SPS-2024-12

M.Sc. Thesis

Distributed ADMM for Target Localization Using Radar Networks

Srikar Chaganti

Abstract

Traditional target tracking using monostatic radar systems typically relies on centralized or decentralized architectures, where all data is transmitted to a fusion center for processing the position and velocity of mobile targets. This approach not only introduces a single point of failure and can lead to increased data transmission times, particularly when the fusion center is distant from individual radar nodes, but it also faces scalability issues and potential bottlenecks when data accumulates at the fusion center. To address these challenges, we introduce a Distributed Alternating Direction Method of Multipliers (DADMM) for target localization within a radar network, wherein each radar node shares its observed data only with immediate neighboring nodes, achieving consensus on the estimated target locations and velocities. Our simulations, which incorporate critical parameters such as the number of radar nodes, radar geometry, and Signal-to-Noise Ratio (SNR), assess their impact on estimation accuracy and convergence speed. The results demonstrate that the proposed DADMM not only effectively eliminates the single point of failure, but also enhances system efficiency and robustness. We also incorporate two distinct stopping criteria for position and velocity estimations, enabling us to promptly fix the first accurately estimated parameter and reallocate computational resources to more effectively refine the remaining parameters, which streamlines computational efforts by focusing on unresolved parameters. We highlight the additional benefits of our proposed framework, and present directions for future work.

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Distributed ADMM for Target Localization Using Radar Networks

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Srikar Chaganti
born in Hyderabad, India

This work was performed in:

Signal Processing Systems (SPS) and Microwave Sensing Signals and Systems (MS3) Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Copyright © 2024 Signal Processing Systems (SPS) and Microwave Sensing Signals and Systems (MS3) Group
All rights reserved.

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Distributed ADMM for Target Localization Using Radar Networks"** by **Srikar Chaganti** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 11/12/2024

Chairman: _____
dr. Raj Thilak Rajan

Advisor: _____
dr. Francesco Fioranelli

Committee Members: _____
dr. Geethu Joseph

_____

Abstract

Traditional target tracking using monostatic radar systems typically relies on centralized or decentralized architectures, where all data is transmitted to a fusion center for processing the position and velocity of mobile targets. This approach not only introduces a single point of failure and can lead to increased data transmission times, particularly when the fusion center is distant from individual radar nodes, but it also faces scalability issues and potential bottlenecks when data accumulates at the fusion center. To address these challenges, we introduce a Distributed Alternating Direction Method of Multipliers (DADMM) for target localization within a radar network, wherein each radar node shares its observed data only with immediate neighboring nodes, achieving consensus on the estimated target locations and velocities. Our simulations, which incorporate critical parameters such as the number of radar nodes, radar geometry, and Signal-to-Noise Ratio (SNR), assess their impact on estimation accuracy and convergence speed. The results demonstrate that the proposed DADMM not only effectively eliminates the single point of failure, but also enhances system efficiency and robustness. We also incorporate two distinct stopping criteria for position and velocity estimations, enabling us to promptly fix the first accurately estimated parameter and reallocate computational resources to more effectively refine the remaining parameters, which streamlines computational efforts by focusing on unresolved parameters. We highlight the additional benefits of our proposed framework, and present directions for future work.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr. Raj Thilak Rajan and Dr. Francesco Fioranelli, for their unwavering support and guidance throughout my thesis journey. At the outset, I had little understanding of distributed optimization, let alone integrating it with the radar field. Their mentorship was invaluable, offering clarity and direction at every step. Whether it was in-person discussions or online meetings, they were always approachable and patient, providing constructive feedback that proved crucial, especially during challenging moments—like my initial misstep of setting the convergence tolerance to '1e-12'. Their insights and encouragement made a profound difference in my academic progress.

I am also immensely grateful to my friends from MS3 and SPS, who stood by me through thick and thin. Their timely feedback and unwavering support, not only during my thesis but throughout my Master's time at TU Delft, have been a source of strength and camaraderie.

A heartfelt thank you goes to my parents, whose constant support and belief in me helped me navigate moments of self-doubt and gave me the confidence to persevere. To my brother, I owe endless gratitude for his unconditional support, sage advice, and encouragement that have guided me not only through this journey but throughout my life.

Finally, I would like to extend my thanks to everyone - friends, colleagues, and mentors — who have supported me directly or indirectly during my Master's journey at TU Delft. Your kindness and encouragement have made this experience truly memorable.

Thank you all for everything.

Srikar Chaganti
Delft, The Netherlands
11/12/2024

Contents

List of Figures

List of Tables

Acronyms and Nomenclature

List of Abbreviations

ADMM	Alternating Direction Method of Multipliers
CPI	Coherent Processing Interval
CRLB	Cramńer Rao Lower Bound
DADMM	Distributed Alternating Direction Method of Multipliers
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FIM	Fisher Information Matrix
LFM	Linear Frequency Modulation
MLE	Maximum Likelihood Estimation
PRI	Pulse Repetition Interval
PRF	Pulse Repetition Frequency
PSD	Positive Semi Definite
RCRLB	Root Cramńer Rao Lower Bound
RF	Radio Frequency
Rx	Receiver(s)
SNR	Signal to Noise Ratio
Tx	Transmitter(s)
UCA	Uniform Circular Array
ULA	Uniform Linear Array
WSN	Wireless Sensor Networks

