boldsymbol{I}_\kappa$ [46], and whose rows are mutually orthogonal. As the order of $\boldsymbol{H}_\kappa$ gets larger, the construction of the $\boldsymbol{H}_\kappa$ matrix for the purposes of radar transmission sequences needs special attention and is beyond the scope of this thesis [47]. For the purposes of this thesis, $\boldsymbol{H}_4$, which consists of four spatial symbols, forms the transmission sequence block, which is denoted by ε. As seen in Figure 3-2, this sequence of symbols is repeated in a round-robin fashion for the duration of the transmitted frame, where the number of blocks N_b depends on N_c and the size of $\boldsymbol{H}_\kappa$.

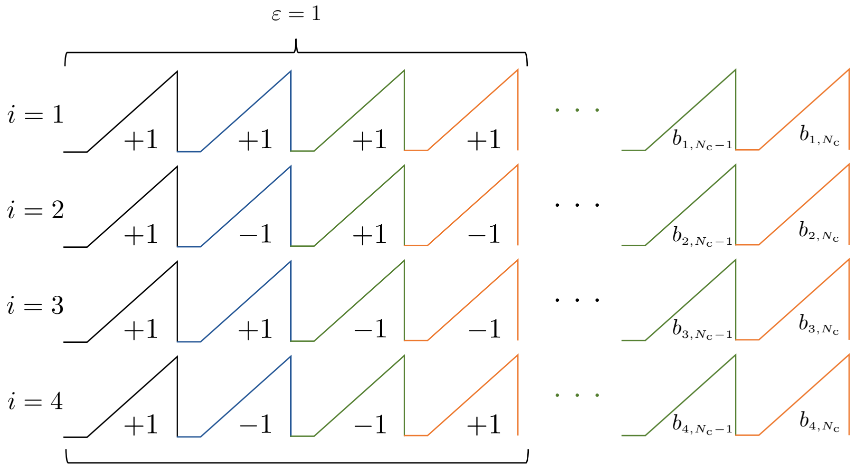


Figure 3-2: The transmission sequence block $\varepsilon = 1$ is shown, which is derived from $\boldsymbol{H}_4$.

The size of the Hadamard matrix is dependent upon the number of simultaneously transmitting radars in the RSN. If $|\mathcal{A}_\mathrm{r}| = 3$, then the RSN transmission scheme would be based off of $\boldsymbol{H}_4$. Each row in the $\boldsymbol{H}_\kappa$ matrix represents a specific radar's transmission sequence ε, whereas each column represents the chirp number in the transmission sequence ε. The sequence derived from the Hadamard matrix is repeated until the desired number of chirps per frame, N_c, is reached.

3.2.2 Doppler Compensation

For each receiver to resolve target position information from each transmitter, it utilizes both their own CDMA sequence as well as all other transmitting radar's orthogonal transmission sequences. If the orthogonality between the transmitted chirps in a decoding sequence is not preserved, which is commonly degraded by Doppler-induced phase shift, the receiving radar will not properly separate all components of its received IF signal. This orthogonality requirement at the receiver for the j-th target is satisfied when:

$$\int_0^{T_\mathrm{c}} s_{i,m,j}(t-\tau_{m,j}+m\nu_{m,j}T_\mathrm{c})\, s^*_{i,m+1,j}(t-\tau_{m+1,j}+(m+1)\nu_{m+1,j}T_\mathrm{c})\; dt = 0\,. \tag{3.9}$$

For a stationary target (i.e., $\nu = 0$), the chirps maintain this orthogonality; however, when $\nu \neq 0$, an additional phase shift is induced onto the received IF signals, negatively affecting the chirp sequence orthogonality. Therefore, a novel Doppler compensation algorithm was developed, as presented in Algorithm 1, to account for this Doppler-induced phase shift across the decoding sequence of chirps.

Suppose a single target of interest is in the sensing environment at range r with Doppler velocity v_r. Furthermore, two subsequent monostatic chirps, m and $m+1$, are transmitted from radar i, where $b_{i,m} = 1$ and $b_{i,m+1} = -1$. Since the target is in motion, a Doppler phase shift $T_\mathrm{c}\nu$, as seen in (3.6), is introduced. Depending on the target's v_r, $T_\mathrm{c}\nu \in [-\pi, \pi]$, such that the target is moving within its unambiguous monostatic velocity resolution limits. This Doppler-induced phase is what deteriorates the orthogonality between the two transmitted chirps. Therefore, the phase needs to be calculated and compensated for.

The phase induced by a single target's motion is found by:

$$\theta_\mathrm{c} = \measuredangle \mathbf{y}_{i,m,k}\mathbf{y}^*_{i,m+1,k} \tag{3.10}$$

where θ_c is the phase due to the target's radial velocity. θ_c is confirmed for the monostatic case by:

$$\Delta f_\mathrm{m} = \frac{2v_\mathrm{r} f_\mathrm{c}}{c} \tag{3.11a}$$

$$\theta_\mathrm{c} = 2\pi\Delta f_\mathrm{m} T_c \tag{3.11b}$$

where v_r is the target's radial velocity directed towards the radar whose signal is being Doppler compensated for. Equations for confirming bistatic θ_c are presented in (4.10) of Section 4.3, which discusses bistatic target detection.

After determining θ_c, $m+1$'s IF signal can be compensated by:

$$\mathbf{y}^{\mathrm{c}}_{i,m+1,k} = \mathbf{y}_{i,m+1,k} e^{\imath\theta_\mathrm{c}} \tag{3.12}$$

where $\mathbf{y}^{\mathrm{c}}_{i,m+1,k}$ denotes the Doppler compensated sampled IF signal for chirp $m+1$.

When Doppler compensating, it is important to preserve the BPSK symbol applied to the transmitted chirp as seen in (3.7) and Figure 3-2 since the sequences these symbols construct are leveraged for received signal decoding.

Doppler compensation for multiple targets in a multistatic RSN is increasingly more difficult to perform compared to a single target in a monostatic setting. In a multistatic RSN where $|\mathcal{A}_\mathrm{r}| = 2$, radar i's IF signals encompass monostatic target reflections, bistatic target reflections, and the distributed nodes' line-of-sight (LOS) transmission.

Each peak corresponds to a component of the receiver's IF signal and has a frequency as seen below:

$$f_\mathrm{p} = \frac{2\, p_{\mathrm{j},\varepsilon}\, \delta r_\mathrm{m}\, S}{c} \tag{3.13}$$

where $p_{\mathrm{j},\varepsilon}$ are the peak indices for CDMA transmission block ε outputted by a target estimator, such as the cell averaging constant false alarm rate (CA-CFAR) algorithm, and $\delta r_\mathrm{m} = c/2B$. The challenge, though, is determining which peak originated from which transmitting radar, calculating the respective Doppler compensation value corresponding to that detection, and preserving the CDMA sequence. Both the v_r and $b_{i,m}$ symbol for each detection are unknown to the receiver. Therefore, Algorithm 1 must be applied to accomplish individual target Doppler compensation in a multi-target, multistatic setting. Note H_BP denotes bandpass filter.

After implementing Algorithm 1, the required condition presented in (3.8) is satisfied for $\mathbf{y}^{\mathrm{c}}_{i,m,k}$ and the chirps in ε for radar i can be decoded.

Algorithm 1 – Multi-target, Multistatic CDMA-based Doppler Compensation

```
for $i = \{1, \dots, |\mathcal{A}_{\mathrm{r}}|\}$ do
    for $\varepsilon = \{1, \dots, N_{\mathrm{b}}\}$ do
        for $m_\varepsilon = \varepsilon\kappa - (\kappa - 1)$ do
            Perform FFT: $\mathscr{F}\{\mathbf{y}_{i,m_\varepsilon,k}\} = \mathbf{y}_{i,m_\varepsilon,k}$.
            Determine all $p_{\mathrm{j},\varepsilon}$ present in $\mathbf{y}_{i,m_\varepsilon,k}$.
            Calculate $f_{\mathrm{p}} \; \forall p_{\mathrm{j},\varepsilon} \in \mathcal{A}_{\mathrm{p},\varepsilon}$ by (3.13).
        end for
        for $m = \{m_\varepsilon, \dots, \varepsilon\kappa\}$ do
            Employ $H_{\mathrm{BP}}(\mathbf{y}_{i,m,k})$ with $f_{\mathrm{p}} \; \forall \; p_{\mathrm{j},\varepsilon} \in \mathcal{A}_{\mathrm{p},\varepsilon}$.
        end for
        Compute $\theta_{\mathrm{c},p_{\mathrm{j},\varepsilon}} \; \forall \; p_{\mathrm{j},\varepsilon} \in \mathcal{A}_{\mathrm{p},\varepsilon}$ similarly to (3.10).
        for $m = \{m_\varepsilon, \dots, \varepsilon\kappa\}$ do
            Apply $\theta_{\mathrm{c},p_{\mathrm{j},\varepsilon}}$ to $\mathbf{y}_{i,m,k}^{p_{\mathrm{j},\varepsilon}}$ components by (3.12).
            $\mathbf{y}_{i,m,k}^{\mathrm{c}} = \sum_{p_{\mathrm{j},\varepsilon} \in \mathcal{A}_{\mathrm{p},\varepsilon}} \mathbf{y}_{i,m,k}^{\mathrm{c},p_{\mathrm{j},\varepsilon}}$.
        end for
    end for
end for
```

3.2.3 Received IF Signal Decoding

After completing Doppler compensation on the received, down-converted chirps, the receiving node must implement a signal decoding algorithm before beginning the multistatic target detection algorithms. The purpose of decoding is to separate the superimposed received signal at each radar based on the originating radar's CDMA sequence. Therefore, the receiving radar can then conduct appropriate information processing for multistatic target detection (i.e., monostatic or bistatic).

Based on the specific order of Hadamard matrix implemented, radar $\tilde{\imath}$'s transmission scheme, $\boldsymbol{a}_{\tilde{\imath}}$, is as follows:

$$\boldsymbol{a}_{\tilde{\imath}} = [\boldsymbol{H}_\kappa]_{\tilde{\imath}}^{\top} \tag{3.14}$$

where $\tilde{\imath}$ is the row in $\boldsymbol{H}_\kappa$ and κ is the dimension of the Hadamard matrix. The $\tilde{\imath}$-th radar synthesizes a nominalized chirp $s_{\tilde{\imath}}$. In accordance with the FMCW functional block diagram in Figure 3-1, each synthesized chirp is phase shifted with the corresponding CDMA sequence:

$$s_{\tilde{\imath}}^{\mathrm{BPSK}}(t) = \boldsymbol{a}_{\tilde{\imath}} s_{\tilde{\imath}}(t)\,. \tag{3.15}$$

In the multistatic network configuration, node i receives its transmitted signal as well as the distributed nodes' transmitted signals:

$$\mathsf{r}_i(t) = \sum_{\tilde{\imath}=1}^{|\mathcal{A}_{\mathrm{r}}|} \alpha_{\tilde{\imath}} s_{\tilde{\imath}}^{\mathrm{BPSK}}(t) + \mathsf{w}(t) \tag{3.16}$$

where $\alpha_{\tilde{\imath}}$ is the position information for all targets in the sensing environment associated with radar $\tilde{\imath}$'s transmitted chirp, $s_{\tilde{\imath}}^{\mathrm{BPSK}}$. Each received signal in decoding block ε is demodulated, sampled, and doppler compensated and in the form of matrix $\boldsymbol{F}_{\kappa\times k}^{\varepsilon}$:

$$\boldsymbol{F}_{\kappa\times k}^{\varepsilon} = \begin{bmatrix} [\mathbf{y}_{i,m_\varepsilon,k}^{\mathrm{c}}]^\top & [\mathbf{y}_{i,m_\varepsilon+1,k}^{\mathrm{c}}]^\top & \dots & [\mathbf{y}_{i,\varepsilon\kappa,k}^{\mathrm{c}}]^\top \end{bmatrix}. \tag{3.17}$$

Receiving node i decodes ε into the respective transmitting radar $\tilde{\imath}$'s signal components as such:

$$\zeta_{i,\tilde{\imath}} = \frac{1}{\kappa} \boldsymbol{F}_{\kappa\times k}^{\varepsilon} \, \boldsymbol{a}_{\tilde{\imath}}^\top\,. \tag{3.18}$$

With (3.14) - (3.18), the target position information $\alpha_{\tilde{\imath}}$ for all received signal components from all transmitting receivers can be derived.

After each node decodes block ε into the respective $\alpha_{\tilde{\imath}} s_{\tilde{\imath}}^{\mathrm{BPSK}}$ components, information processing for multistatic detection for each $\alpha_{\tilde{\imath}}$ component can occur.

Chapter 4

Target Detection

During an observation period, each radar in the RSN transmits N_f frames, which is a fixed series of N_c equispaced chirps. Each frame is processed independently as the recursive, processing algorithms rely solely on the N_c chirps in a frame. Each chirp consists of $f_\mathrm{s}T_\mathrm{c}$ total samples, N_s. If the radar has a uniform linear array (ULA), each incoming signal is received by $N_\mathrm{tx}N_\mathrm{rx}$ receiving antennas. Note, N_tx is equal to the number of collocated transmitters in use during that frame.

One frame of received I/Q samples is encapsulated in matrix $\boldsymbol{D}_{N_\mathrm{c}\times N_\mathrm{s}\times(N_\mathrm{tx}N_\mathrm{rx})}$. Each radar in the network decodes $|\mathcal{A}_\mathrm{r}| \times \boldsymbol{D}$ matrices. One matrix corresponds to the receiving node's monostatic IF signal, whereas the $|\mathcal{A}_\mathrm{r}| - 1$ remaining matrices are decoded, bistatic IF signals. To note, data matrix and data cube are used interchangeably.

4.1 Received Signal Spectral Analysis

The fast Fourier transform (FFT) and appropriate window functions are utilized on each dimension of $\boldsymbol{D}$ to resolve each target's τ and ν from (3.6) [43]. Figure 4-1 depicts the spectral analysis performed on the sampled, received signals.

The 1-D FFT, also referred to as the *Range*-FFT, is performed across the m-th chirp's N_s samples. Following the *Range*-FFT, which is supplemented by a Blackman window, each sample k corresponds to a specific range bin. Based on the range

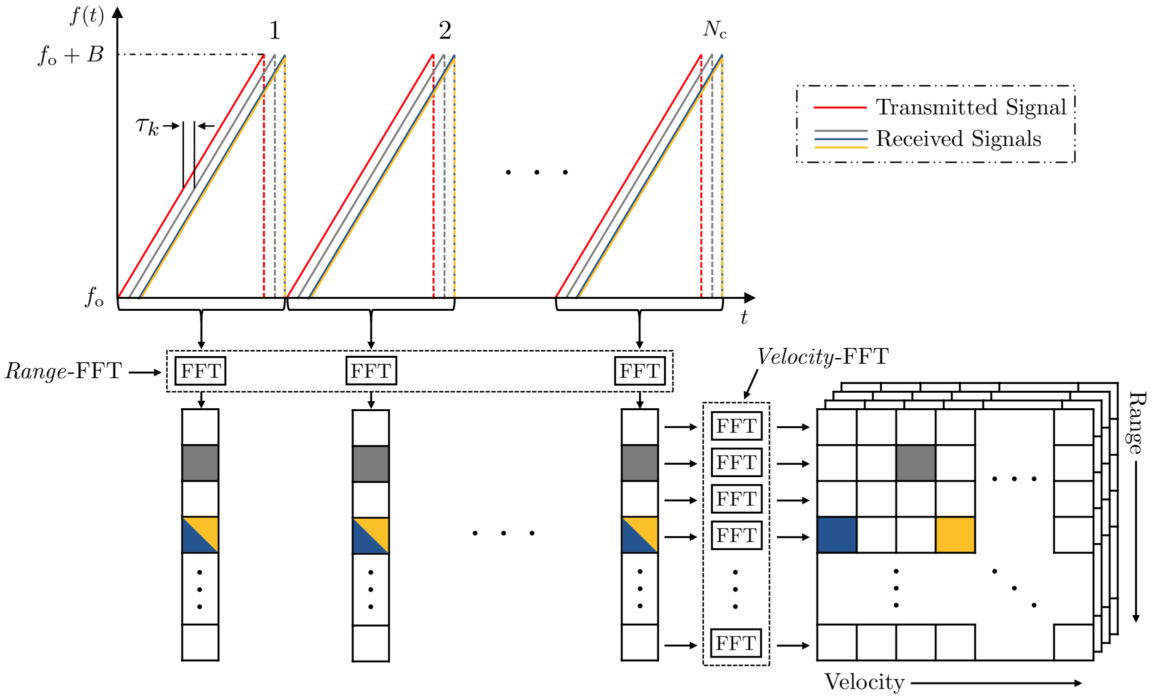


Figure 4-1: Transmission Sequence of N_c chirps in a frame and the corresponding spectral analysis using FFTs, which result in the range-Doppler matrix. Colored cells denote received target echos. The range-Doppler matrix is also adapted to depict the third dimension, $N_{tx}N_{rx}$ for the AoA calculation.

bin, each target's range can be computed. The *Range*-FFT is recursively performed independently for each N_c chirps in a frame. The 2-D FFT, also referred to as the *Velocity*-FFT, is performed for each range bin across all N_c chirps to resolve target Doppler information. A Hanning window is implemented when calculating the 2-D FFT. The range-Doppler matrix, as depicted in the bottom right-hand corner of Figure 4-1, is generated after performing the 2-D FFT.