Nomenclature

$\forall$	'for all' symbol
$\in$	'to be a member of' symbol
$blkdiag(.)$	Block diagonal matrix
a	Scalar quantity
$\mathbf{a}$	Vector quantity
$\mathbf{A}$	Matrix
$\mathbf{a}^T$	Transpose of vector $\mathbf{a}$
$\hat{\text{a}}$	Estimated value of scalar quantity a
$ln(.)$	Log-likelihood of a function
$\mathcal{N}(\mu, \mathbf{\Sigma})$	Normal distribution with mean μ and covariance $\mathbf{\Sigma}$.
$\mathbf{A}^{-1}$	Inverse of matrix $\mathbf{A}$
∇	Gradient of a function
∇^2	Hessian of a function
$\|\|.\|\|$	L_2 norm
$[\mathbf{I}]_{i,j}$	The entry at position (i, j) of a matrix $\mathbf{I}$
$\frac{\partial(.)}{\partial\theta}$	Partial derivative

1 Introduction

1.1 Background: Target Detection & Estimation Methods in Current Radar Systems

In today's technologically driven world, the detection of targets and the estimation of their key parameters with radar systems stand out as pivotal technologies essential to the functionality and safety of numerous industries and applications. Urban traffic control systems rely for example on these technologies to monitor and manage the flow of vehicles, ensuring efficient traffic patterns and reducing congestion. In the aviation industry, the precise detection and estimation of aircraft are crucial for maintaining orderly skies and safeguarding passenger safety. Similarly, indoor target localization technologies are becoming increasingly vital in security systems and logistics, enabling accurate monitoring within complex environments such as shopping centers, warehouses, and large public venues.

At the core of these sophisticated radar systems is the significant advancement in technology, particularly in enhancing the signal-to-noise ratio (SNR) to detect small targets or targets designed to exhibit a low reflectivity. A higher SNR enables clearer differentiation between relevant targets and noise, leading to more accurate detection and estimation of key parameters. Additionally, viewing the target of interest from various spatial positions with a radar network enhances target localization and identification further, providing a more comprehensive understanding of the target's location and movement within the monitored environment [6].

Modern radar networks enhance target detection and localization capabilities through the use of multiple collaborative nodes, each configured to meet specific operational demands. Common setups include monostatic radars, where the simplicity and cost-effectiveness of co-located transmitters and receivers make them a popular choice [7]. For improved coverage and reduced interference, bistatic [8] and multistatic [9] configurations employ separately located transmitters and receivers, ideal for complex environments. More advanced systems like Multiple Input and Multiple Output (MIMO) [10] utilize numerous transmitting and receiving antennas, either co-located or distributed, to form a high-resolution, robust detection grid, crucial in cluttered settings where maximizing target resolution and detection probability is necessary. Additionally, cognitive radar systems [11], which adjust their parameters in real-time to respond to changing environments and target behaviors, are among the most advanced radar technology, offering unparalleled effectiveness in dynamic and unpredictable scenarios.

As radar systems grow more complex and generate increasing volumes of data, the need for effective and efficient data processing to accurately localize targets becomes paramount. This is particularly important when considering radar networks, where the data exchange needs to be implemented across potentially long distances, and often in conditions of congested and contested electromagnetic spectrum. To manage and utilize this data in radar networks, two primary methodologies are currently employed: centralized [10] and decentralized approaches [4].

With centralized processing, all data collected by the radar network's nodes are transmitted to a single fusion center. This center takes on the crucial role of integrating, processing, and analyzing all incoming data, thereby providing a unified output for target localization. This method ensures comprehensive data handling but concentrates the workload and responsibility in a single location, which could be one of the radar nodes themselves.

Conversely, decentralized processing decentralizes data handling responsibilities, with each node managing its own segment of data. Nodes independently process their data to compute partial localization information or preliminary assessments. These partial results are then transmitted to a fusion center, where they are aggregated to produce the final decision on the target's position and velocity in 2D space. This approach not only reduces the computational load on the fusion center but also decreases the volume of data that needs to be transmitted there, potentially enhancing the overall efficiency of the system.

These methodologies are designed to harness the full potential of the radar data collected, optimizing how it is processed and analyzed to achieve the most accurate localization and velocity estimation of targets. However, open challenges remain, as detailed in the following sections.

1.2 Motivation of this work

Though decentralized methods provide almost the same detection rate at high signal-to-noise ratios (SNRs) as centralized methods [4], both approaches have several drawbacks that can compromise their effectiveness in radar data handling. A primary issue is the potential for a single point of failure [12][4]. In centralized systems, this risk arises from reliance on one fusion center to process all incoming data in their entirety, which could lead to systemic failure if the center encounters problems. Similarly, in decentralized systems, while each node processes data locally, the final aggregation and decision-making still depend on a single fusion center, maintaining the risk and vulnerability of a critical failure point.

Additionally, as the number of radar nodes increases to attempt to enhance target detection and tracking accuracy, both systems struggle with scalability. The increased data volume demands more substantial computational resources

and more sophisticated data management strategies, which can lead to bottlenecks in data processing, especially in centralized systems where all data converge on one single point. Moreover, the physical distance between radar nodes and the fusion center can exacerbate these issues, as transmitting large volumes of data over long distances introduces delays and increases the likelihood of data loss, corruption or interception. These challenges require careful consideration to ensure the robustness and reliability of radar networks in critical applications.

Given the limitations inherent in traditional 2D target estimation methods, the work of this thesis is motivated to enhance these techniques by adopting distributed optimization approaches that have proved applicable and successful in other types of sensor networks such as Wireless Sensor Network (WSN) and Internet of Things (IoT) [13][14]. These areas can serve as references to guide further advancements in our work. These approaches address specific challenges posed by both centralized and decentralized methods, incorporating consensus mechanisms and neighbor communication to solve issues such as the single point of failure, and reduce computational redundancy. By enabling multiple radar nodes to collaboratively adjust parameters through iterative communication, distributed optimization not only decentralizes decision-making, but also significantly reduces the load on each node to estimate target parameters. This leads to increased system resilience and improved efficiency, providing robust and reliable target estimation. More details on different distributed techniques will be presented in Chapter 2.