For each peak value in the range-Doppler matrix, a third FFT, referred to as the *Angle*-FFT, is applied to the third dimension of the $\boldsymbol{D}$ matrix. The *Angle*-FFT exploits the phase change across chirps separated in space. To compute each target's AoA, $N_{tx}N_{rx} \geqslant 2$. A robust target estimator, such as the CA-CFAR algorithm [39], is leveraged to obtain the peak indices from the range-Doppler matrix. Depending on $\boldsymbol{D}$, either monostatic or bistatic detection algorithms are applied to determine target range, Doppler velocity, and AoA.

4.2 Monostatic Target Detection

A target's range is linearly related to its peak frequency value resolved by the *Range*-FFT; therefore, the j-th target's range, r_j, is calculated by multiplying its frequency index $p_{\mathrm{j},\varepsilon}$ by the radar's monostatic range resolution, δr_{m}:

$$\delta r_{\mathrm{m}} = \frac{c}{2ST_{\mathrm{c}}} = \frac{c}{2B} \, . \tag{4.1}$$

A target's Doppler velocity is determined by measuring the phase difference across consecutive chirps. The Doppler phase shift $\Delta\phi_{\mathrm{d}}$ across two consecutive chirps due to a target's v_{r} is:

$$\Delta\phi_{\mathrm{d}} = \frac{4\pi v_{\mathrm{r}} T_{\mathrm{c}}}{\lambda} \tag{4.2}$$

where $\lambda = c/f_{\mathrm{c}}$ is the transmitted wavelength. Each target must move slower than the maximum unambiguous velocity, such that $\Delta\phi_{\mathrm{d}} < |\pi|$. The user-defined chirp length dictates the maximum resolvable radial velocity $v_{\mathrm{m_{max}}}$:

$$v_{\mathrm{m_{max}}} = \frac{\lambda}{4T_{\mathrm{c}}} \, . \tag{4.3}$$

The target's radial velocity, v_{r}, is also calculated by multiplying each Doppler index from the range-Doppler matrix by the Doppler resolution, δd_{m}:

$$\delta d_{\mathrm{m}} = \frac{\lambda}{2N_{\mathrm{c}}T_{\mathrm{c}}} \, . \tag{4.4}$$

Unlike range and velocity, angle estimation is non-linear. The radar's angle estimation is most accurate when the target is at the radar's boresight (i.e., $\theta_{\mathrm{t}} = 0$) and degrades non-linearly as θ_{t} approaches the limits of the radar's FOV. Therefore, angle resolution, $\delta\theta_{\mathrm{t}}$, is approximated by assuming the distance between receiving

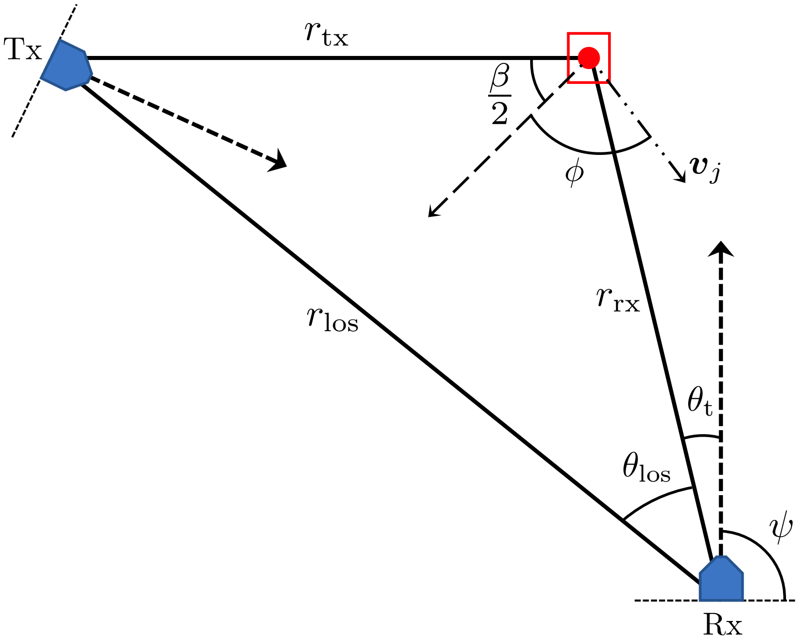


Figure 4-2: Example RSN geometry when $|\mathcal{A}_{\mathrm{r}}| = 2$. The blue pentagons represent the RSN transceivers, whereas the red circle denotes the target.

antennas is $h = \lambda/2$ and the target is at the radar's boresight $\theta_{\mathrm{t}} = 0$, as seen below:

$$\delta\theta_{\mathrm{t}} = \frac{\lambda}{N_{\mathrm{tx}} N_{\mathrm{rx}} h \cos\left(\theta_{\mathrm{t}}\right)} \tag{4.5a}$$

$$\delta\theta_{\mathrm{t}} \approx \frac{2}{N_{\mathrm{tx}} N_{\mathrm{rx}}} . \tag{4.5b}$$

The maximum target AoA, $\theta_{\mathrm{t_{max}}}$, a radar can properly resolve is:

$$\theta_{\mathrm{t_{max}}} = \sin^{-1}\left(\frac{\lambda}{2h}\right) . \tag{4.6}$$

4.3 Bistatic Target Detection

Bistatic target detection is more nuanced compared to monostatic target detection because two way propagation is not in effect. Figure 4-2 presents a portion of a multistatic RSN, which is used to derive the bistatic target detection algorithm.

The bistatic $\boldsymbol{D}$ decoded by radar i incorporate the distributed transmitter's single LOS transmission and $|\mathcal{A}_{\mathrm{t}}|$ target reflections. The LOS transmission arrives before any target detections as it travels the shortest path between the transmitting and receiving radars. All indices detected from the CA-CFAR after the LOS are considered bistatic target detections.

In order to calculate the LOS distance, r_{los}, and each target's bistatic range, $r_{\mathrm{b}} = r_{\mathrm{tx}} + r_{\mathrm{rx}}$, all indices are multiplied by the bistatic range resolution:

$$\delta r_{\mathrm{b}} = \frac{c}{ST_{\mathrm{c}}} = \frac{c}{B}\,. \tag{4.7}$$

A target's bistatic range can be converted to the true target range, r_{rx}, relative to the receiver by:

$$r_{\mathrm{rx}} = \frac{r_{\mathrm{b}}^2 - r_{\mathrm{los}}^2}{2(r_{\mathrm{b}} + r_{\mathrm{los}}\sin(\psi))} \tag{4.8}$$

where $\psi = -90 - \theta_{\mathrm{los}} - \theta_{\mathrm{t}}$. Also, each radar in the network calculates r_{rx} based off a relative orientation to the North rather than a global orientation. This orientation nuance is essential when calculating ψ. θ_{los} and θ_{t} are computed by multiplying the *Angle*-FFT indices by $\delta\theta_{\mathrm{t}}$, similarly to the monostatic case in (4.5b).

Given the FPS condition is satisfied, the bistatic Doppler velocity resolution is derived as such:

$$\delta d_{\mathrm{b}} = \frac{\lambda}{N_{\mathrm{c}}T_{\mathrm{c}}}\,. \tag{4.9}$$

Each target's velocity is calculated by multiplying each *Velocity*-FFT index by δd_{b}. To note, a target's maximum unambiguous velocity is limited in the monostatic case so if the target satisfies (4.3), then a bistatic Doppler velocity can be resolved.

For stationary radar nodes, a target's bistatic Doppler frequency shift, Δf_{b}, is calculated by the below equations. Δf_{b} also serves to confirm the necessary Doppler compensation value, θ_{c}, for subsequent bistatic signals, as discussed in Section 3.2.2:

$$v_{\mathrm{r}} = 2\|\boldsymbol{v}_j\| \cos(\phi) \cos\left(\frac{\beta}{2}\right) \tag{4.10a}$$

$$\Delta f_{\mathrm{b}} = \frac{v_{\mathrm{r}} f_{\mathrm{c}}}{c} \tag{4.10b}$$

$$\theta_{\mathrm{c}} = 2\pi \Delta f_{\mathrm{b}} T_c \tag{4.10c}$$

where $\|\boldsymbol{v}_j\|$ is calculated in (6.1). For more in-depth analysis on the bounds of a target's velocity estimation in a distributed MIMO RSN, the readers are directed to [48].

Chapter 5

Multiple Target Tracking

High performing multistatic MIMO RSNs require robust MTT algorithms capable of jointly detecting and estimating the trajectories of an unknown and time-varying number of targets based on noisy radar measurements. The optimal algorithm depends on the specific system's requirements, constraints, and implementation. A widely accepted approach to address MTT is based on Random Finite Sets (RFS) [49, 50]. Therefore, the Bayesian MTT algorithm presented in this thesis utilizes a framework based on labeled random finite sets (LRFSs) to address the variability of targets present as well as the uncertainty in each target's state [32].

5.1 MTT Challenges

Inherent challenges must be addressed when developing a MTT system. False-alarm measurements and missed detections are two challenges that stem from the non-ideal detection process at each receiving channel. Furthermore, target existence presents another challenge to consider. That is, the number of targets present at any given time is unknown and time-varying.

Data association, also referred to as measurement origin uncertainty (MOU), is an additional challenge that emerges, which refers to the hidden mapping from measurements to target states. Each radar produces measurements that are generated by either valid targets or non-targets (i.e., false-alarms due to non-informative multipath). This mapping process, which is challenged by false clutter measurements, missing target measurements, target births, and target deaths, is depicted in Figure 5-1. Without precise and careful consideration of MOU, the filter may create false target tracks or prune out true ones, and therefore, deteriorate the system performance.

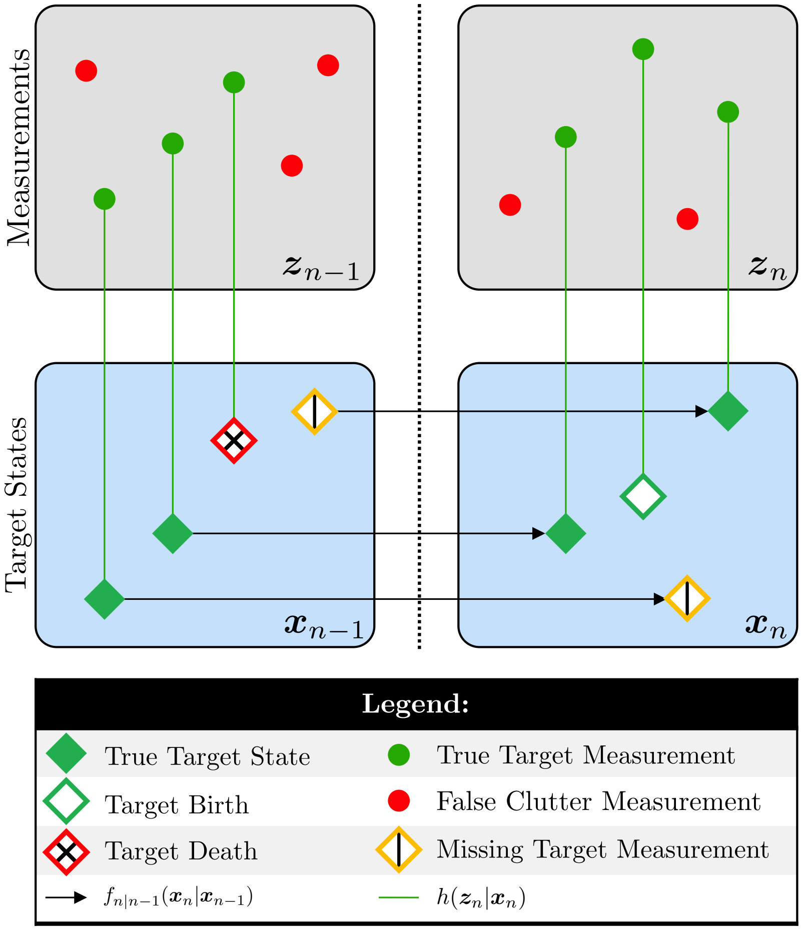


Figure 5-1: MOU provides the greatest challenge for MTT. Reference the legend for the various measurements and target states addressed in our system's data association methods, which utilizes the measurement model $h(\boldsymbol{z}_n|\boldsymbol{x}_n)$ and dynamic model $f_{n|n-1}(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})$.

To overcome these pertinent MTT challenges, the highly efficient labeled multi-Bernoulli (LMB) model for point target tracking is employed [25, 32, 51, 52].

5.2 MTT Background

All target states and measurements are represented as sets and modeled by RFSs, providing our system a unified framework for the MTT problem. Above all else, sets are invariant to order, which provides extreme flexibility in both the measurement and target models. Furthermore, RFSs and their respective properties lend themselves to good evaluation performance metrics for real-world scenarios. In order to deal with multiple targets moving throughout the sensing area, LRFS are introduced.

It is not sufficient for the MTT algorithm to simply estimate each target's location; the algorithm must also attribute a unique label to each estimated target. These labels, also known as target identifiers, help the system preserve situational awareness in the sensing environment. A track is the collection of all target estimates that have the same label across time. Algorithms that consistently produce target estimates, attribute them to the correct label, and sustain such tracks throughout time enhance the RSN performance.

First, let us model the labeled multi-target state by a LRFS X, such that:

$$\mathsf{X} \triangleq \{(\mathbf{\mathsf{x}}^{(1)}, \mathsf{l}^{(1)}), \ldots, (\mathbf{\mathsf{x}}^{(\mathsf{j})}, \mathsf{l}^{(\mathsf{j})})\}$$

where the number of targets, j, is a random variable on $\mathbb{N}$. Furthermore, the target states $\{\mathbf{\mathsf{x}}^{(j)}\}$ are modeled as random vectors on $\mathbb{X}$ and the target labels $\{\mathsf{l}^{(j)}\}$ are modeled as random variables that take values in $\mathbb{L}$. In other words, a LRFS is a random variable taking values in $\mathcal{F}(\mathbb{X} \times \mathbb{L})$. For instance, a realization of LRFS X is modeled as $\mathcal{X} = \{(\boldsymbol{x}^{(1)}, l^{(1)}), \ldots, (\boldsymbol{x}^{(j)}, l^{(j)})\} \in \mathcal{F}(\mathbb{X} \times \mathbb{L})$.

The Bernoulli RFS is the basis for the multi-Bernoulli process. The labeled Bernoulli RFS is used to model a single appearing/disappearing target, such that the set can either be empty (with probability $1 - r(l)$) or have one element (with probability $r(l)$) with a fixed label $l \in \mathbb{L}$. The labeled Bernoulli RFS X is parameterized by $r(l)$ and $p(\cdot, l)$, which correspond to the target's probability of existence and state probability density, respectively. By Bernoulli law, X has a cardinality distribution $p(j) = \mathbb{P}\{|\mathsf{X}| = j\} = r^j(1-r)^{(1-j)}$ for $j \in \{0, 1\}$. One construct that encapsulates the probability distribution of a LRFS is the belief density function $\pi\colon \mathcal{F}(\mathbb{X}\times\mathbb{L}) \rightarrow \mathbb{R}_{+}$. Given that $\mathsf{X} = \{(\mathbf{x}, l)\}$, the belief density function of X is expressed as:

$$\pi_{\mathsf{X}}(\mathcal{X}) = \begin{cases} 1 - r(l) & \text{if} \mathcal{X} = \emptyset \\ r(l)\, p(\boldsymbol{x}, l) & \text{if} \mathcal{X} = \{(\boldsymbol{x}, l)\} \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.1}$$

The LMB RFS is central to MTT and is used to model up to $|\mathbb{L}|$ potential targets. The LMB RFS $\mathsf{X} = \{(r(l), p(\cdot, l))\}_{l\in\mathbb{L}}$ describes a union of independent, labeled Bernoulli RFSs, each parameterized by $r(l)$ and $p(\cdot, l)$. The probability density function (PDF) of an LMB X is defined in [53]. Each distinct label allows for the sequential estimation of target states and their trajectories across different time steps.

5.3 MTT Assumptions

When considering various target tracking systems for multistatic RSNs, each approach varies in both its structure and implementation. Given the MTT challenges in Section 5.1 and the LMB RFS background, a comprehensive set of assumptions made about our dynamic, stochastic tracking system are specified below:

A1) An unknown and time-varying number of targets are considered throughout an observation period. A LMB X_n models the multi-target collection at time n. The label of each Bernoulli component in X_n is defined on the label space $\mathbb{L}_n$.

A2) The targets evolve independently according to the Markovian kernel $f_{n+1|n}(\cdot|\boldsymbol{x}_n, l)$.

A3) The set of targets present at time $n+1$ is the union of new-born targets and surviving targets. An existing target with state $\boldsymbol{x}_n$ and label l survives with probability $P^{\mathrm{S}}_{n+1}(\boldsymbol{x}_n, l)$, whereas new-born targets are modeled via a LMB with parameters $\big\{\big(r^{\mathrm{B}}_{n+1}(l),\, p^{\mathrm{B}}_{n+1}(\cdot, l)\big)\big\}_{l\in\mathbb{B}_{n+1}}$.

A4) At time n, the multistatic RSN produces L total measurement sets from $|\mathcal{A}_{\mathrm{r}}|$

monostatic channels and $|\mathcal{A}_{\mathrm{r}}|^2 - |\mathcal{A}_{\mathrm{r}}|$ bistatic channels. All measurement sets are propagated through the RSN to a centralized fusion node that performs MTT.

A5) The probability of detecting a target with state vector $\boldsymbol{x}$ and label l in sensing channel ℓ at time n is $P^{\mathrm{D}}_{\ell,n}(\boldsymbol{x}, l)$. If detected, the target's state vector $\boldsymbol{x}$ is mapped to a measurement vector $\boldsymbol{z}_{\ell,n} \in \mathbb{Z}_\ell$ through device-specific transformation functions, such as those derived in (6.1). Each measurement vector is characterized by a PDF $h_{\ell,n}(\boldsymbol{z}|\boldsymbol{x}, l)$.

A6) At time n, sensing channel $\ell \in L$ produces a measurement set $\mathcal{Z}_{\ell,n} \triangleq \{\boldsymbol{z}^1_{\ell,n}, \ldots, \boldsymbol{z}^{m_{\ell,n}}_{\ell,n}\}$, such that $|\mathcal{Z}_{\ell,n}| = m_{\ell,n}$, that is affected by MOU. Given that a target can generate at most one measurement per sensing channel, the remaining measurements in $\mathcal{Z}_{\ell,n}$ are false-alarm measurements.

A7) Each measurement set $\mathcal{Z}_{\ell,n}$ is comprised of both target-oriented measurements and false-alarm measurements. False-alarm measurements are collected in both monostatic and bistatic sensing channels and are generated according to a Poisson point process model with an intensity function $f^{\mathrm{FA}}_{\ell,n} \colon \mathbb{Z}_\ell \to \mathbb{R}_+$ for each sensing channel $\ell \in \{1, 2, \ldots, L\}$.

A8) Given the target set $\mathcal{X}_n$, all measurement sets from the various L sensing channels are conditionally independent, i.e., $\pi(\mathcal{Z}_{1,n}, \ldots, \mathcal{Z}_{L,n}|\mathsf{X}_n = \mathcal{X}_n) = \pi(\mathcal{Z}_{1,n}|\mathcal{X}_n) \cdots \pi(\mathcal{Z}_{L,n}|\mathcal{X}_n)$.

5.4 MTT Filtering

The MTT filtering operation recursively predicts, updates, and infers target state information at each time step during an observation period. The filtering operation begins by assuming an initial LMB density $\pi_0(\mathcal{X})$. Figure 5-2 and Figure 5-3 present a schematic for this recursive procedure and complement the discussed LMB filtering equations in the following sections.

5.4.1 Prediction

As depicted in Figure 5-2, prediction begins at time step n, where the prior belief density is $\pi_{n|n}(\mathcal{X})$. The prior belief density is comprised of LMB components with parameters $\left\{\left(r_{n|n}(l),\, p_{n|n}(\cdot, l)\right)\right\}_{l\in\mathbb{L}_n}$.

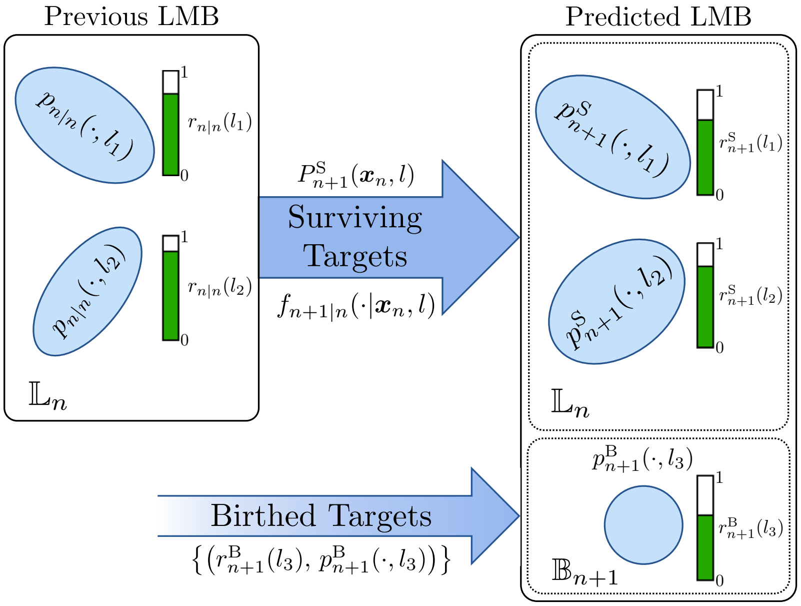


Figure 5-2: During the LMB filter's prediction step, the set of Bernoulli components evolves from time n to $n+1$ by accounting for both surviving targets (components l_1 and l_2) via their dynamic evolution and newly birthed targets (component l_3). The distribution for each Bernoulli component is represented by a blue ellipse.

The predicted LMB and its respective components encompass both surviving targets, which follow the target-specific kinematic model, as well as newly birthed targets. The surviving targets are modeled as a LMB with parameters $\left\{\left(r^{\mathrm{S}}_{n+1}(l),\, p^{\mathrm{S}}_{n+1}(\boldsymbol{x}, l)\right)\right\}_{l\in\mathbb{L}_n}$. The surviving target labels at time step $n+1$ acquire the same labels as those at time step n. Furthermore, the survival probability, which is denoted as $P^{\mathrm{S}}_{n+1}(\cdot, l)$, is employed, in conjunction with the transition density $f_{n+1|n}(\boldsymbol{x}|\cdot, l)$, to reweigh both the predicted existence probabilities as well as the spatial distributions. This is derived in detail below [25]:

$$r^{\mathrm{S}}_{n+1}(l) = r_{n|n}(l)\left\langle P^{\mathrm{S}}_{n+1}(\cdot,l),\, p_{n|n}(\cdot,l)\right\rangle \tag{5.2a}$$

$$p^{\mathrm{S}}_{n+1}(\boldsymbol{x},l) = \frac{\left\langle P^{\mathrm{S}}_{n+1}(\cdot,l) f_{n+1|n}(\boldsymbol{x}|\cdot,l),\, p_{n|n}(\cdot,l)\right\rangle}{\left\langle P^{\mathrm{S}}_{n+1}(\cdot,l),\, p_{n|n}(\cdot,l)\right\rangle}\,. \tag{5.2b}$$

In addition to the surviving targets, newly birthed targets are included in the predicted LMB with new, distinct labels $l \in \mathbb{B}_{n+1}$ and are modeled in accordance with Assumption A3) as birth LMBs with parameters $\left\{\left(r^{\mathrm{B}}_{n+1}(l),\, p^{\mathrm{B}}_{n+1}(\cdot,l)\right)\right\}_{l\in\mathbb{B}_{n+1}}$. Therefore, the predicted LMB is the union of surviving LMB components and newly birthed LMB components, such that $\mathbb{L}_{n+1} = \mathbb{L}_n \cup \mathbb{B}_{n+1}$ and $\mathbb{L}_n \cap \mathbb{B}_{n+1} = \emptyset$. The resulting predicted LMB has parameters

$$\left\{\left(r_{n+1}(l),\, p_{n+1}(\cdot,l)\right)\right\}_{l\in\mathbb{L}_{n+1}} = \left\{\left(r^{\mathrm{S}}_{n+1}(l),\, p^{\mathrm{S}}_{n+1}(\cdot,l)\right)\right\}_{l\in\mathbb{L}_n} \bigcup \left\{\left(r^{\mathrm{B}}_{n+1}(l),\, p^{\mathrm{B}}_{n+1}(\cdot,l)\right)\right\}_{l\in\mathbb{B}_{n+1}}$$

and a belief denoted by π_{n+1}.

5.4.2 Update

The filter's update step leverages the existence probabilities and spatial PDFs of the predicted LMB by propagating them through the update steps to sequentially estimate the LMB's posterior belief density, $\hat{\pi}_{n+1|n+1}(\mathcal{X})$. This posterior belief density is obtained by applying the single-channel update equation [25] recursively for each sensing channel's measurement set [51, Sec. XIII.A].

As stated in Assumptions A6) and A7), the associations between the measurements of $\mathcal{Z}_{\ell,n+1}$ and targets of $\mathcal{X}_{n+1}$ is unknown. Therefore, association maps $a_{\ell,n+1}$ are introduced as a way to manage the MOU. One may think of an association map as a specific numerical assignment $m \in \{0, 1, \ldots, |\mathcal{Z}_{\ell,n+1}|\}$ to each Bernoulli component l, where the assignment m is either a unique index that maps to a specific

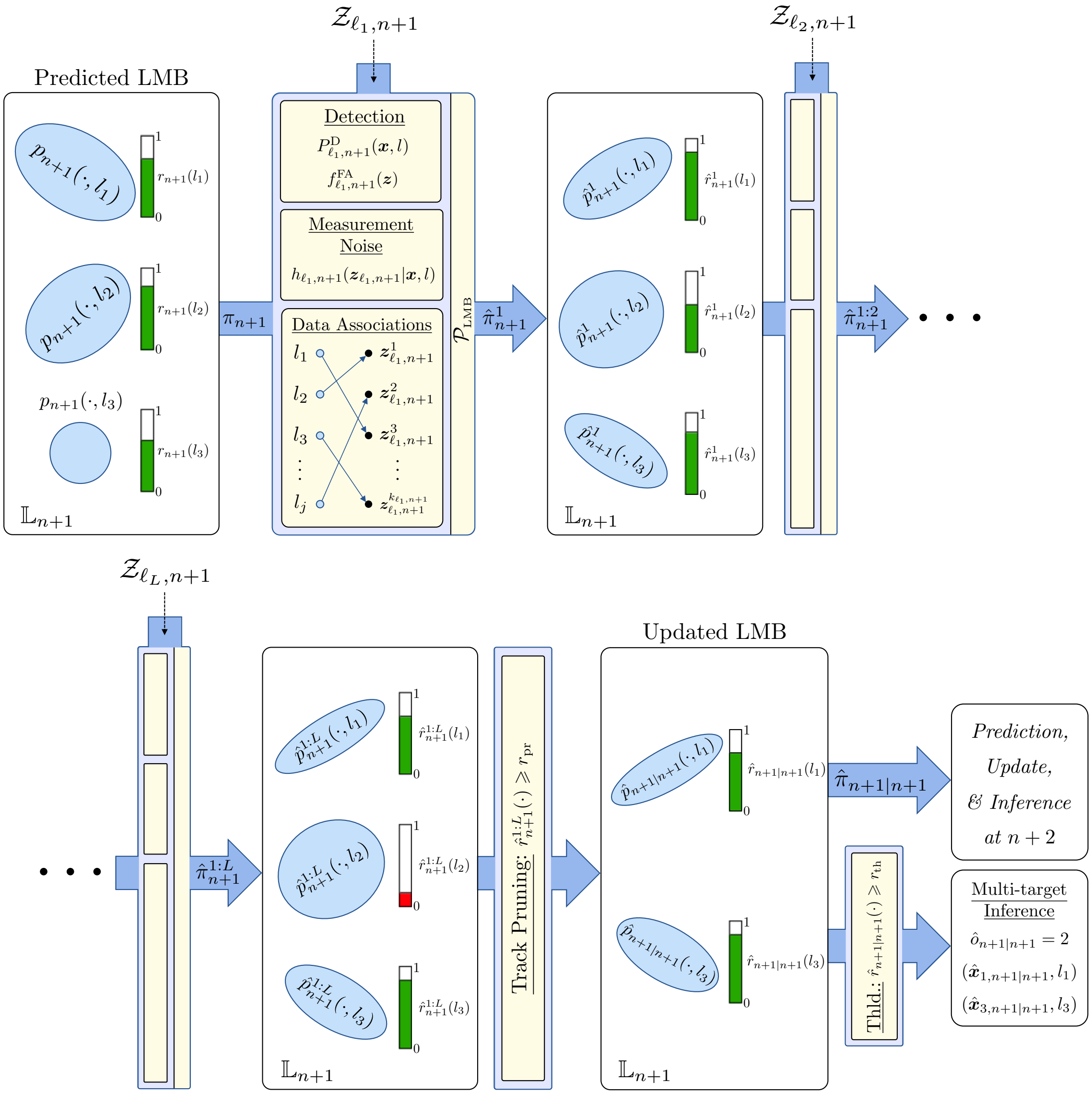


Figure 5-3: To accomplish the LMB filter's multi-measurement update and multi-target inference, an update step is sequentially applied for each collected measurement from all sensor measurement sets. The distribution of each Bernoulli component evolves with every measurement set included, which is represented by the shrinking or expanding of the component's respective blue ellipse.

measurement in $\mathcal{Z}_{\ell,n+1}$ or a misdetection index 0, as seen below [51, Sec.IV-B]:

$$a_{\ell,n+1}(l) = \begin{cases} m \in \{1 : |\mathcal{Z}_{\ell,n+1}|\}, & \text{if tgt } l \text{ generated } \boldsymbol{z}^{(m)}_{\ell,n+1} \\ & \text{in channel } \ell \text{ at } n+1 \\ m = 0, & \text{if tgt } l \text{ was misdetected} \\ & \text{in channel } \ell \text{ at } n+1\,. \end{cases} \tag{5.3}$$

Each Bernoulli component's valid association map $a_{\ell,n+1}$ is part of $\mathcal{A}^{\mathcal{X}}_{\ell,n+1}$, the set of all valid association maps. If the Bernoulli component is not misdetected, $a_{\ell,n+1}(l) > 0$, then no two Bernoulli components can have the same association map; that is, $a_{\ell,n+1}(l) \neq a_{\ell,n+1}(l')$ for $l \neq l'$.

Given an association map m and measurement set $\mathcal{Z}_{\ell,n+1}$, the set-likelihood function $\mathfrak{h}^{m}_{\ell,n+1}(\boldsymbol{x}, l; \mathcal{Z}_{\ell,n+1})$ for sensing channel ℓ is defined in concurrence with Assumption A5) and A7) as:

$$\mathfrak{h}^{m}_{\ell,n+1}(\boldsymbol{x}, l; \mathcal{Z}_{\ell,n+1}) = \begin{cases} \frac{P^{\mathrm{D}}_{\ell,n+1}(\boldsymbol{x},l)\, h_{\ell,n+1}(\boldsymbol{z}^{m}_{\ell,n+1}|\boldsymbol{x},l)}{f^{\mathrm{FA}}_{\ell,n+1}(\boldsymbol{z}^{m}_{\ell,n+1})} & \text{if } m > 0 \\ 1 - P^{\mathrm{D}}_{\ell,n+1}(\boldsymbol{x}, l) & \text{if } m = 0\,. \end{cases} \tag{5.4}$$

Therefore, by combining Assumptions A6)-A8), the association maps defined in (5.3), and the set likelihood function in (5.4), the multi-target likelihood for sensing channel ℓ can be calculated by [53, Prop. 7]:

$$\begin{aligned} \mathfrak{g}_{\ell,n+1}(\mathcal{Z}_{\ell,n+1}|\mathcal{X}) = {} & e^{-\langle f^{\mathrm{FA}}_{\ell,n+1},1\rangle} \left[f^{\mathrm{FA}}_{i,k}(\cdot)\right]^{\mathcal{Z}_{\ell,n+1}} \\ & \times \left[\sum_{a_{\ell,n+1} \in \mathcal{A}^{\mathcal{X}}_{\ell,n+1}} \delta_{a^{-1}_{\ell,n+1}(\{0:|\mathcal{Z}_{\ell,n+1}|\})}(\mathcal{L}_{\mathcal{X}}) \left(\prod_{(\boldsymbol{x},l)\in\mathcal{X}} \mathfrak{h}^{m}_{\ell,n+1}(\boldsymbol{x}, l; \mathcal{Z}_{\ell,n+1})) \right) \right] \end{aligned} \tag{5.5}$$

where $a^{-1}_{\ell,n+1}(\mathcal{B}) \triangleq \{l \in \mathbb{L}_{n+1} | a_{\ell,n+1}(l) \in \mathcal{B}\}$ is the pre-image of the set $\mathcal{B}$ under the mapping $a_{\ell,n+1}$. Furthermore, due to the MOU addressed in Assumption A6), the inner summation is conducted over all valid association maps $a_{\ell,n+1} \in \mathcal{A}^{\mathcal{X}}_{\ell,n+1}$ between the elements of $\mathcal{L}_{\mathcal{X}}$ and the measurements from channel ℓ.

By leveraging assumption A8) and (5.5), the predicted Bernoulli components may be updated via Bayes' theorem for RFSs [53]. As mentioned earlier, the posterior belief density is obtained by sequentially applying the single-channel update equation. Recalling that π_{n+1} denotes the belief of the predicted LMB, the first sensing channel's updated belief density is shown below:

$$\pi^{1}_{n+1}(\mathcal{X}) \propto \pi_{n+1}(\mathcal{X})\, \mathfrak{g}_{1,n+1}(\mathcal{Z}_{1,n+1}|\mathcal{X}) \tag{5.6}$$

where the superscript indicates the channels for which measurement sets have been included in the updated belief thus far. To note, this resulting multi-target posterior belief density, π^{1}_{n+1}, is a mixture of LMB densities, and therefore, no longer an LMB [32].

In order to arrive at a closed recursion, the posterior density is approximated by projecting it onto the family of LMB densities [25], as defined below:

$$\hat{\pi}^{1}_{n+1}(\mathcal{X}) \triangleq \mathcal{P}_{\text{LMB}}\{\pi^{1}_{n+1}(\mathcal{X})\} \tag{5.7}$$

where $\mathcal{P}_{\text{LMB}}$ denotes the projection of resulting LRFS onto the Bernoulli LMB belief densities [32]. The approximating LMB components are denoted as $\hat{\pi}^{1}_{n+1}$ with individual Bernoulli components that are parameterized by $\big\{\big(\hat{r}^{1}_{n+1}(l),\, \hat{p}^{1}_{n+1}(\cdot, l)\big)\big\}_{l\in\mathbb{L}_{n+1}}$.