1.3 Research Goal

The goal of this thesis is to investigate an approach to transition from traditional methods of estimating target position and velocity in 2D space within radar networks, to implementing distributed optimization techniques which can be beneficial in applications such as Air Traffic Control or Urban Traffic Monitoring, where such advancements could significantly improve operational efficiency. This approach aims to eliminate the single point of failure by enabling consensus among all radar nodes, and to reduce the computational load by facilitating data exchanges primarily between neighboring nodes.

The proposed approach begins with developing a simulator to generate range and Doppler measurements of a target. A decentralized framework is then used to estimate the target's parameters of interest. Following this, a distributed framework is implemented, aiming to maintain the same level of accuracy in target estimation as for the decentralized methods. This progression ensures that we effectively integrate distributed optimization techniques, enhancing the precision and efficiency of 2D target localization.

In this thesis, three main novel points are introduced within distributed target estimation methods, moving beyond traditional approaches to address and solve

the challenges within radar networks in the current literature.

- First, we transition from traditional approaches, which often have a single point of failure, to distributed approaches. Primal and dual residual monitors have been specifically designed for convergence within the distributed approach. Furthermore, the relationship between the number of radar nodes each node communicates with and the number of update iterations required to achieve convergence has been explored. This analysis includes an examination of the extent of communication and data processing needed at each node to make accurate estimations.

- Second, a subproblem of the main issue has been addressed by implementing two distinct stopping criteria—one for position and another for velocity estimation. This approach allows the parameter that is estimated first to be fixed, subsequently redirecting computational resources to focus solely on the remaining parameter. By focusing on unresolved parameters, rather than recalculating all variables continuously, this method significantly reduces the computational effort.

- Third, another subproblem identified in the first novelty has been addressed by analyzing the relationship between the penalty terms used for convergence in distributed methods and the signal-to-noise ratio (SNR). It has been demonstrated that selecting appropriate penalty terms can reduce the number of update equations required for convergence, thereby streamlining the process and enhancing the efficiency of the distributed estimation method.

This work is currently undergoing review for the IEEE International Conference on Acoustics, Speech, and Signal Processing 2025 (ICASSP) at the Workshop on Distributed Signal Processing and Machine Learning for Autonomous Systems.

1.4 Thesis outline

In this section, the outline of the whole thesis has been defined as follows.

- Chapter 2 serves as a literature review that explores various radar system configurations and the current state-of-the-art techniques used in 2D target estimation. It reviews traditional methods for determining range and Doppler measurements in radar systems, offering a comprehensive understanding of the underlying principles and technologies. This review also looks at distributed techniques found in the open literature, exploring how they are used and their potential to improve radar data processing and target estimation.

- Building on the comprehensive review in Chapter 2, Chapter 3 describes the design and development of our simulator, which is tailored specifically for this thesis. The simulator uses simplified models based on key assumptions to approximate real-world applications, employing techniques from the literature to reflect the geometrical configurations of radar nodes accurately. This chapter explains the simulator's technical specifications and design decisions,

emphasizing its ability to generate a data model. This model is essential for implementing both decentralized and distributed methodologies to accurately estimate the position and velocity of a target in 2D space.

- Chapter 4 explores two methodologies for estimating a target's position using the data model from our simulator. First, we examine the decentralized approach, currently the standard method in radar target estimation. Next, we introduce the Alternating Direction Method of Multipliers (ADMM), a newer, distributed approach. This chapter clearly outlines how each method processes the simulated data to estimate target positions, preparing for their further application and evaluation in the chapters that follow.

- Chapter 5 presents the simulations conducted and analyzes the results to evaluate the methodologies described in Chapter 4. Initially, we utilize the decentralized approach to validate the statistical correctness of the measurements generated by our simulator and to fine-tune the simulator's parameters. Following this, we transition to the distributed approach using the Alternating Direction Method of Multipliers (ADMM). This section demonstrates how radar nodes, by sharing data with their neighbors, can estimate the position and velocity of a target effectively, thus eliminating the single point of failure that is often a drawback in decentralized systems. The chapter also discusses various techniques designed to foster faster convergence.

- Chapter 6 encompasses the thesis's conclusion, providing a summary of the efforts and outcomes related to the proposed research objectives. Additionally, it outlines potential future avenues for this project, highlighting the limitations encountered during the thesis work and offering related recommendations and suggestions for future research.

2 Literature Survey

The literature review is structured into two main sections. The first section focuses on radar systems and the different methods employed for target estimation, including techniques for extracting range and Doppler measurements. The second section presents distributed techniques, exploring their fundamental concepts and how they can be applied effectively, particularly in addressing challenges like scalability and eliminating single points of failure.

2.1 Review of Current Radar Techniques for 2D Target Estimation

This section begins with an overview of traditional radar systems that are currently in use, followed by an analysis of methodologies, both centralized and decentralized, for estimating targets in 2D space. This analysis will include a detailed examination of these approaches, assessing their scalability and efficiency in operational scenarios.

2.1.1 Radar Networks

For target localization, radar systems like monostatic, bistatic, and multistatic are commonly used. Monostatic radars, where the transmitter and receiver are in the same location and often use the same antenna, are simple and reliable. They are especially effective in situations where having a strong and steady signal is important for accurate detection [4][7][15]. Bistatic radars, which have the transmitter and receiver separated at different physical locations, are valuable in scenarios where it is beneficial to detect objects from different angles or when the radar's physical security is a concern, as they are harder to jam or detect, at least in their receiver-only unit [8][16]. Multistatic radars, involving one transmitter and multiple receivers, excel in extensive area coverage and reducing blind spots, providing superior detection capabilities over larger geographical areas [9][6][17].