This projection is achieved via marginalization in two discrete steps. First, each target's marginal association probabilities between each Bernoulli component and each measurement are evaluated using message passing on factor graphs [51]. Second, each of the approximating posterior posterior LMB components of $\hat{\pi}^{1}_{n+1}$ are obtained via a weighted update scheme that relies on all measurements from the current channel ($\ell = 1$) and the previously computed marginal association probabilities [32].

For all remaining sensing channels, the recursive update step is shown below:

$$\pi^{1:\ell}_{n+1}(\mathcal{X}) \propto \hat{\pi}^{1:\ell-1}_{n+1}(\mathcal{X})\, \mathfrak{g}_{\ell,n+1}(\mathcal{Z}_{\ell,n+1}|\mathcal{X}) \tag{5.8a}$$

$$\hat{\pi}^{1:\ell}_{n+1}(\mathcal{X}) \triangleq \mathcal{P}_{\text{LMB}}\{\pi^{1:\ell}_{n+1}(\mathcal{X})\} \tag{5.8b}$$

which is iterated $\forall\, \ell \in \{2, \ldots, L\}$. After each iteration, the projection from (5.7) is also applied. The resulting LMB $\hat{\pi}^{1:\ell}_{n+1}(\mathcal{X})$ has components $\big\{\big(\hat{r}^{1:\ell}_{n+1}(l),\, \hat{p}^{1:\ell}_{n+1}(\cdot, l)\big)\big\}_{l\in\mathbb{L}_{n+1}}$. After all measurement sets have been included, track pruning is implemented for computational and memory tractability. For each iteration, tracks with an existence probability lower than a user defined threshold, r_{pr}, are pruned from the LMB. This approximating LMB $\hat{\pi}_{n+1|n+1}(\mathcal{X})$ serves as the prior belief density for subsequent time steps and is employed to perform multiple target inference.

5.4.3 Multiple Target Inference

After producing the posterior LMB from the filter's update procedures, multiple target inference, as demonstrated in Figure 5-3 as the final update step, is implemented. This inference extracts specific Bernoulli components from the updated LMB with probabilities of existence greater than a specific threshold $r_{\text{th}} \in (0, 1)$.

By declaring a larger value for r_{th}, the filter will extract only those targets with higher probabilities of existence and forgo those that have disappeared or are due to clutter. The cost incurred, though, due to a larger threshold value is not including new-born tracks as quickly (i.e., increase in required $r(l)$ to be confirmed). By declaring a smaller value for r_{th}, the filter will extract new-born tracks more quickly, but at a cost of confirming extraneous, cluttered, or disappeared tracks.

The expected number of targets present (i.e., cardinality) at time $n + 1$ is given as

$$\hat{o}_{n+1|n+1} = \left| \left\{ l \in \mathbb{L}_{n+1|n+1} : \hat{r}_{n+1|n+1}(l) > r_{\text{th}} \right\} \right| . \tag{5.9}$$

Moreover, the target state $(\hat{\boldsymbol{x}}_{n+1|n+1}, l)$ for all confirmed Bernoulli components is inferred as

$$\hat{\boldsymbol{x}}_{n+1|n+1} = \int_{\mathbb{X}} \boldsymbol{x} \, \hat{p}_{n+1|n+1}(\boldsymbol{x}, l) \, d\boldsymbol{x} \, . \tag{5.10}$$

Chapter 6

Network Experimentation

In the following, we present our RSN implementation and several indoor experiments that showcase our developed mmWave RSN's MTT capability.

6.1 Implementation

The developed mmWave CDMA-FMCW RSN employs the Texas Instruments (TI) AWR1642BOOST (AWR1642) to model any commercially available mmWave sensing device [54]. Our system overhauled the pre-existing AWR1642 firmware so that each AWR1642 could communicate based on the multistatic network configuration. Each node in the RSN is time synchronized with the others through a developed triggering circuit. Our system disseminates both a common set of radio frequency (RF) parameters, such as those delineated in Table 6.1, and the $\boldsymbol{H}_4$ CDMA sequences as discussed in Section 3.2 throughout the RSN by a tailored configuration file. Each sensing node, as seen in Figure 6-1, also employs two additional pieces of hardware: the TI DCA1000EVM (DCA) to capture the AWR1642's post-ADC data in real-time [54] and the Raspberry Pi (RPi) to automatically transfer the collected data to the central fusion node for information processing.

Our developed system's information processing algorithms perform Doppler compensation, CDMA decoding, and multistatic target detection as discussed in Chapters 3 and 4. Our MTT discussed in Chapter 5 implements a LMB tracking algorithm

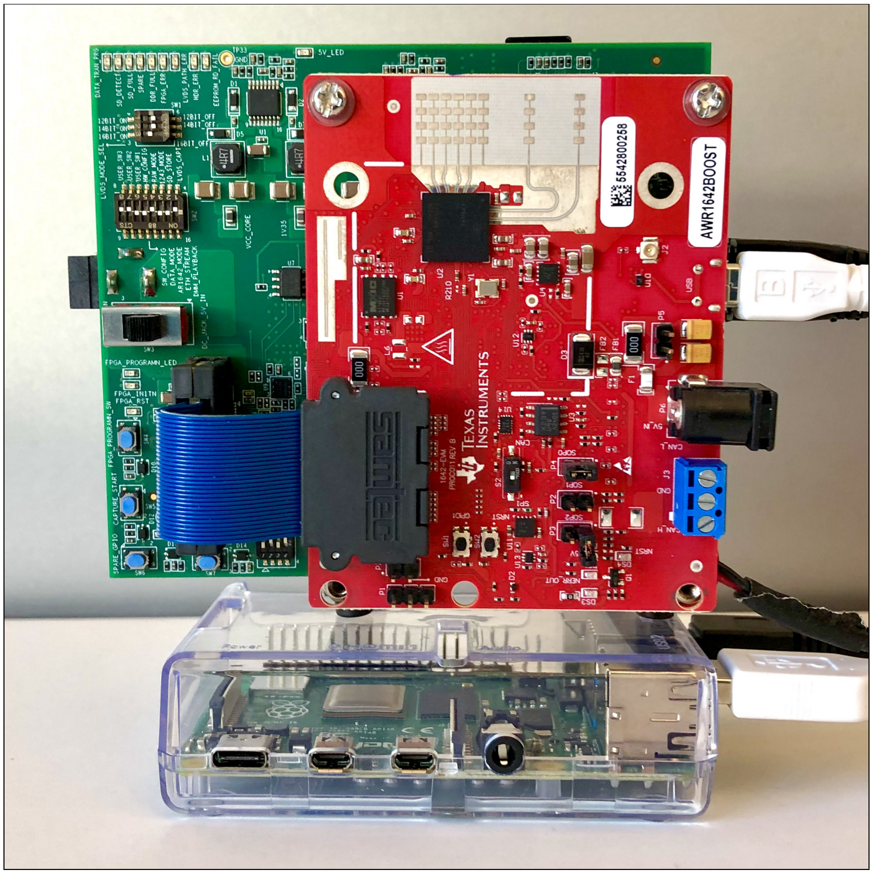


Figure 6-1: Individual mmWave sensing node, which includes one AWR1642 (red board), one DCA (green board), and one RPi.

with a particle representation [55–57] and makes certain probabilistic decisions to intelligently select measurements to reduce the effects from false-alarms and increase them from true target detections. Appendix C and Appendix D discuss our developed system's hardware implementation and software implementation, respectfully, in much greater detail.

6.2 Scenario

All indoor experiments were conducted in a large conference room at the Massachusetts Institute of Technology (MIT). The conference room, which included whiteboards and desks, resembles a typical indoor sensing environment where our system would be deployed. Given the enclosed space and close proximity between radars, clutter and

Figure 6-2: Our developed system was deployed in an indoor localization environment at MIT. The white crosses on the floor are used to recreate each target's ground truth tracks.

wireless congestion is introduced into the sensing environment. Figure 6-2 depicts the physical test setup where our mmWave RSN was deployed.

Various diverse MTT scenarios were conceptualized prior to the start of each experiment with a purpose to demonstrate our developed system's MTT versatility and robustness, independent of the number of targets present in the sensing environment. For instance, situations where targets both appear/disappear from the sensing environment as well as obscure the vision of other radars (i.e., occlusions) are presented to demonstrate the effectiveness of mmWave RSN's CDMA transmission approach. In this thesis, our system's MTT performance is presented for $|\mathcal{A}_{\mathrm{t}}| \leqslant 3$ due to the size constraints of the conference room, which is approximately 6m x 6m.

Each target $j \in \mathcal{A}_{\mathrm{t}}$ is characterized as an average-sized person walking at a normal indoor pace. In this 2-D scenario, all targets have a four dimensional state vector $\mathbf{x}_n = [\mathsf{x}_n, \mathsf{y}_n, \dot{\mathsf{x}}_n, \dot{\mathsf{y}}_n]^\top \in \mathbb{R}^4$, where x_n and y_n are the coordinates that construct the target position $\mathbf{p}_n$ and $\dot{\mathsf{x}}_n$ and $\dot{\mathsf{y}}_n$ are the target velocities along the two axes that construct the target velocity v_n. We consider a white-noise acceleration model [58, Ch. 6.3.2] for the target dynamical model with a sampling period $t_n = 0.25\text{sec}$ and a linear constant velocity transition kernel $f_{n|n-1}(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) = \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{A}_{\mathrm{t},n}\, \boldsymbol{x}_{n-1}, \boldsymbol{Q}_n)$ with

$$\boldsymbol{A}_{\mathrm{t},n} = \begin{bmatrix} 1 & t_n \\ 0 & 1 \end{bmatrix} \otimes \boldsymbol{I}_2 \quad \text{and} \quad \boldsymbol{Q}_n = \sigma_{\mathrm{t}}^2 \begin{bmatrix} \frac{t_n^3}{3} & \frac{t_n^2}{2} \\ \frac{t_n^2}{2} & t_n \end{bmatrix} \otimes \boldsymbol{I}_2$$

where $\boldsymbol{I}_2 = \mathrm{diag}\big\{[1,\, 1]\big\}$ is the identity matrix of size 2 and $\otimes$ denotes the tensor product.

Radar measurements are produced in accordance with $h_{\ell,n}(\boldsymbol{z}|\boldsymbol{x}_n, l) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{g}(\boldsymbol{x}_n), \boldsymbol{R})$, where $\boldsymbol{g}(\cdot)$ is a function that maps target states to radar measurements as defined in (6.1). The measurement covariance matrix, $\boldsymbol{R} = \mathrm{diag}\big\{[\sigma_{\mathrm{r}},\, \sigma_{\mathrm{d}},\, \sigma_{\mathrm{b}}]\big\}$, is commonly determined from radar calibration tests.

This thesis considers a non-linear transformation from $\boldsymbol{x}_n$ to $\boldsymbol{z}$, such that:

$$\boldsymbol{g}(\boldsymbol{x}_n) = \begin{bmatrix} \sqrt{\big(x_n\big)^2 + \big(y_n\big)^2} \\ \sqrt{\big(\dot{x}_n\big)^2 + \big(\dot{y}_n\big)^2} \\ \tan^{-1}\big(x_n/y_n\big) \end{bmatrix}. \tag{6.1}$$

6.3 Network Setup

The network setup of our developed system consists of the configuration of two main parts: the radar, which includes its general and chirp-specific parameters, and the MTT filter. The radar configuration is comprised of a common set of parameters that allow the radars to communicate on the physical layer. Table 6.1 presents the specific parameters that we employed for the environment in Figure 6-2. Table 6.1 also presents the filter-specific parameters employed by the LMB-particle filter (PF) to produce target state estimates. To note, $\boldsymbol{R}$ is the same for each radar deployed as they are identical models from the manufacturer.

Table 6.1: RSN Configuration

	Parameter	**Symbol**	**Value**
General	Chirps per Frame	N_{c}	64 chirps
	Frames per Experiment	N_{f}	240 frames
	Collocated Transmitters	N_{tx}	1
	Collocated Receivers	N_{rx}	4
Chirp-Specific	Start frequency	f_{c}	77 GHz
	Bandwidth	B	4 GHz
	Slope	S	20 MHz/μsec
	Duration	T_{c}	160 μsec
	ADC samples per chirp	N_{s}	512 samples
	ADC sampling rate	f_{adc}	6.20 Ms/sec
	ADC sampling duration	T_{adc}	82.58 μsec
Filter-Specific	Particles Employed	N_{p}	5e4
	Probability of Survival	P^{S}	0.99
	Probability of Detection	P^{D}	0.90
	Probability of Birth	P^{B}	1e−4
	Measurement Variance (Range)	σ_{r}	0.10 m
	Measurement Variance (Doppler)	σ_{d}	1.20 m/sec
	Measurement Variance (Bearing)	σ_{b}	4.00 deg
	Dynamic Noise	σ_{t}	1 m/sec^2

6.4 Performance Evaluation - Target Detection

MTT is only as effective as the quality of measurements produced by the RSN. In this section and prior to presenting the performance evaluation of our MTT algorithm, we examine the improved measurement capabilities brought by the CDMA transmission scheme as opposed to a TDMA scheme. To this end, we introduce detection gain as a performance measure for the RSN measurements. Detection gain quantifies how often the RSN is capable of producing at least one measurement for a target per frame. For a multiple target scenario, the detection gain for individual targets are averaged together. Based on our experimental data, Table 6.2 quantifies the detection gain afforded by the CDMA scheme when a varying number of targets are present and a varying number of radar's measurements are considered when calculating the gain.

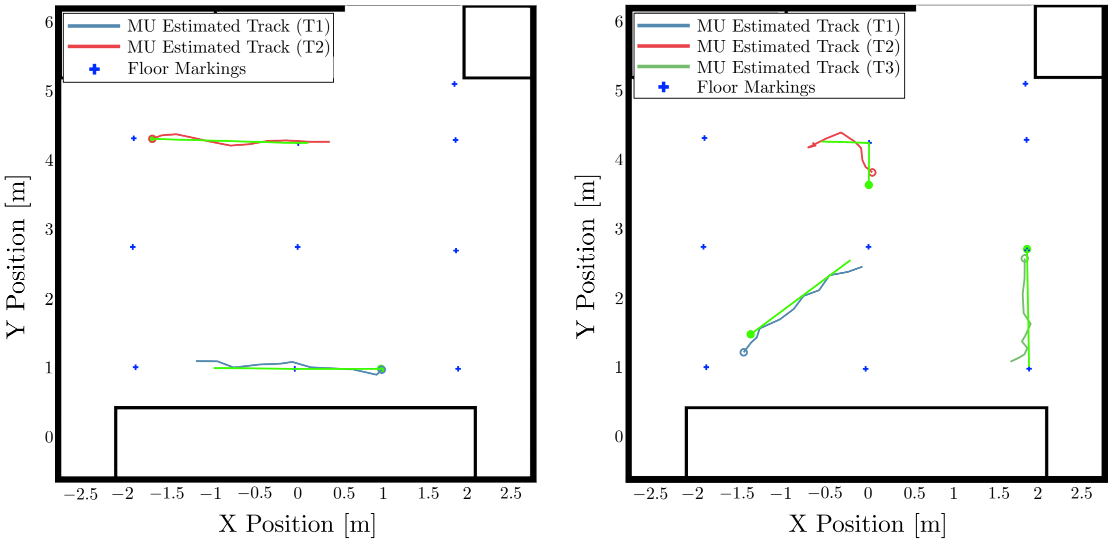
Table 6.2: Detection Gain from RSN Spatial Cooperation

		Number of Targets, $\lvert\mathcal{A}_{\mathrm{t}}\rvert$		
		1	2	3
$\lvert\mathcal{A}_{\mathrm{r}}\rvert$	1	72.22%	67.99%	66.64%
	2	91.05%	88.62%	89.27%
	3	98.77%	97.98%	98.94%

As the number of radar measurements are included, the detection gain increases, further signifying the importance of the simultaneous transmission and reception by the CDMA scheme. To note, the developed mmWave RSN achieves these measures when not all bistatic detections could be resolved due to some AWR1642 hardware limitations. Furthermore, our system accomplishes a higher, unprecedented target detection rate thanks to its CDMA-based transmission scheme that leads to a threefold detection gain increase as opposed to its TDMA-based counterpart [28].

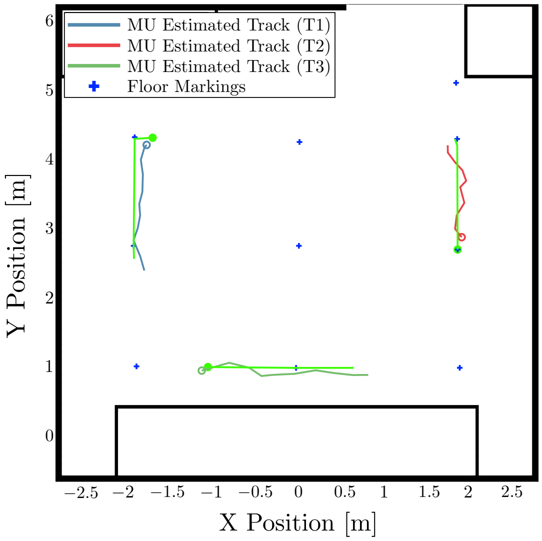
6.5 Performance Evaluation - Target Tracking

Indoor network localization experiments were conducted to evaluate our system's MTT capability. Three multi-target scenarios are presented with an increasing level of complexity due to either an increase in the number of targets present, occlusions, target births/deaths, or a combination of all. All estimated states are assessed against a set of true target tracks, where, with the assistance of the true tracks, each MTT filtering result is evaluated by two main metrics, Optimal SubPattern Assignment (OSPA) and time-on-target (TOT). An additional target tracking scenario is included in Appendix F, which compares the performance of our LMB-PF to the previously implemented Bernoulli-Unscented Kalman filter (B-UKF) in [28].