Given the scope of this thesis, the focus will remain strictly on radar networks made of distributed, independent monostatic radar systems for several reasons. Firstly, the monostatic configuration minimizes path loss by keeping the signal travel distance short, which supports accuracy in 2D target localization [7][15]. This arrangement also allows for a more predictable performance, essential for developing reliable localization algorithms. Additionally, the scalability of monostatic radars, when arranged in an array, enhances detection and localization accuracy without the logistical complexity, transmitter-receiver synchronization requirements, and higher costs associated with managing the more dispersed setups

of bistatic and multistatic systems. This makes monostatic radars a suitable choice for study, as they provide a practical and efficient framework for in-depth study and application in target localization, particularly within controlled environments or specific operational contexts where scalability is also a consideration. A basic block diagram illustrating how a monostatic radar is structured is shown in Figure 2.1 The configuration of the monostatic radar when tracking a single target is illustrated in Figure 2.2. In this diagram, 'R' represents the range of the target from the radar, and 'v' denotes the velocity of the target.

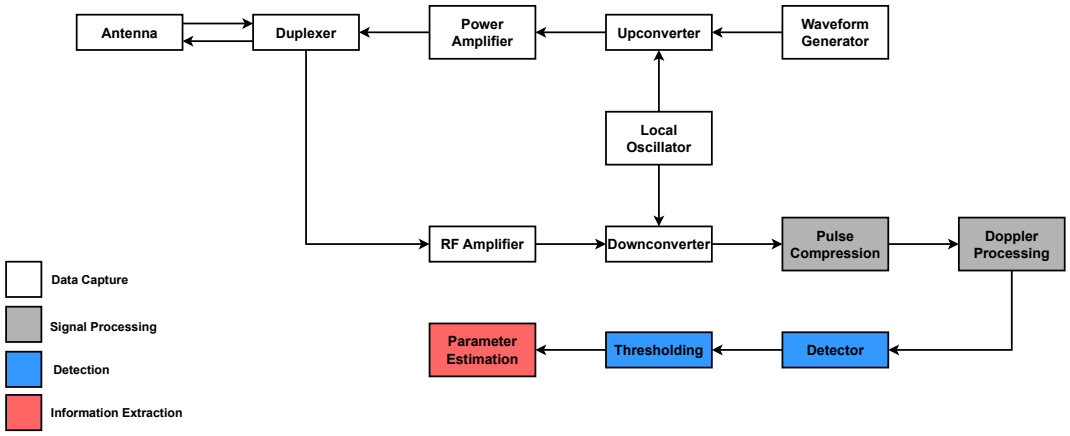

Figure 2.1: Block diagram showing generic monostatic radar structure [1].

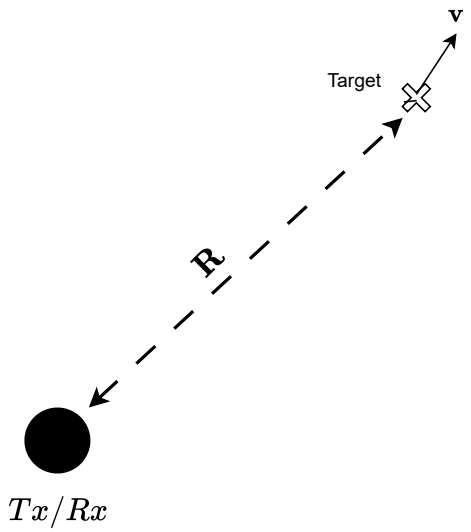

Figure 2.2: Monostatic Radar Configuration [2].

2.1.2 Current Methodologies for 2D Target Estimation

Once we have established our focus on monostatic radar, it is important to note that a single monostatic radar primarily provides raw radar signal data, from

which estimates of target range and Doppler shift frequency can be derived [1][18]. These raw signals are the unprocessed reflections received from targets, containing information about their distance and velocity relative to the radar. To effectively localize a target in 2D space, multiple monostatic radar nodes must be arranged in a given geometric configuration to then use approaches such as multilateration [7][15]. In this section, we will explore the current methodologies prevalent in radar literature for estimating both the position and velocity of targets in 2D space, which are categorized into centralized and decentralized approaches.

2.1.2.1 Centralized Framework

In the centralized approach, each radar node receives raw radar signal data and processes it locally to create range profiles. These range profiles, which represent the distances of targets from the radar based on the travel time of the radar signals, are then sent to the central fusion center [19][10]. By combining these range profiles from all radar nodes at the fusion center, the system can accurately and synchronously estimate the target's position. This centralized processing framework is depicted in the basic block diagram in Figure 2.3.

At the fusion center, range profile from all the nodes is superimposed with others to form a unified profile [10][20]. This profile integrates the data from every node within the radar system. By combining these outputs and positional data, the fusion center can produce a detailed and comprehensive estimation of the target's location, effectively utilizing the processed data from the entire radar network to maximize accuracy and detail in target detection and localization [21][22].

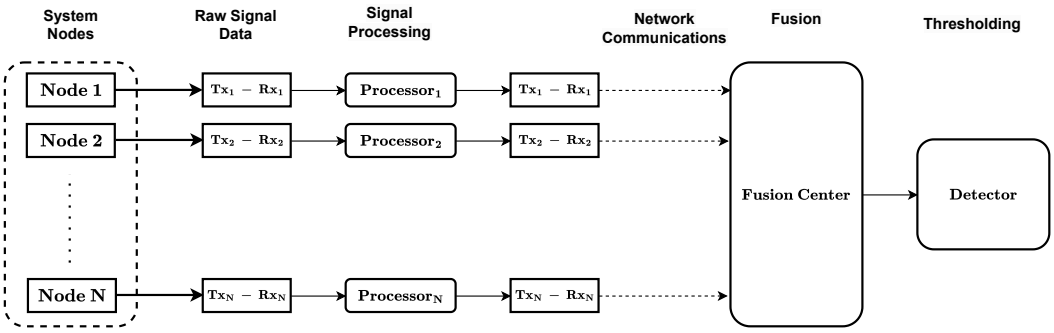

Figure 2.3: Block diagram showing general system flow for centralized approach.

2.1.2.2 Decentralized Framework

The decentralized methodology is an improvement over the centralized approach because it simplifies data handling. In this method, each node in the system independently collects raw radar signal data, and performs signal processing and monostatic thresholding and detection locally [4][12]. The result of this local processing is a set of range estimates and Doppler frequency shifts derived from a

specific Coherent Processing Interval (CPI).

The decentralized detection method capitalizes on its multiple radar nodes to enhance detection capabilities while reducing the volume of data that needs to be transmitted to a fusion center [23][4]. This approach involves data fusion at a high level of abstraction, where data processing and analysis occur locally at each radar node before any information is shared. This method streamlines the system's operation by limiting the data exchange to only essential, processed information, thereby optimizing both the bandwidth usage and the responsiveness of the radar network [4].

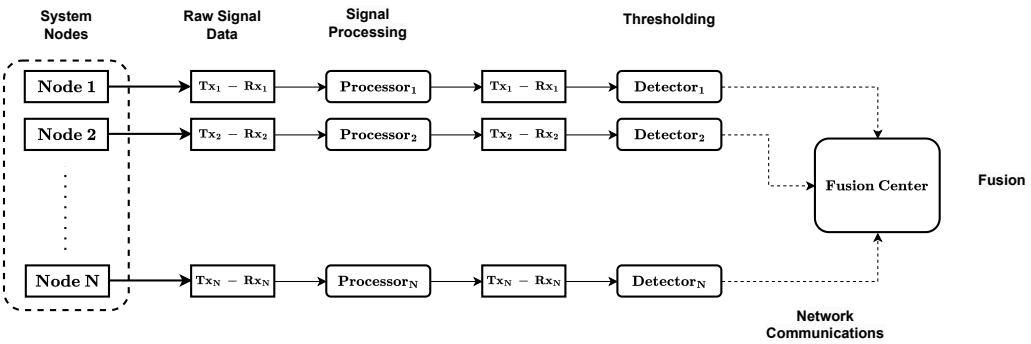

Figure 2.4: Block diagram showing general system flow for decentralized approach.

In monostatic radar systems, as previously discussed, the primary outputs from signal processing, thresholding, and detection are estimates of range and Doppler frequency shift. The methods used to extract these estimates at each radar node will be explored in the following section.

2.1.3 Methods for Extracting Range and Radial Velocity in a Decentralized Approach

Radar systems use several techniques to derive crucial information about target characteristics, such as range and radial velocity, from returned signals. Two prominent methods are widely utilized for estimating these parameters: the Fast Fourier Transform (FFT) and the range rate method. The FFT method is employed to calculate the radial velocity of a target by analyzing in the frequency domain. This technique effectively determines how the frequency of the returned signal has shifted due to the Doppler effect, which is indicative of the target's radial velocity towards or away from the radar. Conversely, the range rate method is used to estimate the radial velocity in the time domain. This approach measures the rate at which the distance between the radar and the target changes over time, providing a direct calculation of the target's velocity along the line of sight but does not yield Doppler shift frequency directly.

2.1.3.1 FFT method

Within a single Coherent Processing Interval (CPI), the radar system organizes the received echo signals into a two-dimensional matrix. Each row of this matrix corresponds to one of the L signals transmitted during the CPI, while each column represents a sample within an individual pulse. The process of applying a Fast Fourier Transform (FFT), which is depicted as a block in Figure 2.5, is critical for analyzing the signal in both the range and Doppler domains. The matrix is structured along two axes: the fast-time axis, which captures the time delay of the echo within each pulse and is pivotal for range estimation, and the slow-time axis, which represents the temporal evolution across the L signals. The slow-time axis encodes phase shifts caused by the relative motion of the target, providing critical information for Doppler frequency estimation. This matrix framework underpins the processing of range and velocity data in radar systems.

To calculate the range of targets, the radar system processes the signals it captures. For Frequency Modulated Continuous Wave (FMCW) radar, a Fast Fourier Transform (FFT) is applied along the fast-time axis, resulting in range profiles. In contrast, for pulse radar systems, matched filtering is employed to extract the matched outputs from the received signals, enhancing the detection of range to the targets. This process in FFT for FMCW, transforms the time-domain data of each pulse into its frequency-domain counterpart, which correlates directly to the range of the targets due to the time delay of the received echoes [2][24]. By converting these time delays into frequency information, the system achieves a precise determination of target distances from the radar, leveraging the detailed resolution offered by frequency domain analysis. This structured approach in processing and interpreting radar signals is integral to modern radar systems, enabling accurate and efficient target detection.

Simultaneously, to measure the target's velocity, the radar employs Doppler processing. This involves performing an FFT along the slow time axis, which corresponds to the time interval across the L signals. The Doppler processing analyzes the frequency shift caused by the target's motion relative to the radar. Motion towards the radar increases the frequency of the reflected signal (positive Doppler shift), while motion away from the radar decreases it (negative Doppler shift). This frequency shift provides accurate measurements of the target's velocity along the line of sight [2][24].

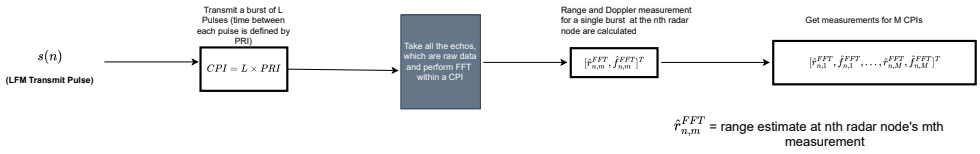

Figure 2.5: Example of estimating range and Doppler measurements from received signals using FFT method [1].