(a) Two Target Experiment

(b) Three Target Experiment



(c) Three Target Experiment (includes target deaths)

Figure 6-3: Estimated Target Tracks versus Ground True Tracks for various MTT scenarios.

6.5.1 True Track Development

All network localization experiments were video recorded with a GoPro. During RSN operation, each person was instructed to walk with a constant velocity in a direct path between any two floor markings, which are depicted in Figure 6-2. The true track is estimated by leveraging the time stamps when targets arrive at floor markings and linearly interpolating the target position between them. Because non-linear motion is not accounted for in the true tracks, there are marginal discontinuities between the ground truth and implemented true track, most notably when non-linear changes in target trajectory (i.e., when the target makes a sharp turn) occur. There are indications on Figure 6-4 when targets turn corners to signify where the OSPA errors may skewed higher than they realistically should. All true tracks are overlaid on top of all estimated target tracks in Figure 6-3.

6.5.2 Performance Metrics

OSPA and TOT are two metrics used to evaluate the MTT performance. Results for each target tracking scenario are summarized in Table 6.3.

OSPA Metric

The first performance metric employed to evaluate our system's LMB-PF is the OSPA metric [59], which is a distance between sets. At each time step, the OSPA distance is evaluated between the estimated set and true track set of target positions. This OSPA distance metric is comprised of localization and cardinality components. The localization component quantifies the accuracy of each state-estimate, whereas the cardinality component captures the error from false tracks, redundant tracks, or missed true tracks. Appendix E discusses our system's implementation of the OSPA error in greater detail.

TOT Metric

TOT quantifies the tracking algorithm's performance of correctly estimating the presence of targets in the sensing environment. TOT is calculated by computing the percent of time a target is successfully estimated when such a target is present during an observation period. The frames when the target is absent from the sensing environment are not considered. TOT is calculated for each individual true track and a total TOT is computed as the average of all individual TOT, as seen below:

$$\mathrm{TOT}_j = \frac{1}{N_{\mathrm{f}_j}} \sum_{n=1}^{N_{\mathrm{f}_j}} u_{j,n} \tag{6.2a}$$

$$\mathrm{TOT}\,[\%] = \left[\frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} \mathrm{TOT}_j \right] \times 100 \tag{6.2b}$$

where $|\mathcal{T}|$ is the total count of true tracks, N_{f_j} is the total frames target j is present, and $u_{j,n}$ is a propositional variable denoting if target j was successfully detected. Target j is successfully estimated at time n if the LMB-PF produces a state-estimate within a certain distance of the target's true position, such that:

$$u_{j,n} = \begin{cases} 1, & \text{if } \exists\ \boldsymbol{p}_{n,l} \text{ such that } \|\boldsymbol{t}_{j,n} - \boldsymbol{p}_{n,l}\| \leqslant c, l \in \mathbb{L}_n \\ 0, & \text{otherwise} \end{cases}$$

where $\boldsymbol{p}_{n,l}$ is the associated position information of the Bernoulli-component $(\boldsymbol{x}_n, l)$, $\boldsymbol{t}_{j,n}$ is target j's true track position vector, and c is the threshold value, measured in meters.

6.5.3 Multiple Target Tracking Results

Three MTT instances are presented in this section, each of which introduce a more challenging target tracking scenario. This section's results are summarized in Table 6.3 for the first order OSPA distance measuring per-target error for two different threshold values, $c = 0.35$ and $c = 0.5$. This section's OSPA figures employ a threshold value $c = 0.5$.

The first MTT scenario presented demonstrates the performance of the LMB-PF when two targets are present in the sensing environment. With two targets present, randomly walking throughout the sensing environment, the RSN is capable of correctly tracking each target. Other than abrupt changes in target trajectories, which are indicated by green stars, the average OSPA shown in Figure 6-4 is relatively low and within the bounds of the radar's measurement standard deviation for range.

Table 6.3: Multiple Target Tracking Performance

	$c = 0.35$ m		$c = 0.50$ m	
Scenario	**OSPA** [m]	**TOT**	**OSPA** [m]	**TOT**
$\lvert\mathcal{A}_t\rvert \leqslant 2$	0.1835	92.23%	0.1872	99.66%
$\lvert\mathcal{A}_t\rvert \leqslant 3$	0.1878	90.62%	0.1937	99.78%
$\lvert\mathcal{A}_t\rvert \leqslant 3$ (B/D)	0.1985	94.85%	0.1996	100.0%

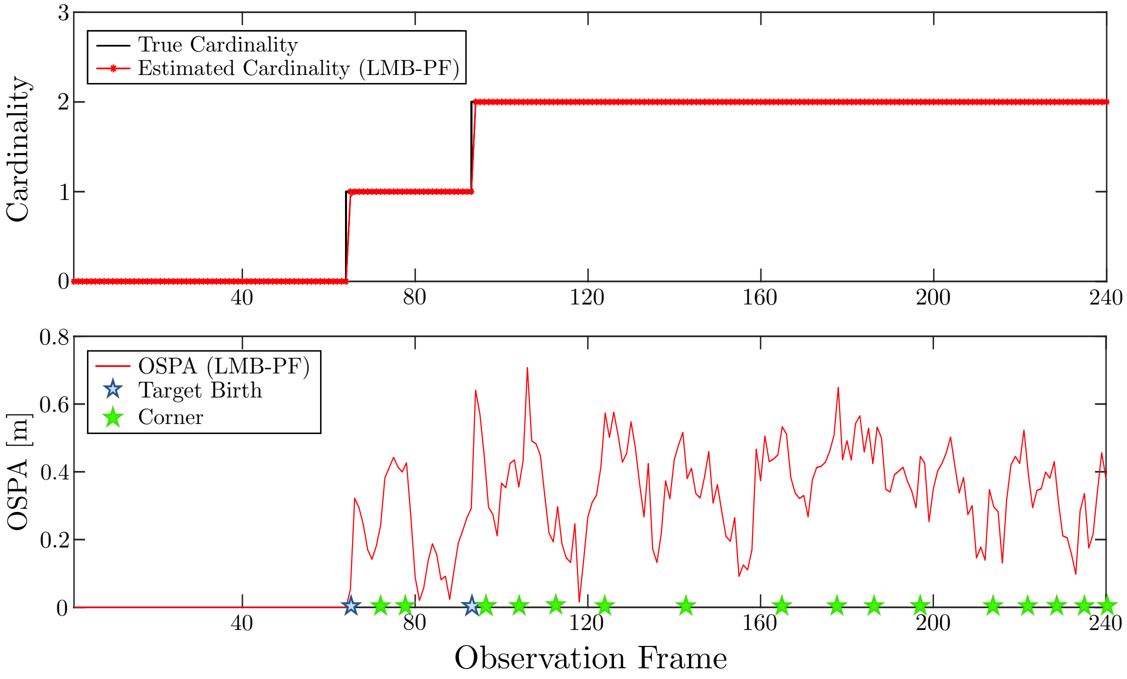


Figure 6-4: Two Target OSPA.

Furthermore, Figure 6-3(a) demonstrates the estimated target tracks versus the target true tracks for the two target scenario.

The second MTT scenario presented introduces another target to the sensing environment. Due to the increase in the number of targets present, there is an increase in occlusions. Our RSN compensates for this challenge by relying on its CDMA-based transmission scheme, such that the LMB-PF may fuse data from all spatially seperated nodes in the RSN to gain a complete picture of the sensing environment. Furthermore, all three targets are capable of entering the sensing environment from any location. There is a uniform distribution of birthing components across the entire

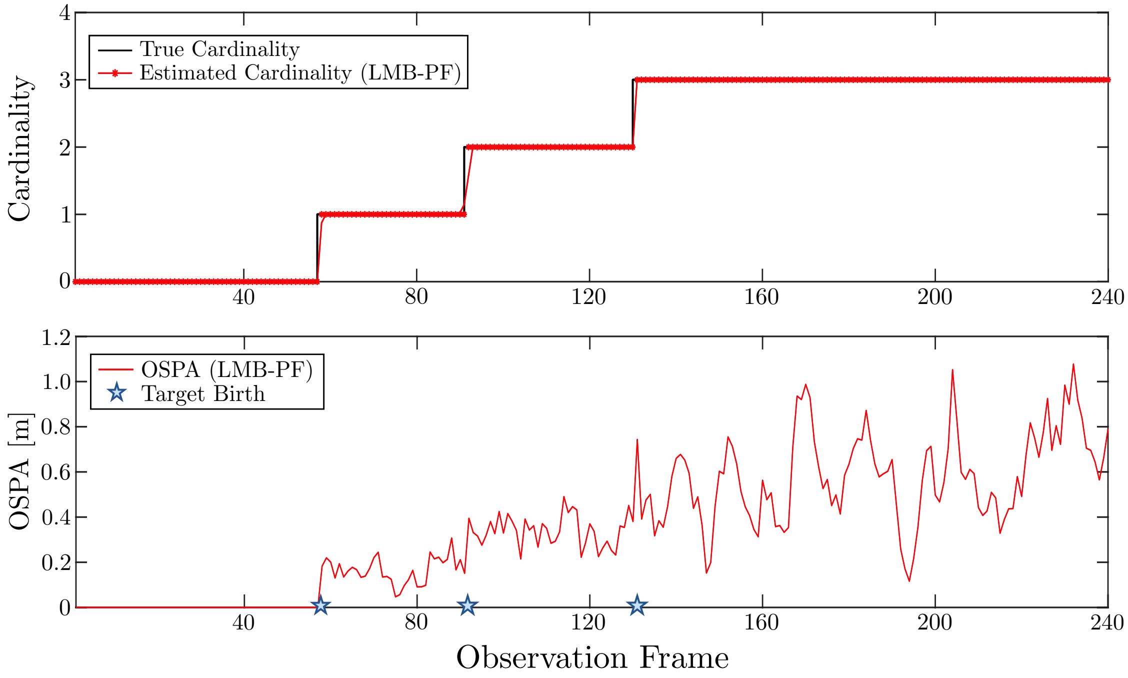


Figure 6-5: Three Target OSPA.

sensing environment so there is no bias towards picking up targets from a certain location. Figure 6-3(b) depicts a snapshot of all three estimated tracks overlaid on the true tracks when all targets were changing trajectories. The LMB-PF is capable of compensating for the sudden changes in trajectory and keeps the estimates close to the true target's location.

The final MTT scenario introduces target deaths. In this indoor localization experiment, similar to the previous scenario, a maximum of three targets are present at one time, which can be seen in Figure 6-3(c). Unlike the previous scenario, though, targets are instructed to leave the sensing environment one by one around the test's halfway mark. The LMB-PF handles target deaths extraordinarily well, as seen in the cardinality graph in Figure 6-6. While there is some delay before the filter picks up the disappearance of a target, this delay is less than one second in length, which is less than four frames. This delay is shown in the cardinality graph from the slanted red lines as well as in the OSPA graph by the magenta stars, which indicate target deaths. Once the target disappears, the OSPA error returns to an average OSPA error. Compared to the other test scenarios in Table 6.3, the total OSPA error does

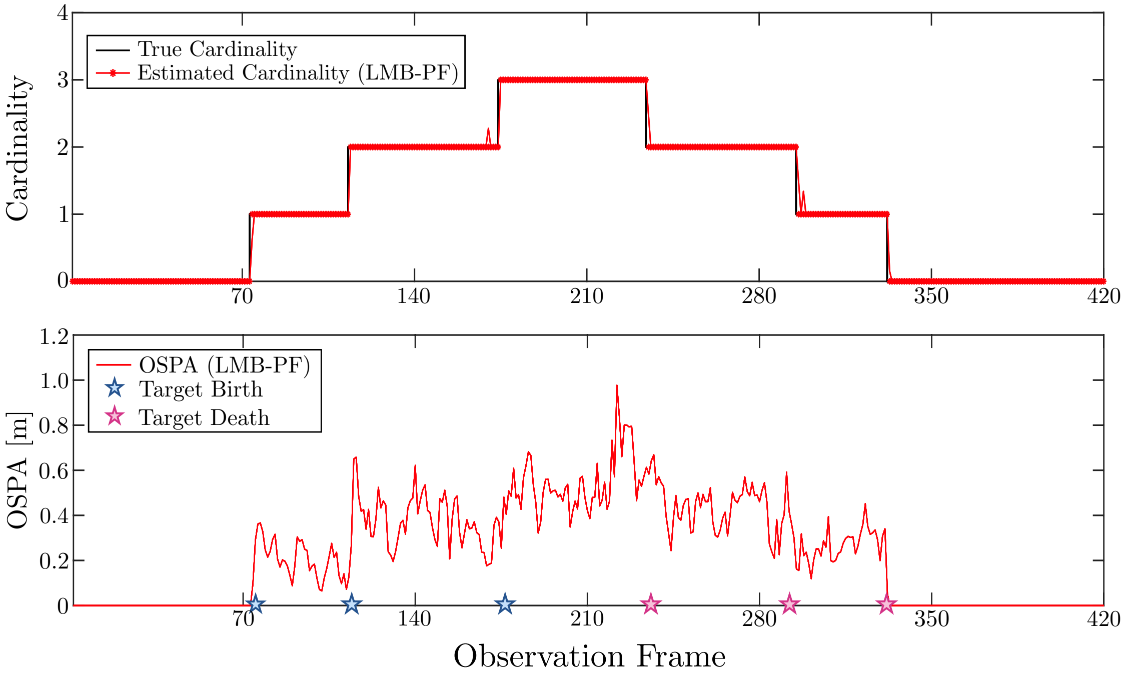


Figure 6-6: Three Target OSPA to include target births and deaths.

not vary significantly based on the complexity of the tracking scenario.

Chapter 7

Conclusion

As the IoT becomes more prominent in our society, the need for efficient and reliable RSNs, which leverage mmWave commercial-off-the-shelf (COTS) devices to localize and track multiple non-collaborative targets indoors, is imminent. Through system implementation and experimentation, we learned multistatic distributed MIMO RSNs that implement both CDMA and multi-Bernoulli filtering algorithms improve system performance for MTT in cluttered environments.

Our developed mmWave RSN integrates these solutions into a coherent, scalable, hardware-agnostic system. The presented techniques allows the developed system to achieve its goal of accurately tracking multiple non-collaborative targets indoors using the mmWave frequency band. By working with low-cost COTS devices, our system is capable of seamlessly integrating with current and future wireless technologies and inheriting their ubiquitous coverage to provide supreme location awareness of the sensing environment through MTT. This system serves as a catalyst for success in many practical applications, such as autonomous driving, indoor localization, search and rescue, security systems, and medical patient monitoring. Topics for further exploration are presented in Appendix G. Finally, the results presented in this thesis confirm pivotal RSN advancements and capabilities, which impact the RSN research community at large as well as the integration of LoT into society today.

Appendix A

System Architecture for Multistatic Distributed MIMO RSN

Multistatic, distributed MIMO RSNs rely not only on a well-defined system model, but also on a well-developed system architecture. The system model integrates a radar measurement model, object motion model, and a network tracking model, each of which touched on in the RSN preliminaries in Section 2.1.

This appendix on the system architecture presents a functional overview of each major hardware and software subcomponent necessary for scalable, robust, non-collaborative target tracking RSNs; these components include node synchronization, node communication, data collection and aggregation, and data analysis. Independent of the specific sensing hardware, node synchronization is imperative for sharing target state information throughout the network. Once the nodes are synchronized, the system must implement an effective and efficient communication strategy among the nodes to mitigate interference in the sensing channel. Finally, the system must support seamless data collection and aggregation between the nodes for data analysis and target tracking. All of these underlying architectural components are discussed in greater depth in the following sections and are synergistically shown together in Figure A-1.

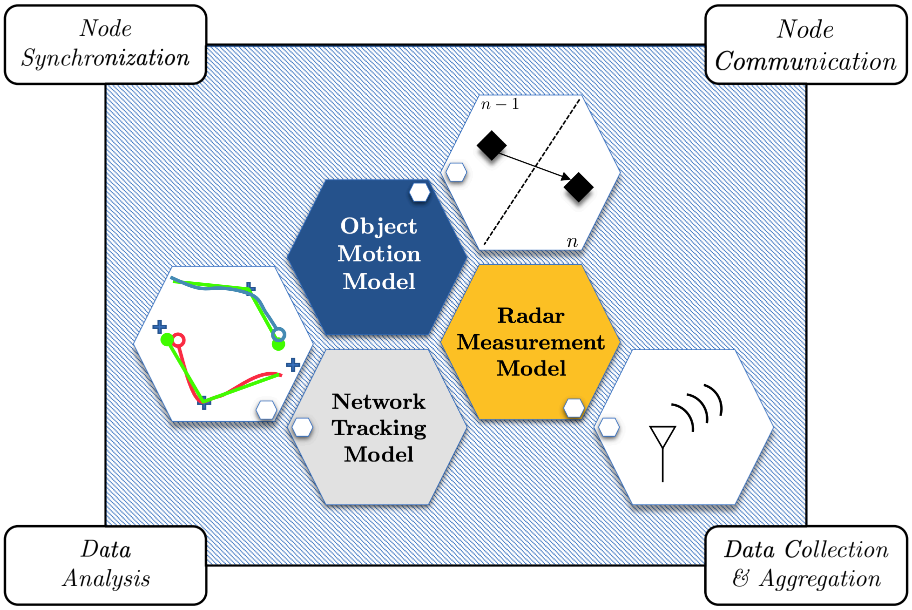


Figure A-1: The synergistic relationship between the system model and system architecture.