The Range-Doppler map provides a representation of the target's range and Doppler frequency, displaying the power density at each combination of range and Doppler frequency. Assuming a point target, the point with the highest power density on the map corresponds to the target's precise range and Doppler frequency. This accuracy allows the radar to effectively estimate the target's position and velocity. Consequently, the radar can distinguish between stationary and moving objects and track the speed and direction of multiple targets within its field of view [2][1].

2.1.3.2 Range Rate Method

In radar systems that employ pulse signals, the range rate method is used to calculate the distance to a target. This is done by measuring the time it takes for a radar pulse to travel to the target and return as an echo. To enhance the radar's resolution and sensitivity, pulse compression is applied through matched filtering. This technique maximizes the signal-to-noise ratio (SNR) of the received signal and compresses the pulse width of the echo, enabling more precise timing of the echo's arrival, which is crucial for accurately determining the range [1][24]. The formula used to estimate the range is

$$R = \frac{ct}{2} \tag{2.1}$$

where c is the speed of light, approximately $3 \times 10^8 m/s$, and t represents the round-trip time of the radar signal. The division by 2 accounts for the distance the signal travels to the target and then back to the radar.

To calculate the radial velocity of a target directly in the time domain, the range rate is utilized. This measures how quickly the distance between the radar and the target changes over time, providing direct insights into the target's motion toward or away from the radar. [1][24]. The formula to determine the radial velocity in the time domain directly is given by

$$\mathrm{v_r} = \frac{\Delta R}{\Delta t} \tag{2.2}$$

where, $\mathrm{v_r}$ represents the radial velocity of the target relative to the radar. Radial velocity refers to the component of the target's overall velocity that is

directed along the line of sight connecting the radar and the target.

In addition to the FFT method and Range Rate method, other methods for estimating range and radial velocity include the Vector Phase Change and Scalar Phase Change techniques, as discussed in [24].

2.1.4 Overview and research gap in current methods

Based on our literature review of current methodologies for estimating a target's position and velocity in 2D using multiple monostatic radar systems in a network, we have identified two prevailing approaches. The first is a centralized approach where all preprocessed time series data from the radar nodes are sent in their entirety to a fusion center for analysis. The second approach, which offers improvements over the first, involves each radar node locally estimating range and Doppler shift before sending these data to the fusion center for final position and velocity estimation. This method reduces the computational burden on the fusion center by limiting the data sent to essential range and Doppler frequency shift estimates.

However, both approaches rely on a centralized fusion center to aggregate and process data, which introduces potential issues such as a single point of failure and increased power requirements for data transmission, especially from nodes that are located farther away from the center. Additionally, a central processing hub may become a bottleneck as the number of radar nodes increases, potentially compromising system scalability and efficiency. To address these challenges and mitigate the risks associated with a single point of failure and scalability issues, we are exploring distributed techniques that can offer a more robust and scalable solution for radar network operations. The field of distributed optimization has seen significant growth, particularly within Wireless Sensor Networks (WSN) and the Internet of Things (IoT). These areas can serve as foundational reference points for understanding and developing distributed optimization strategies.

2.2 Distributed Optimization

Distributed target tracking is commonly addressed in research, often involving a state space model of the target [25][11]. However, our focus diverges from this norm as we aim to localize the target using a measurement model instead. This method utilizes direct inputs from range and Doppler frequency shift measurements, circumventing the complexities of constructing and updating a state space model. By focusing solely on measurement data, we take a step back from developing a state space model, allowing us to concentrate on directly utilizing the data obtained from measurements.

To understand how we can effectively localize a target in 2D, it is essential to explore the various distributed signal processing techniques currently employed within the distributed systems community and their benefits, especially in terms

of eliminating the risk of a single point of failure—a critical goal of our study. Distributed techniques distribute data processing and decision-making tasks across multiple nodes or systems, enhancing both system resilience and reliability. This decentralized approach not only balances the computational load but also minimizes system vulnerability to localized failures, thereby ensuring operational continuity even if some nodes fail.

2.2.1 Prevalent Distributed Optimization Techniques in Current Research

There are several distributed techniques employed for data processing and decision-making across networks, each with its specific applications and benefits. These include distributed averaging, which involves calculating the average of values stored at each node to reach consensus [26]; gossip algorithms, which allow information to spread through a network as nodes randomly communicate with each other [27]; proximal methods, designed to handle optimization problems by breaking them down into simpler sub-problems that are easier to solve locally [28]; and the method of multipliers, a strategy to solve constrained optimization problems by decomposing them into more manageable parts [29]. Each of these techniques offers different advantages depending on the network's structure, the nature of the data, and the specific goals of the system, such as efficiency, accuracy, or robustness against failures.

Distributed averaging is a widely recognized consensus technique in distributed systems [30][26], but it has its limitations when applied to the task of target localization. This method typically operates directly on 2D estimates of a target's position and velocity. However, in our case, we only have a data model available, which presents unique challenges. Specifically, when position and velocity of a target are estimated at each radar node—by communicating with only a few neighboring nodes within the radar network—averaging these estimates across the entire network can lead to biased results. This bias occurs because each radar node measures the target's radial velocity, which is the component of the target's velocity directed along the line of sight from the radar to the target. Since each node may be positioned differently relative to the target, their measurements of radial velocity can vary. As a result, when these diverse measurements are averaged together, the combined value may not accurately represent the actual velocity of the target. This discrepancy arises because the averaging process does not account for the differences in each node's perspective, potentially distorting the overall estimation of the target's true motion parameters.

Gossip-distributed protocols are communication schemes used in networks where nodes iteratively exchange information with randomly selected peers to achieve data consistency or aggregation across the network [31][32][33]. While these protocols are prized for their simplicity and fault tolerance, they have notable disadvantages. First, they often exhibit slow convergence rates, especially in large networks, due to the inefficiencies in random communication patterns that

require numerous iterations to reach a consensus. Another drawback is the lack of determinism; the stochastic nature of these protocols can lead to unpredictability in performance and outcomes, which is problematic in applications demanding reliable and consistent results. Lastly, gossip protocols are not inherently designed to handle constrained optimization problems efficiently.

Proximal distributed algorithms are a class of optimization methods used for solving large-scale problems that can be decomposed into simpler sub-problems, each solvable in a parallel or distributed manner [28]. However, they exhibit some drawbacks. For one, the convergence rate of proximal methods can be relatively slow, especially in scenarios involving non-smooth functions or when precise adjustment of hyperparameters, such as the step size, is necessary [34][35]. This can make them less efficient for rapidly changing or real-time systems, such as radar networks tracking moving targets. Additionally, proximal methods are primarily designed to handle unconstrained or simply constrained optimization problems. When faced with complex constraints, including multiple equality and inequality conditions, their performance and applicability can significantly diminish, requiring more sophisticated approaches to effectively manage and incorporate these constraints within the optimization process [29].

After careful consideration, we have chosen to focus on the Alternating Direction Method of Multipliers (ADMM) for the scope of this thesis [29]. ADMM is a powerful optimization technique that combines the strengths of dual decomposition and augmented Lagrangian methods, making it highly effective in distributed optimization scenarios. This method is particularly advantageous because it breaks complex optimization problems into smaller, more manageable sub-problems that can be solved independently and in parallel across different nodes or agents. Each sub-problem involves its own local constraints and variables, while ADMM coordinates these local solutions to conform to global objectives and constraints through iterative updates of dual variables [29][36]. This approach not only enhances computational efficiency but also significantly improves scalability and robustness in distributed environments [37]. Moreover, ADMM is capable of handling both convex and certain non-convex problems, providing flexibility in addressing a wide range of practical optimization challenges in distributed settings [38].

2.2.2 Application of ADMM in Wireless Sensor Networks

To address the literature gap in distributed optimization within radar networks, we approached the problem by conceptualizing a wireless sensor network (WSN) as analogous to a radar system, with each WSN node functioning similarly to a radar network node. This analogy enabled us to utilize the extensive research and methodologies already developed for WSNs and apply them to radar networks. Our investigation included reviewing several papers that discuss the use of the ADMM in WSNs [29][39]. These studies describe how ADMM facilitates network-wide optimization while accommodating node-specific constraints, providing crucial insights into how similar distributed optimization techniques can

be adapted for radar network operations [40][14][13].

The foundational concepts of ADMM and its application towards achieving global consensus in distributed systems are thoroughly explained in S.Boyd et al. [29]. In the context of wireless sensor networks (WSNs), the papers I. D. Schizas et al. [14] and P. A. Forero et al.[13] delve into the specifics of how ADMM is adapted to handle local constraints within these networks. Further exploration into the efficiency of ADMM, specifically regarding its convergence criteria, is provided by J. He et al. [41]. Additionally, the implementation of ADMM in more complex non-convex scenarios is discussed in depth in Y. Wang et al. [38] and M. Hong et al. [42], highlighting its versatility and robustness in handling a variety of optimization challenges.

After reviewing the current literature on radar and distributed optimization, a transition can be made from the existing decentralized approach—where range and Doppler frequency shift measurements are sent to a fusion center for estimating the position and velocity of targets in 2D—to employing distributed optimization techniques. In the distributed approach, range and Doppler frequency shift measurements are transmitted to nearby nodes for local estimations. The goal will be to resolve these measurements using the ADMM, with data sharing limited to neighboring nodes. This method aims to compute the position and velocity of the target in 2D and achieve consensus among all nodes, thereby enhancing network resilience and promoting decentralized decision-making. This distributed optimization approach helps reduce the computational load at each radar node and eliminates the single point of failure by enabling radar nodes to communicate with their nearest neighbors and reach consensus.

Simulator Design for Target Range and Doppler Shift Measurements

3

This chapter details the design and functionality of the simulator developed for this thesis, which focuses on simulating target range and Doppler shift measurements for targets observed by a radar network. The simulator employs a Rectangular Pulse Linear Frequency Modulated (LFM) waveform as the transmit signal. Unlike traditional radar systems that utilize pulse compression techniques to enhance resolution and signal-to-noise ratio as mentioned in 2.1.3, our approach directly generates range and Doppler measurements by simply adding theoretical noise to the true values. This noise, modeled as a Gaussian process with zero mean and a specified covariance matrix, is added to the true range and Doppler values to simulate the inaccuracies and uncertainties inherent in actual radar measurements. This simple method bypasses complex signal processing stages, allowing us to control and manipulate the data for our simulation needs.

3.1 LFM Pulse Signal Model

Rectangular Pulse Linear Frequency Modulated (LFM) signals are extensively utilized in modern radar systems for their superior range resolution and robustness to Doppler effects [1][43]. These signals maintain a constant amplitude while the frequency linearly increases or decreases across the pulse duration, making them ideal for complex radar applications. This simplicity in implementation and the ability to process these signals for enhanced target differentiation and tracking in cluttered environments are key reasons for their widespread use [44][1]. Incorporating frequency modulation within each pulse maximizes the utility of the rectangular pulse waveform, defining it as an LFM pulse—a preferred waveform in pulse Doppler radar systems for its signal clarity and interference mitigation capabilities.