A.1 Synchronization

In order to reap the many benefits that distributed systems have to offer, all nodes within the network must be synchronized. Specifically for multistatic, distributed MIMO RSNs, the radar system must implement both TS and FPS to properly share target state information over the physical layer [38]. These synchronization requirements ensure the network's various transceivers have the same reference point for proper signal transmission and reception. Without proper TS and FPS between the nodes, errors in the target state estimates, such as range and velocity, may be introduced into the system and degrade the system's performance.

RSNs commonly achieved synchronization via each node's internal clock, such as their local oscillator [60]. Clock synchronization within the network may be achieved through two distinct approaches: externally or internally. External clock synchronization is achieved via a single external reference signal, which is propagated throughout the network for all nodes to refer to and adjust their own clocks as necessary. Conversely, internal clock synchronization is achieved via a process where each node shares its own clock source with the other nodes and based on these shared reference signals and network timing protocols, all nodes are able to synchronize their clocks with each other. Depending on the network implementation and scale of the system, each synchronization approach has their own advantages and corresponding tradeoffs. With the more centralized, external approach, all nodes refer to a single, more accurate reference signal compared to the decentralized, internal approach. Although, in the event that the external clock's source fails, the entire system cannot be synchronized.

A.2 Communication

Radars communicate through wireless channels by transmitting RF signals into the open air to illuminate targets and detect their range and bearing. Since each radar's transmitted and received signal contains target position information, the nodes in the network must communicate without interfering with each other to preserve such information; therefore, various communication techniques are considered when developing target tracking RSNs. There are many modulation schemes and channel sharing protocols to mitigate this interference—TDMA, FDMA, and CDMA, to name a few. Some channel sharing protocols may also increase the system's efficiency by increasing the network throughput (e.g., CDMA).

Depending on the system's approach to sharing the wireless medium, system-tailored signal resolution strategies are required at the receiver. Without proper channel sharing protocols and signal resolution strategies, the system's performance will be degraded due to increased clutter, missed detections, and false alarm measurements. Each system's tailored communication strategies will also dictate the information gathering and processing architectures.

A.3 Data Collection and Aggregation

Synchronized RSNs capable of communicating with each other are only as effective as their data collection and aggregation procedures. While this component of the system architecture is highly dependent on the system model, the system constrains, the target detection and tracking algorithms, and the various other implementation details, there are specific architectural components that every successful RSN must contain.

Once the communication links between the nodes have been established, the system must be able to configure each node in the network with a set of common system parameters. Common parameters, such as transmission bandwidth and duration, are essential for data collection (i.e., maximizing the collection of accurate target state information). In addition to properly parameterizing the radars, the configuration of each physical node within the sensing environment, such as its orientation relative to the other radars, increases collection rates and therefore, the overall system performance.

Given the radars have collected the target state information locally, data aggregation and fusion must be performed to reap all the benefits of multistatic RSNs. The system can aggregate the collected data through various layers in the Open Systems Interconnection (OSI) model, such as the physical layer, data link layer, and network layer. Additionally, this data aggregation can be performed in both an online and offline manner. Online aggregation requires real-time data processing, analysis, and sharing within the network, whereas offline data aggregation is not constrained to the same time-based requirement. Each approach presents distinct challenges and depends on the system's application and implementation.

A.4 Data Analysis

Data analysis, which is the fourth key component of the system architecture, transforms, analyzes, and interprets the data collected from the RSN and produces target state estimates. Data analysis is similar to data collection and aggregation in that it also depends on the system model implemented. Both the target motion model and radar measurement model dictate the types of data collected and therefore drive how the system will analyze such data. That said, there are some generalized data analysis features that every system should implement for maximum system performance.

Akin to the data sharing and aggregation methods, data processing may happen in both an online and an offline manner. Online processing unleashes real-time target localization and tracking within the system environment for increased practicality; that said, developing and debugging real-time processing algorithms can be difficult and inefficient on the radar system itself. Therefore, once the offline processing chain produces proper state estimates, the processing chain should then be converted to online. Furthermore, the node's memory, energy requirements, and processing capability will drive the specific adaptations necessary for this transformation from offline to online. It is also important to consider where the data is analyzed within the system. The data may be processed in a decentralized manner (e.g., on each individual node) or in a centralized manner (e.g., on a server, main computer). For a multistatic RSN, a combination of both decentralized and centralized processing presents the best way forward since there is a significant performance gain for RSNs due to the fusion of spatially diverse measurements.

Appendix B

Distributed MIMO Framework

Through the use of multiple, simultaneously transmitting and receiving antennas, MIMO radar systems are able to communicate over the same RF channels, increasing the system's throughput and the received signal-to-noise ratio (SNR). MIMO radar systems can be classified into two categories based on their antenna separation: closely spaced, collocated antennas [61] and widely separated, distributed antennas [17]. While collocated MIMO presents certain advantages, such as increased angle resolution through virtual antennas [62], distributed MIMO affords the radar system a spatial diversity gain that is beneficial for MTT; therefore, the distributed MIMO framework is the main framework discussed in this thesis.

Distributed MIMO radar systems can be categorized into two additional classes: coherent MIMO radar and noncoherent MIMO radar. Coherent MIMO radar requires phase synchronization between the distributed nodes in the RSN whereas noncoherent MIMO radar does not. Understanding the differences and intricacies between these frameworks directly affects the system's information processing algorithms, such as the ones discussed in Chapter 4. This thesis focuses on coherent MIMO RSNs.

Since each node in a distributed MIMO RSN typically operates with its own unique LO, the phase of the LO signal may differ among the other nodes in the network. Without perfect phase coherence, the system performance is often degraded [63]. Algorithms such as the master-slave closed-loop algorithm, the round-trip algorithm, and the broadcast consensus-based algorithm achieve phase coherence in MIMO radar systems [64]. These algorithms compensate for the phase offset among the nodes and provide the nodes with a common phase reference.

Once phase coherence has been established among the nodes, channel access methods that utilize orthogonal waveforms, such as CDMA, aided by our novel signal processing approaches to overcome the impact of waveform cross-correlation are implemented in this thesis for proper signal separation at the RSN's receiving nodes.

Appendix C

Hardware Implementation of mmWave RSN

Our developed system employs the TI AWR1642 to model any commercially available mmWave sensing device. The AWR1642 is part of TI's family of automotive radar integrated circuits and is considered a complete radar system on a chip (SoC). Each AWR1642, as depicted by the red board in Figure 6-1, is outfitted with two transmitting and four receiving antennas, all of which have their own respective transmitting and receiving subsystems. Each transmitting subsystem mainly consists of a timing engine (i.e., crystal oscillator), an analog circuit for signal and chirp generation, and power amplifier. Likewise, the essential receiving subsystem components that comprise the complex baseband architecture are the low-noise amplifier, RF mixers for demodulation, ADC, and digital signal processor (DSP). The AWR1642 houses other supporting digital subsystems leveraged by our system, such as an low-voltage differential signaling (LVDS) interface for high speed ADC output and on-chip memory. Greater details about the AWR1642 and its board schematics can be found in [54].

All AWR1642 operations rely on three main software-controlled subsystems: the radar subsystem (BSS), the DSP subsystem (DSS), and the master subsystem (MSS). The BSS includes the digital front end, the ramp generator, and a dedicated Cortex R4F microcontroller unit for RF/analog component configuration, monitoring, and calibration; to note, the BSS is not commercially accessible, which constrains our system's possible improvements. The DSS is responsible for processing all received signals and outputting target detections to the end user. The MSS controls the entire operation of the sensing device, to include the BSS and DSS through dedicated message passing. Since the AWR1642 is inherently designed as a monostatic transceiver out-of-the-box (OOTB), our system implements major firmware enhancements to enable multistatic configuration and ensure proper system performance.

This appendix discusses both the required firmware enhancements as well as our system's approach to node synchronization, node communication, data collection and aggregation, and data analysis.

C.1 Synchronization

Our system synchronizes each node in the network through a hardware-based approach. That is, each node is tethered to a triggering circuit with 22AWG wires. This method of synchronization achieves TS among the nodes and fulfills the RSN TS requirement. For this implementation, our system assumes FPS among all nodes in the network, given that each AWR1642 is the same model and version. Section G.1 proposes software synchronization courses of action, such that FPS is not assumed, but rather ensured in future implementations.

Each AWR1642 has two 20-pin BoosterPack connectors for direct access to the radar's microprocessor. More specifically, each radar's Pin 9 on the J6 connector provides an external synchronization source direct access to the AWR1642's synchronization architecture. The external source our system inputs is from the 22AWG wires connected to the triggering circuit, which are equal in length to ensure that additional synchronization offset is not introduced into the system. The triggering circuit's specific design depends on the RSN size, but in general, it is composed of a breadboard, various resistors, and $|\mathcal{A}_\mathrm{r}|$ bipolar junction transistors (BJTs). Furthermore, since the AWR1642s require 3.3V and 50mA to operate, an external power source provides 3.3V and $|\mathcal{A}_\mathrm{r}|$ * 50mA to the fabricated triggering circuit, which is then distributed to each radar for synchronization. Each radar begins transmission on the rising edge of the 3.3V pulse propagated from the power source to the radar nodes through the 22AWG wires. Depending on the AWR1642's configuration file, which is discussed in depth in the communication section, this pulse can be triggered every frame or once an observation period.

C.2 Communication

Our system establishes and implements a robust, scalable CDMA-based communication framework among the networked nodes. This framework is twofold: first, it defines what each radar physically transmits and second, how each radar transmits it. In order for the AWR1642s to communicate target detection information over the physical layer, they must share a common set of RF parameters. Once these parameters are established among the nodes, each node must communicate orthogonal chirps based on the Hadamard-derived CDMA sequence to mitigate interference and increase network throughput. Since each AWR1642 is designed for monostatic sensor configuration, our system overhauled the pre-existing AWR1642 firmware so that each AWR1642 could communicate based on the developed multistatic framework.

C.2.1 Dissemination of Shared Chirp Parameters

Common RF parameters shared throughout the network are paramount to the performance of any multistatic FMCW-based RSN. For a receiving AWR1642 to dechirp another distributed radar's transmitted signal, the transmitting AWR1642 must also share the same chirp parameters, such as starting frequency, bandwidth, period, and interchirp idle time. Chapter 6 lists all parameters and their values used by our system to collect target detection information. Our system disseminates these common configuration parameters to each node in the network through a tailored configuration file, which is remotely uploaded to each radar through an automation process discussed in Section C.3. Our system's ability to create, send, and input all configuration files to their respective radars through a single Python script enables quick and easy access to the network's sensing capabilities for any user.

C.2.2 AWR1642 Constraints

Once all AWR1642s share a common set of parameters, they must communicate with each other by transmitting orthogonal chirps. Orthogonality among the nodes can be accomplished in a myriad of ways, such as TDMA, FDMA, and CDMA. TDMA has previously been implemented and was therefore not considered in this thesis. When deciding between FDMA and CDMA, our system had to consider three main constraints posed to the system by the AWR1642. First, the AWR1642 internal architecture only provides our system with one chirp generator for all transmitting antennas. Therefore, our system was not able to implement FDMA, which requires different carrier frequencies to transmit (and also receive/demodulate) multiple simultaneous signals. Second, the AWR1642's transmitting antennas are not built with individual, tunable phase shifters, like the AWR1843 is. Rather, the AWR1642s have binary phase shifters. Furthermore, the firmware hardcodes the operation of each of these phase shifters in accordance with the $\boldsymbol{H}_2$ matrix for collocated MIMO since the AWR1642 is inherently monostatic. Our system overrides these firmware changes and liberates the system from this constraint so that the system has full access to each transmitter's binary phase shifters. The firmware changes are discussed below. Third, the AWR1642 binary phase shifters are only capable of slow-time BPSK and not fast-time BPSK (i.e., dynamic intra-chirp phase shifting). This constraint further extenuates the Doppler compensation problem addressed due to the greater length of time between orthogonal chirps in a transmitted sequence.

C.2.3 Firmware Changes

The OOTB AWR1642 version implements collocated MIMO with two transmitters, where the first transmitting antenna always transmits nominal chirps and the second transmitting antenna transmits every other chirp with a binary phase shift. This transmission scheme is also hardcoded into TI's firmware because the engineers designed the radar for monostatic operations. Therefore, our system developed a specialize firmware suite in the C++ programming environment that enables multistatic operation so that each transmitting antenna has the ability to phase shift any chirp simultaneously. These changes elevate the potential to integrate more nodes into the RSN and therefore, enhance the network performance. For proper operation, the developed RSN-specific firmware must be loaded onto the AWR1642 device; directions for these procedures can be found in the AWR1642's Software Development Kit (SDK) [54].

C.2.4 Transmission Scheme

After our developed system's firmware has been loaded onto all AWR1642s, the radars are capable of transmitting properly parameterized, BPSK-enabled chirps. Our developed RSN currently implements a network of three AWR1642s (i.e., $|\mathcal{A}_\mathrm{r}| = 3$). Therefore, each radar transmits BPSK chirps based on their respective row in $\boldsymbol{H}_4$. Each node's orthogonal codes are defined in the configuration file and shared in a similar fashion to how the chirp parameters are shared. Our system's firmware also allows for remote BPSK coding, which provides the user flexibility in changing each radar's specific transmission scheme depending on the test.

C.3 Data Collection and Aggregation

Once triggered by the hardware-synchronized triggering mechanism, the radars begin to transmit chirps based on the parameters found in their configuration files. In the configuration file, each radar's ADC sampling rate is also defined. The ADC uses the defined sampling rate to produce ADC samples. Given that our system performs offline processing, it utilizes TI's DCA to collect the ADC data in real-time and write it to an external processor. This written detection data is then transferred automatically through a RPi information passing infrastructure to our system's central fusion node, where information fusion and processing occurs. Each of the main data collection and aggregation components are discussed in detail in this section.

C.3.1 DCA1000

Rather than utilizing the AWR1642's outputted state estimates that are tailored to a monostatic radar configuration, our system employs the DCA to capture the AWR1642's post-ADC LVDS data for multi-target, multistatic RSN algorithm development. Each DCA, which is depicted by the green board in Figure 6-1, is connected to its respective AWR1642 by a 60-pin Samtec cable; more specifics on the DCA can be found at [54]. Once the LVDS data samples are captured by the DCA, they are converted to binary and streamed over the Ethernet interface to each sensor node's external processor for storage and then dissemination to the central fusion node.

C.3.2 Processors

The RPi serves as the external processor for each of the developed system's sensing nodes; it is cheap, small, and performs all the necessary functions required. Rather than using computers, which have been utilized in previous implementations, our system uses RPis for greater network versatility, such as as the introduction of mobile nodes discussed in Section G.2.

One key factor our system considered when choosing an external processor was packet loss, which is when packets of data are dropped during routing from the DCA to the RPi. Our system could not afford data loss between the DCA and the external processor. As compared to computers, where packet loss can range from 5% to 35% of all data streamed, RPis are capable of consistently achieving 0% packet loss due to their high ethernet throughput.

In order to achieve no packet loss, our system utilizes the RPi's random access memory (RAM) rather than the SD card as the SD card's write mechanism is too slow for the incoming data packets from the DCA. It also preallocates space on the RPi RAM by making a matrix variable and as each packet arrives, the program appends the packet to the variable. Once the desired test is complete, the matrix variable is written to the corresponding test's text file. Once all data is written to the text file, the external processor automatically exports the data to the central fusion node for data processing.

C.3.3 Automation

Our system's design includes a myriad of system automation algorithms to increase the user experience and decrease the potential for human error when conducting sensing experiments (e.g., during node configuration). This novel automation infrastructure relies on our system's wireless ad-hoc network, which is instantiated by the central processor at the beginning of every observation period. Once each RPi is powered on and deployed into the environment, it automatically connects to the ad-hoc network. Given all RPis have connected to the network, the central processor operates each sensing node through the secure shell protocol.

The first task our system automates is the configuration of the RSN, which occurs once each radar's configuration file, to include the common, application-specific set of parameters and the corresponding radar's CDMA sequence, is written and sent to each radar's respective RPi. The RPi interfaces with the AWR1642 via the universal asynchronous receiver/transmitter (UART) protocol over the universal serial bus (USB) interface. Before deployment, each RPi is installed with our system's configuration package, which is a suite of Python scripts that communicates with and automatically configures the AWR1642 before any sensing begins. This infrastructure allows users to deploy all nodes quickly, easily, and without much thought. The next task our system automates is the instantaneous exportation of data to the fusion node. Given that the RSN could be deployed across large swaths of land, expeditious data aggregating is essential. Not only does our system transfer the received data to the central processor, but it automatically prepares the RPi for the next observation period (i.e., clears memory, deletes unwanted configuration files). Our system also automates the preparation of all aggregated data at the central processor for information processing, which is discussed specifically in the following section.

C.4 Data Analysis

Prior to beginning the information processing chain, our system must convert the aggregated binary ADC files into a more workable format, such as a comma separated value (CSV) file. This preprocessing is accomplished by a set of Python scripts located on the central processing node. Once each data file arrives at the central processor, the scripts transform each packet of data, which corresponds to an ADC sample, into a complex number. After this conversion, the Python scripts also produce a log file, which is a combination of both the radar's respective configuration file with all the application-specific parameters and the number of dropped packets during either data collection or aggregation. Once this process is complete, our system begins the information processing chain to produce target state estimates.

Appendix D

Softare Implementation of mmWave RSN

Our developed system's information processing chain deploys novel algorithms to resolve target state information, estimate target position, and produce accurate target tracks over time. This information processing occurs once the central processing node receives all data from the RSN.

D.1 Multistatic Detection

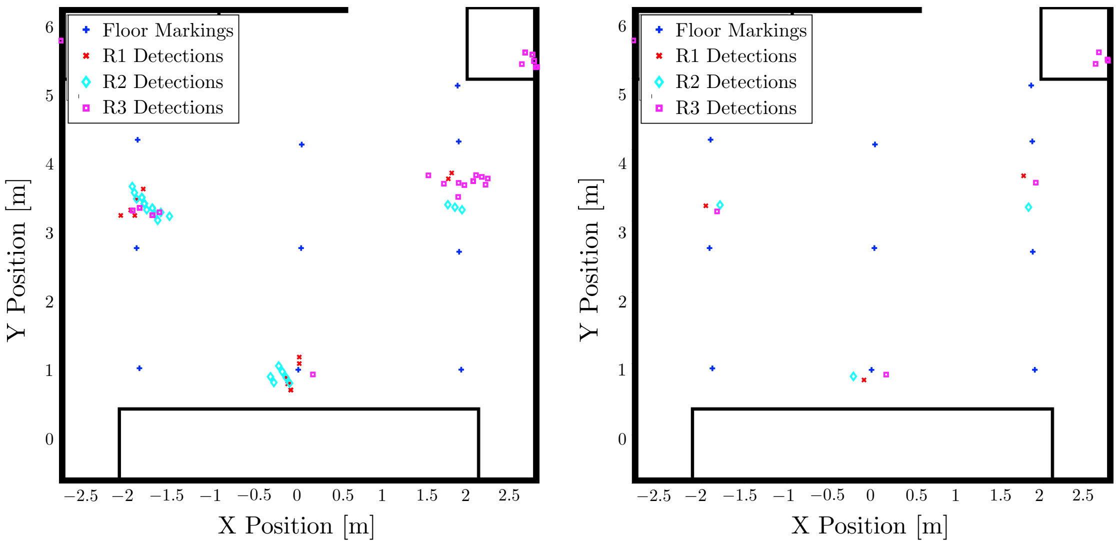
In our system's current implementation, all multistatic detection algorithms are developed and deployed in the MATLAB programming environment. Target detection begins by loading in each radar's CSV data file and respective log file into the MATLAB-based algorithm. Our system utilizes each log file to define variables based on the specific chirp and frame parameters used during the observation period. This design feature provides our system flexibility in processing any RSN configuration. The log file also dictates each radar's BPSK sequence employed during that particular test.

Given the CDMA sequences, the central processor performs both Doppler compensation and IF signal decoding on all data files. Each data file produces one monostatic and two bistatic data sets. Post CDMA-decoding, the monostatic and bistatic detection algorithms are applied to each respective data set. All detections produced from this operation are in reference to each receiver's local reference frame. Before inputting these detections into our system's MTT algorithm, all detections need to be reoriented to the RSN's global reference frame; the global reference frame corresponds to one of the radar's local reference frames, which is predefined in all configuration files. With all detections reoriented to the same, perspective device, our system then interleaves all detections from the same time step together into one matrix such that the tracking algorithm can perform many updates per time step. As mentioned previously, providing the MTT with many spatially diverse measurements every time step is a major benefit of our system's distributed MIMO multistatic RSN.

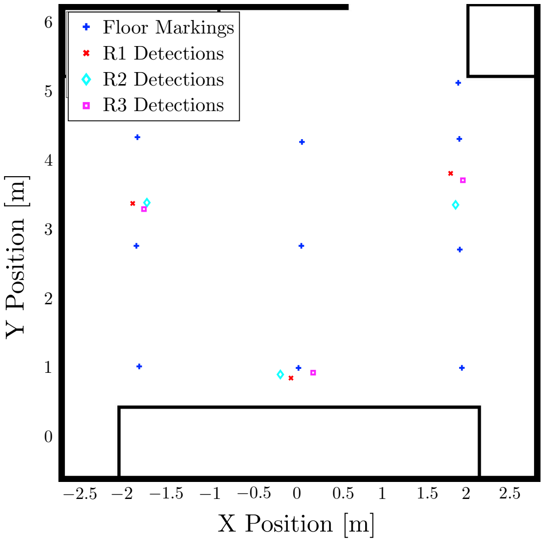
D.2 Target Tracking Algorithm

Our system implements a LMB tracking algorithm with a particle representation [55–57]. The system also assumes a constant target motion model, which is explored in (6.1). For a desired increase in MTT performance, our system produces a greater number of spatially diverse detections per time step from its distributed MIMO RSN. However, this increase in the number of detections per time step poses a tougher data association problem for the system. Therefore, certain probabilistic decisions are made to intelligently select certain data points to reduce the false-alarm rate and increase the target detection probability.

In light of Assumption A6), each target is assumed to generate at most one measurement per sensing channel per time step. In addition to the false-alarm measurements, multiple true radar measurements can be produced by a single target due to the AWR1642's fine range resolution. Furthermore, static targets in the environment, such as walls, tables, and chairs, produce undesirable detections. Therefore, peak grouping and background removal, two additional algorithms, are implemented so that the MTT is provided with the appropriate detection inputs. Figure D-1 demonstrates the preprocessing progression through these algorithms on a set of detections from a single frame.



(a) All Detections (b) Peak Grouping



(c) Background Removal and Peak Grouping

Figure D-1: Our system implements preprocessing algorithms in order to satisfy the single detection per-target assumption made by the MTT. Figure (a) shows all detections collected from one frame. Figure (b) shows the peak grouped detections from the same frame. Figure (c) shows the resulting detections used for state estimates after both background removal and peak grouping.

D.2.1 Peak Grouping

Prior to generating any state estimates, our system distills each target's set of produced detections down to a single averaged peak detection. This grouping reduces the MOU and specifically aids in fulfilling Assumption A6) for the LMB MTT. To accomplish this, our system recursively finds all detections within a pre-defined distance of a certain detection and once all neighbors of that detection have been explored, produces a new, averaged peak detection.

D.2.2 Background Removal

Our system further preprocesses the now-clustered detections by removing all non-informative background detections prior to the MTT producing estimates. Background detections skew the target belief states. Background removal begins by obtaining all detections collected when no desired targets exist in the sensing environment. Then, every detection collected thereafter during observation periods of interest are compared to a threshold corresponding to each background detection. If such a detection crosses that threshold, the detection is considered background and is pruned out of the set of detections for the MTT. Furthermore, our system is able to update the set of background detections depending on how long the observation period is since the sensing environment may change over time.

D.2.3 LMB Particle Filter

The PF is an effective algorithm used to solve a myriad of estimation problems ranging many domains. This filter works exceptionally well in non-linear and/or non-Gaussian state-space systems because rather than restricting the underlying estimation model, which may require linearity and Gaussian PDFs, the PF approximates the posterior PDF by a discrete PDF (i.e., a set of weighted points). Our system implements a LMB-tailored version of the PF to fit the system's non-linear MTT estimation problem.

As discussed in Section 5.2, the LMB RFS X is characterized by the parameter set $\{(r_n(l), p_n(\boldsymbol{x}, l))\}_{l \in \mathbb{L}_n}$, where each individual Bernoulli component indexed by a unique label l has a probability of existence $r_n(l)$ and density $p_n(\boldsymbol{x}, l)$. Our system represents each target density via independent weighted particle sets $\{(w_n^{(l,j)}, \boldsymbol{x}_n^{(l,j)})\}_{j=1}^{N_\mathrm{p}}$, such that the target density is approximated by:

$$p(\boldsymbol{x}_n, l) \approx \sum_{j=1}^{N_\mathrm{p}} w_n^{(l,j)} \delta(\boldsymbol{x}_n - \boldsymbol{x}_n^{(l,j)}) \tag{D.1}$$

where $\delta(\cdot)$ is the Dirac delta measure, with defining properties: (1) $\delta(x-a) = 0$ for $x \neq a$, and (2) $\int_{-\infty}^{\infty} \delta(x-a)dx = 1$.

Each particle, $\boldsymbol{x}_n^{(l,j)}$, represents a possible realization of target l's state vector and has a corresponding weight, $w_n^{(l,j)}$, that dictates the particle's relative importance in comparison to all N_p particles. The larger the weight, the closer in location the particle is to the true target state. Furthermore, the particle sets are considered normalized (i.e., $\sum_{j=1}^{N_\mathrm{p}} w_n^{(l,j)} = 1 \ \forall \ l \in \mathbb{L}_n$) and sequentially propagated in time through the predict (Section 5.4.1) and update (Section 5.4.2) steps [32,51]. After propagating the particles through the LMB equations, the LMB-PF resamples all particles to generate a replacement set of particles, where the number of particles remains unchanged and their weights are typically uniform. However, particles with greater weights prior to resampling are duplicated and particles with negligible contribution to the overall estimate are disregarded. Resampling is meant to conserve computational effort by disregarding non-informative particles and increasing particles effective for estimation [65].

Furthermore, particle sets are generally produced by sampling a particular distribution, commonly referred to as the importance density, to estimate another distribution. In our system's case, samples are generated from sampling the kinematic transition kernel, which is denoted by $f_{n+1|n}(\cdot|\boldsymbol{x}_n, l)$ in Assumption A2) [66–68]. Our system also implements an adaptive birthing approach, where the generation of particle sets are based on the measurement sets themselves [51]. If the measurement has a high likelihood of originating from an existing track, commonly referred to as a surviving target, additional particles are generated following the kinematic kernel. Conversely, if a new target enters the sensing environment and its detection has a low likelihood of originating from an existing track, then the measurement is either considered clutter or a birthed target. For the latter, a new labeled Bernoulli component is initiated from a distribution such that all particles are spread throughout the sensing environment uniformly. In the following time steps, if additional measurements have high likelihoods of originating from previously initiated labeled Bernoulli components, the target's probability of existence increases. Given clutter appears sporadically throughout the sensing environment, the recently birthed labeled Bernoulli component is pruned once the probability of existence is too low.

Appendix E

Expansion on OSPA Performance Metric

The OSPA metric is a statistical distance between the estimated set of target tracks and the ground-truth set of target tracks. OSPA is not conditioned on the temporal history of either the target's true tracks nor their estimated tracks. This metric is calculated for the current time step's corresponding track assignments as follows [59]:

$$\bar{d}_p^c(\mathcal{T},\mathcal{X}) \triangleq \left[\frac{1}{\hat{k}}\left(\min_{\sigma\in\mathrm{Perm}(\hat{k})}\sum_{j=1}^{k}\big(d^c\big(\boldsymbol{t}_j,\boldsymbol{x}_{\sigma(j)}\big)\big)^p + c^p\big(\hat{k}-k\big)\right)\right]^{\frac{1}{p}} \tag{E.1}$$

where $\mathcal{T} = \{\boldsymbol{t}_1,\boldsymbol{t}_2,\ldots,\boldsymbol{t}_k\}$ and $\mathcal{X} = \{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{\hat{k}}\}$ are the ground-truth and estimated track sets, respectively, $\mathrm{Perm}(\hat{k})$ is the set of all permutations of the first $\hat{k}$ integers, and $d^c(\boldsymbol{t},\boldsymbol{x}) = \min(c,\|\boldsymbol{t}-\boldsymbol{x}\|_p)$ is the p-norm distance between vectors $\boldsymbol{t},\boldsymbol{x}$ at cut-off value c, where $\hat{k} \geqslant k$, $p \in [1,\infty]$, and $c > 0$.

This distance metric is comprised of two components, such that each component separately accounts for the localization and cardinality errors between the sets. The localization component of $\bar{d}_p^c$, $\bar{e}_{p,\mathrm{loc}}^c$, quantifies the accuracy of each state-estimate, whereas the cardinality component, $\bar{e}_{p,\mathrm{card}}^c$, captures the error from false tracks, redundant tracks, or missed true tracks, such that:

$$\bar{e}^{c}_{p,\text{loc}}(\mathcal{T},\mathcal{X}) \triangleq \left[\frac{1}{\hat{k}}\left(\min_{\sigma\in\text{Perm}(\hat{k})}\sum_{j=1}^{k} d^{c}\big(\boldsymbol{t}_j,\boldsymbol{x}_{\sigma(j)}\big)^{p}\right)\right]^{\frac{1}{p}} \tag{E.2a}$$

$$\bar{e}^{c}_{p,\text{card}}(\mathcal{T},\mathcal{X}) \triangleq \left(\frac{c^{p}\big(\hat{k}-k\big)}{\hat{k}}\right)^{\frac{1}{p}}. \tag{E.2b}$$

A higher $\bar{e}^{c}_{p,\text{loc}}$ indicates inaccurate target state-estimates, whereas an increase in $\bar{e}^{c}_{p,\text{card}}$ signifies a mismatch between the number of true and estimated tracks. Appropriately choosing values for p and c produces meaningful interpretations of $\bar{d}^{c}_{p}$ as these parameters have a direct affect on how sensitive $\bar{d}^{c}_{p}$ is to outlier state-estimates and differing set cardinalities by imposing varying penalties. For applications that require systems to accurately estimate the position of each target, a smaller value for c is recommended. Conversely, if it is more essential to estimate the exact number of targets, loosening the bounds of c is preferred. In order to keep the penalty for false tracks, redundant tracks, or missed true tracks the same as localization errors, the OSPA metric should be parameterized with $p = 1$ for first-order per-target error, which is measured in meters. Furthermore, by choosing $p = 1$, both $\bar{e}^{c}_{p,\text{loc}}$ and $\bar{e}^{c}_{p,\text{card}}$ can be summed together for a direct interpretation and easier understanding of the OSPA metric.

We evaluate all sensing scenarios with $p = 1$ and cut-off values of 0.35m and 0.50m. These values were chosen based on the number of expected targets in the environment as well as the AWR1642's measurement accuracy. As a reminder, the OSPA metrics are only as reliable as the truth of the produced true tracks. The process by which the true tracks were developed in this thesis are described in Section 6.5.1. Given that the true tracks do not account for indirect target motion as well as abrupt changes in target trajectories (i.e., corners), there are indications on Figure 6-4 and Figure F-1 of such events to signify where the OSPA errors may be skewed based on these occurrences.

Appendix F

Single Target Tracking Comparison

A comparison between our developed system's LMB-PF and a previously implemented B-UKF with sigma point belief propagation is presented. For proper comparison, both target tracking filters are provided the same CDMA-derived measurement set to evaluate their performance, although the previously implemented system was only capable of collecting measurements in a TDMA fashion. Therefore, it is important to note the B-UKF receives a gain in detection quality and quantity from the increase in spatially diverse measurements.

Figure F-1 visually presents the performance evaluation between the two algorithms. Not only does the B-UKF take approximately 3.94x longer to produce a target state-estimate, which is not conducive to real-time target tracking, it's average OSPA with a cut-off value of 0.50m is also approximately 0.45m larger than the LMB-PF's OSPA. Furthermore, for the same set of measurements and true tracks, our system's LMB-PF correctly estimated the true target 10% more often than the B-UKF. Additionally, the LMB-PF is capable of more quickly estimating target births and sudden changes in target trajectories as compared to its B-UKF counterpart.

As mentioned previously, the true track does not perfectly represent the sudden changes in target trajectory. Therefore, green stars are used to indicate when the target turns a corner as this produces an inadvertent increase in OSPA error due to the true track's misrepresentation of true target motion around corners (i.e., acceleration/deceleration). Figure F-2 showcases both filter's estimated target tracks overlaid

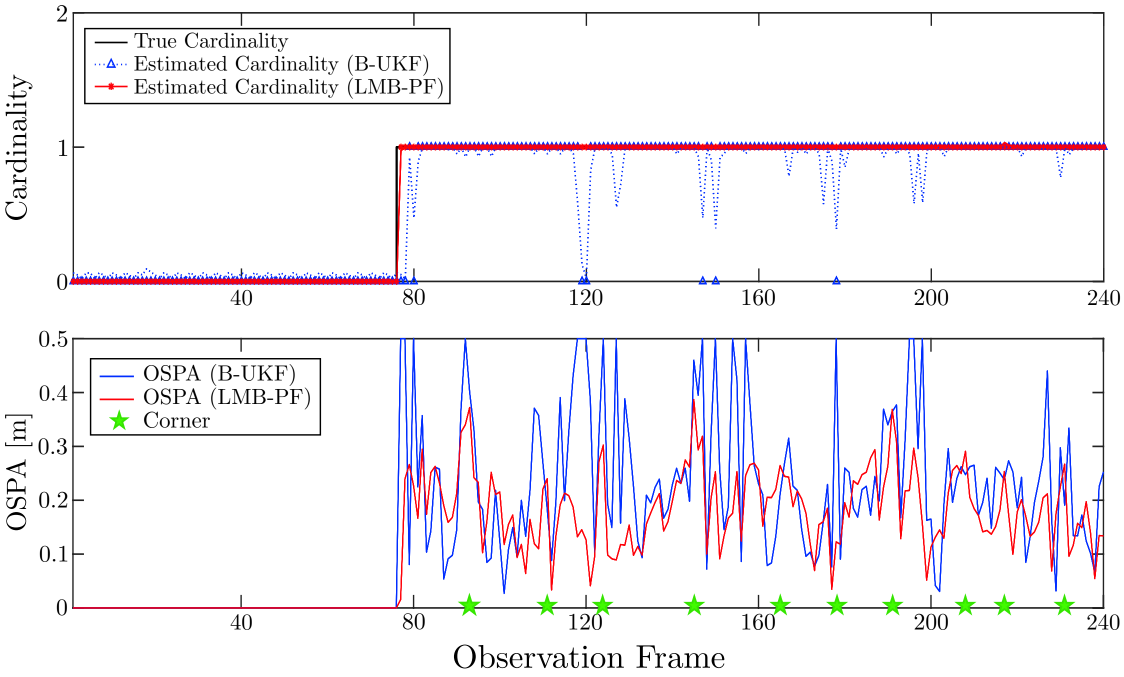


Figure F-1: One Target OSPA comparison between our system's LMB-PF and the B-UKF.

with the ground-truth tracks.

Table F.1 also summarizes both filter's respective metrics relevant to the single target tracking realization.

Table F.1: Single Target Tracking Performance

	$c = 0.35$ m		$c = 0.50$ m	
Scenario	**OSPA** [m]	**TOT**	**OSPA** [m]	**TOT**
$\lvert\mathcal{A}_{\mathrm{t}}\rvert \leqslant 1$ (B-UKF)	0.2211	79.27%	0.2410	89.63%
$\lvert\mathcal{A}_{\mathrm{t}}\rvert \leqslant 1$ (LMB-PF)	0.1855	98.17%	0.1860	100.0%

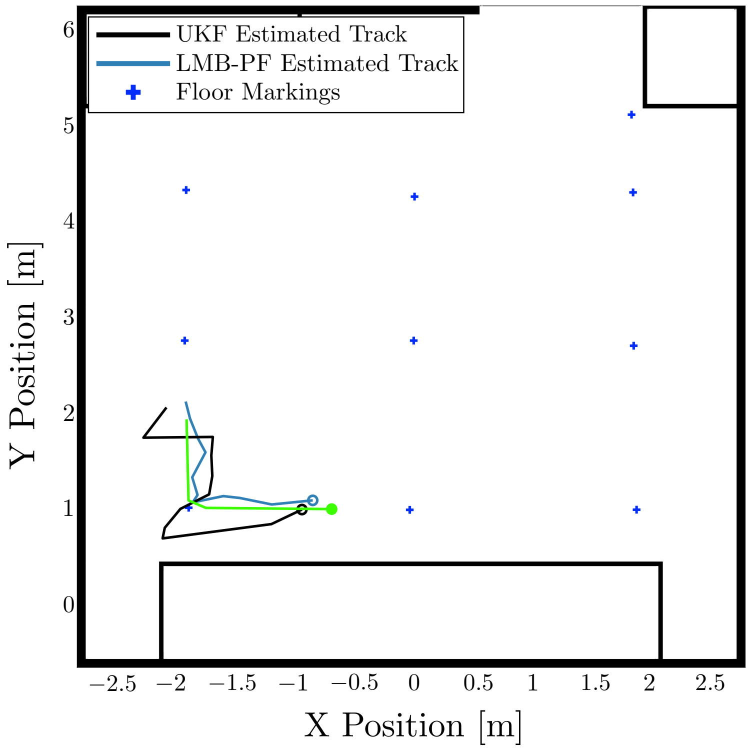


Figure F-2: Estimated Target Tracks versus Ground True Tracks for single target tracking scenario.

Appendix G

Topics for Further Exploration

The developed RSN's system architecture and algorithms are specifically developed to be insensitive to the number of cooperating nodes in the RSN. In order to promote future success in a scaled-up version of our system, future research should be devoted to software synchronization, mobile node deployment, and real-time information processing. Each of these areas have been explored in part to the research conducted for this thesis, but will require more careful consideration before successful integration into our system's current implementation. Below describes each improvement's importance to the overall system performance and recommended actions for implementation.

Furthermore, as part of this thesis, we developed a software-based version of our hardware-based system in the MATLAB programing environment. This system leverages the MATLAB Phased Array toolbox to simulate an FMCW RSN in an indoor environment and produce synthetic target measurements. We designed this system so that these discussed features could be seamlessly integrated into a future implementation of our developed system.

G.1 Software Synchronization

In a distributed RSN, each node is fabricated with its own oscillator. These oscillators produce a periodic, electronic signal, often in the form of a sine wave, to be used by the node as a reference signal. Synchronizing these independently running oscillators is the cornerstone for any scalable, coherent, distributed, all-wireless MIMO RSN [69]. For our system, a software-based synchronization approach would liberate its current implementation from the limitations associated with the hardware-based synchronization approach and provide the network increased versatility and size.

Achieving phase coherence among all nodes in the RSN can be accomplished through various approaches; that is, both passive and active solutions through a range of centralized to distributed approaches that rely on both iterative and non-iterative methods can be implemented [60,64,70–72]. One possible solution introduces a new FMCW hardware architecture, known as over-the-air deramping (OTAD), that utilizes a dual-frequency transmission design [73]. OTAD simultaneously broadcasts two signals from the same transmitter into the sensing environment: one at a lower frequency to serve as the reference signal and another a higher frequency to illuminate the targets in the sensing environment. While a viable option, this solution requires more intricate internal FMCW radar hardware architectures (e.g., additional transmitting and receiving subsystems for dual-frequency transmission/reception to include demodulation), lending itself to increased production costs and not COTS.

Therefore, our system should implement a novel, tailored system similar to AirShare, presented in [74]. Rather than using the AWR1642's internal architecture to achieve phase coherence, AirShare inputs an identical reference signal to each sensor node through an external hardware architecture. This architecture is comprised of one master emitter and multiple receivers, each of which are connected to a single mmWave RSN sensing node, as seen in Figure 6-1. Each AirShare receiver, which is small in size, inputs the reference signal to the AWR1642's input port. As noted in the AWR1642 technical documents [54], the AWR1642 supports an external oscillator that produces either a square or sine wave at 40 MHz. Furthermore, AirShare has the capability to propagate the reference signal over large swaths of area with the use of slave emitters, which regenerate and propagate the master emitter's clock signal. Finally, implementing AirShare will promote mobile node deployment by allowing each node to properly operate without the need to be tethered together by the current

AWG wires.

G.2 Mobile Node Deployment

RSNs with mobile sensing nodes are capable of deploying in a greater number of scenarios and applications (e.g., autonomous unmanned aerial vehicle (UAV) aided search and rescue) as compared to RSNs with stationary nodes. Once a reliable, robust software synchronization architecture is implemented, each radar will be capable of leveraging the multistatic detection algorithms presented in Section 4.3 to communicate to the other radars in the network their relative state (i.e., location, velocity, orientation) through the physical layer (e.g., LOS peak).

Additionally, mobile node deployment will provide the network with more accurate multistatic detections. For instance, in scenarios with stationary RSNs, stationary targets are not resolved with as high of certainty as they should be. Therefore, with the help of the RSN's relative motion, these stationary targets will be resolved and processed more confidently. Furthermore, mobile nodes have the flexibility to move and maneuver dynamically depending on how the sensing environment evolves. As discussed in Section 4.2, the radar's detection accuracy decreases as targets move towards the radar's broadside. If the nodes are mobile, they will be able to reorient themselves during sensing and therefore, further increase the multistatic detection capability by keeping the targets at boresight. Also, as the targets depart the sensing environment, the mobile nodes will be capable of traveling with the targets, providing increased situational awareness to the end user.

G.3 Real-Time Processing

The third and final proposed advancement for the next iteration of our system is to transform the implemented offline information processing into real-time processing [75]. In our system's current state, each radar sensor collects and exports the raw ADC data to its individual processor. From there, the entire data set is automatically transferred to a centralized computer for processing and fusion. While effective, this process is time consuming due to the large amount of data transferred wirelessly. With this in mind, real-time processing will provide our system's end users two key benefits: a cheaper, smaller RSN and near real-time target state estimates for immediate visualization.

Each of the mmWave RSN sensing nodes are currently comprised of both the AWR1642 radar sensor as well as the DCA1000 real-time data capture card. The DCA1000 is essential for our system's current system implementation, but will be obsolete with real-time processing as all data will be processed on the AWR1642's internal digital signal processor. As the network scales in size, disregarding the DCA1000 will reduce each node's form factor and drive down the RSN production cost, truly making our system a low-cost system.

To accomplish this, all information processing algorithms need to be converted to the C++ programing environment and flashed onto the AWR1642 as per the instructions found in the radar's SDK [54]. This flashing will replace the default algorithms in TI's current DSS. These real-time processing algorithms will provide the end user with near real-time target state estimates, which is more practical for target tracking. The current system implementation has also already established a robust infrastructure for communication between each node and the master computer. Therefore, as one detection is being processed on the AWR1642, another can simultaneously be sent to the fusion center for target tracking and visualization.

Bibliography

[1] 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.

[2] S. Bartoletti, A. Conti, D. Dardari, and A. Giorgetti, "5G Localization and Context-Awareness," in *5G Italy White eBook: from Research to Market*, 2018, pp. 167–187.

[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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] K. Shafique and B. A. Khawaja, "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios," *Special Section on Antenna And Propagation for 5G and Beyond*, January 2020.

[9] 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.

[10] 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.

[11] J. Liang and Q. Liang, "Design and Analysis of Distributed Radar Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 11, pp. 1926–1933, November 2011.

[12] 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.

[13] 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.

[14] A. Ledergerber and R. D'Andrea, "A Multi-Static Radar Network with Ultra-Wideband Radio-Equipped Devices," *Sensors 2020*, March 2020.

[15] D. W. O'Hagan, S. R. Doughty, and M. R. Inggs, *Academic Press Library in Signal Processing: Array, Radar and Communications Engineering*, R. Chellappa and S. Theodoridis, Eds. Elsevier, 2018, vol. 7.

[16] M. I. Skolnik, "An Analysis of Bistatic Radar," *IRE Transactions on Aerospace and Navigational Electronics*, vol. 8, pp. 19–27, March 1961.

[17] 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.

[18] M. Q. Nguyen, R. Feger, J. Bechter, M. Pichler-Scheder, M. H. Hahn, and A. Stelzer, "Fast-Chirp FDMA MIMO Radar System Using Range-Division Multiple-Access and Doppler-Division Multiple-Access," *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 1, pp. 1136 – 1148, December 2020.

[19] A. Zwanetski and H. Rohling, "Continuous wave MIMO radar based on time division multiplexing," in *2012 13th International Radar Symposium*, May 2012, pp. 119–121.

[20] 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.

[21] 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.

[22] 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.

[23] Y. Bar-Shalom, F. Daum, and J. Huang, "The Probabilistic Data Association Filter: Estimation in the Presence of Measurement Origin Uncertainty," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[24] 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.

[25] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, “The Labeled Multi-Bernoulli Filter,” *Trans. Signal Processing*, vol. 62, no. 12, pp. 3246–3260, February 2014.

[26] T. Kropfreiter, F. Meyer, and F. Hlawatsch, “A Fast Labeled Multi-Bernoulli Filter Using Belief Propagation,” *Transactions on Aerospace and Electronic Systems*, vol. 56, pp. 2478–2488, September 2019.

[27] A.-A. Saucan, M. J. Coates, and M. Rabbat, “A Multisensor Multi-Bernoulli Filter,” *Transactions on Signal Processing*, vol. 65, no. 20, pp. 5495 – 5509, July 2017.

[28] S. Miller, “Object Tracking in Multistatic Millimeter Wave Radar Networks,” Master’s thesis, Massachusetts Institute of Technology, May 2020.

[29] A. A. Kadhim and A. Alubaidy, “Throughput improvement for wireless networks using MIMO network coding,” *First National Conference for Engineering Sciences*, February 2012.

[30] 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.

[31] 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.

[32] A. A. Saucan and M. Z. Win, “Information-seeking sensor selection for ocean-of-things,” *IEEE Internet of Things J.*, vol. 7, no. 10, pp. 10 072–10 088, Oct. 2020, special issue on *Internet of Things for Smart Ocean.*

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

[34] 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.

[35] 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.

[36] 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.

[37] 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 513 873, 2008, special issue on *Cooperative Localization in Wireless Ad Hoc and Sensor Networks.*

[38] M. Weib, "Synchronization of Bistatic Radar Systems," *International Geoscience and Remote Sensing Symposium*, pp. 1750–1753, Sep. 2004.

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

[40] M. Chiani and A. Elzanaty, "On the LoRa Modulation for IoT: Waveform Properties and Spectral Analysis," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 8463–8470, 2019.

[41] S. Lutz, D. Ellenrieder, T. Walter, and R. Weigel, "On fast chirp modulations and compressed sensing for automotive radar applications," *Proc. 15th Int. Radar Symp. (IRS)*, pp. 1–6, June 2014.

[42] C. Aydogdu, M. F. Keskin, G. K. Carvajal, O. Eriksson, H. Hellsten, H. Herbertsson, E. Nilsson, M. Rydstrom, and H. Wymeersch, "Radar Interference Mitigation for Automated Driving: Exploring Proactive Strategies," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 72 – 84, July 2020.

[43] D. E. Barrick, *FM/CW Radar Signals and Digital Processing*, ser. FM/CW Radar Signals and Digital Processing. Environmental Research Laboratories, 1973, vol. 55.

[44] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.

[45] M. Mbeutcha and V. Krozer, "CDMA-based MIMO FMCW Radar System Performance using Intra-Pulse Phase Modulation," *2019 16th European Radar Conference (EuRAD)*, pp. 233–236, 2019.

[46] S. W. Golomb and G. Gong, *Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar.* Cambridge University Press, 2005.

[47] L. Zhang and Y. Sun, "The Optimal Assignment of Orthogonal Polyphase Sequences in CDMA Systems," *IEEE COMMUNICATIONS LETTERS*, vol. 22, no. 1, pp. 109–112, January 2018.

[48] Q. He, R. S. Blum, H. Godrich, and A. M. Haimovich, "Target Velocity Estimation and Antenna Placement for MIMO Radar With Widely Separated Antennas," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 79–100, February 2010.

[49] R. Mahler, *Statistical Multisource-Multitarget Information Fusion.* Norwood, MA: Artech House, 2007.

[50] R. Mahler, *Advances in statistical mutisource-multitarget information fusion.* Norwood, MA: Artech House, 2014.

[51] F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[52] J. L. Williams, "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA and association-based MeMBer," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1664–1687, Jul. 2015.

[53] B.-T. Vo and B.-N. Vo, “Labeled Random Finite Sets and Multi-Object Conjugate Priors,” *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3460–3475, Jul. 2013.

[54] Texas Instruments, “AWR1642BOOST and DCA1000 Technical Documents,” July 2018. [Online]. Available: https://www.ti.com/sensors/mmwave-radar/overview.html

[55] 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.

[56] J. Elfring, E. Torta, and R. van de Molengraft, “Particle Filters: A Hands-On Tutorial,” *Sensors (Basel)*, vol. 21, no. 2, p. 438, January 2021.

[57] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, and P. J. Nordlund, “Particle filters for positioning, navigation and tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

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

[59] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.

[60] A. Dürr, B. Schweizer, J. Bechter, and C. Waldschmidt, “Phase Noise Mitigation for Multistatic FMCW Radar Sensor Networks Using Carrier Transmission,” *IEEE Microwave and Wireless Components Letters*, vol. 28, no. 12, pp. 1143–1145, December 2018.

[61] J. Li and P. Stoica, “MIMO radar with colocated antennas,” *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 106–114, Sep. 2007.

[62] J. Liu, C. Gu, Y. Zhang, and J.-F. Mao, "Analysis on a 77 GHz MIMO Radar for Touchless Gesture Sensing," *IEEE Sensors Letters*, vol. 4, no. 5, May 2020.

[63] H. Godrich and A. M. Haimovich, "Localization performance of coherent MIMO radar systems subject to phase synchronization errors," *International Symposium on Communications, Control and Signal Processing (ISCCSP)*, May 2010.

[64] Q. He, YangYang, and R. S.Blum, "MIMO Radar with Widely Separated Antennas—From Concepts to Designs," *Academic Press Library in Signal Processing, Chapter 13*, vol. 2, pp. 667–727, 2014.

[65] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[66] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice.* Springer, 2001.

[67] B. Ristic, M. S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Norwood, MA: Artech House, 2004.

[68] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, Oct. 2005.

[69] U. Madhow, D. Brown, S. Dasgupta, and R. Mudumbai, "Distributed massive MIMO: Algorithms, architectures and concept systems," *Information Theory and Applications Workshop (ITA)*, vol. 1, no. 7, pp. 9–14, February 2014.

[70] E. Olson, "A Passive Solution to the Sensor Synchronization Problem," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1059–1064.

[71] Y. Yang and R. S. Blum, "Phase Synchronization for Coherent MIMO Radar: Algorithms and Their Analysis," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5538 – 5557, November 2011.

[72] H. Rahul, S. Kumar, and D. Katabi, “MegaMIMO: Scaling Wireless Capacity with User Demands,” in *ACM SIGCOMM 2012*, Helsinki, Finland, 2012.

[73] 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, December 2015.

[74] O. Abari, H. Rahul, D. Katabi, and M. Pant, “AirShare: Distributed coherent transmission made seamless,” *IEEE Conference on Computer Communications (INFOCOM)*, pp. 1742–1750, August 2015.

[75] O. Toker and B. Kuhn, “A Python Based Testbed for Real-Time Testing and Visualization using TI’s 77 GHz Automotive Radars,” in *IEEE Vehicular Networking Conference*. Los Angeles, CA, USA: IEEE, December 2019.