The real-valued representation of an LFM pulse waveform at RF, as used in a pulse-Doppler radar, with an instantaneous frequency that increases throughout the pulse, is given by [1]

$$s(t) = A\cos\left(2\pi\left(f_t + \frac{B}{2T}t\right)t + \delta\right) \tag{3.1}$$

where A represents the amplitude of the transmitted signal, which dictates the signal's strength. The term f_t is the transmit frequency, setting the baseline/starting frequency of the waveform. B denotes the bandwidth, which defines the

frequency range over which the signal sweeps, while T is the pulse duration, indicating the length of time over which this frequency sweep occurs. Lastly, δ is the initial phase, representing the phase offset at the beginning of the pulse.

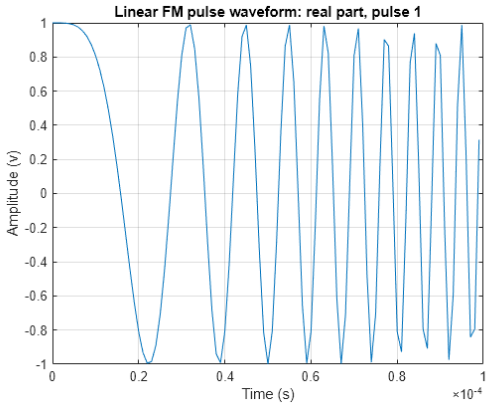

Figure 3.1: Example of an LFM (Linear Frequency Modulation) waveform with a pulse duration of $100\mu s$ [3].

3.2 Designing a Custom Simulator: Integrating Noise into True Measurements

To simplify our simulation and model validation process, we opt to circumvent the detailed sequence of transmitting and processing actual radar pulses as outlined previously in Section 2.1.3. Instead, each radar node in the simulation transmits a series of L pulses at regular Pulse Repetition Intervals (PRI) within the Coherent Processing Interval (CPI). From this process, we extract a single true measurement of range and Doppler shift. We directly utilize these theoretical values for range and Doppler measurements, to which we intentionally introduce a specific amount of noise. This method facilitates the simulation of real-world signal behavior under controlled conditions, allowing us to approximate semi-realistic operational scenarios. This process is applicable at high SNRs, where target estimation uncertainty approaches the theoretical CRLB [45]. Each radar node collects M measurements of range and Doppler shift, meaning the time spent tracking the target is $M \times \text{CPI}$ before sending all measurements to the fusion center for target localization. Figure 3.2 illustrates the design of our custom simulator for obtaining range and Doppler measurements, with the procedure detailed in the following sections.

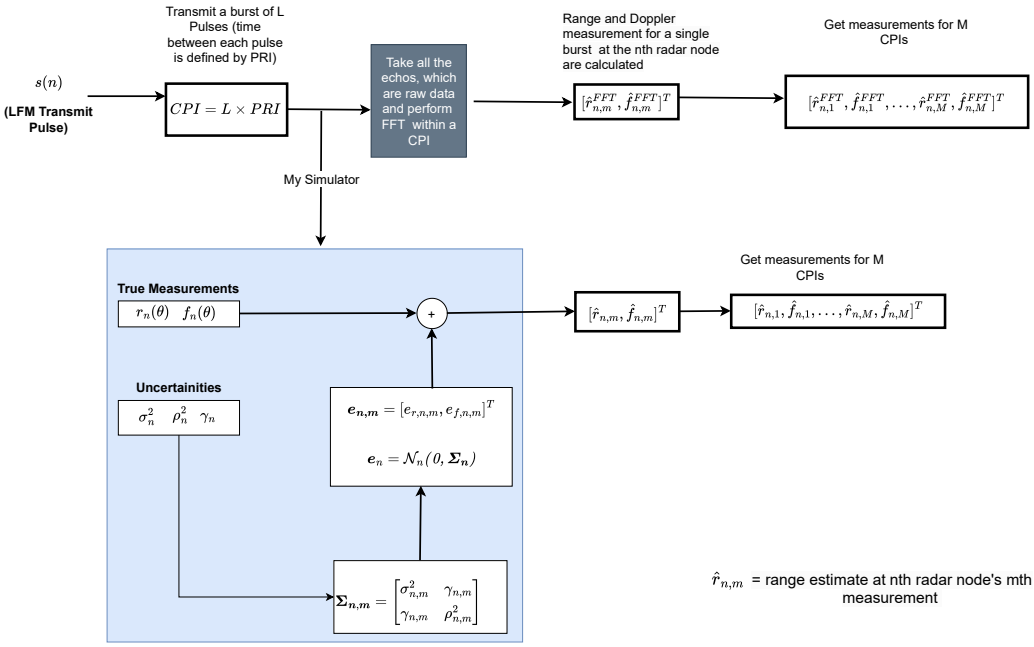

Figure 3.2: Measurement generation process for the nth radar node and mth measurement where noise is added to true values to produce immediate range and Doppler estimates, mimicking semi-realistic measurement outputs.

3.2.1 Uncertainties

When the signal transmitted by the radar and reflected back by targets reaches each receiver, it is a noisy, time-delayed version of the transmitted signal, with additive noise affecting the true signal. The uncertainties in the received signal are assumed to follow the theoretical Cramér-Rao Lower Bound (CRLB), as this represents the ideal conditions for an unbiased estimator, making the measurements as realistic as possible [46]. The CRLB provides a fundamental benchmark for the minimum variance or error that can be achieved in an unbiased estimator, thereby setting a theoretical limit on the precision of our measurements. To simulate a more accurate depiction of actual radar performance under operational conditions, the added noise is designed to adhere to the CRLB.

By incorporating these CRLB-based noise levels, the simulated measurements attempt to mirror the practical limitations and statistical errors encountered in actual radar systems, thereby providing a better framework for evaluating radar performance and developing algorithms that are effective in semi-realistic scenarios [46][45]. An important aspect of using LFM waveforms in radar systems is the phenomenon of range-Doppler coupling, where any uncompensated Doppler shift introduces an apparent range shift in the detected signal. This effect, known as range migration, can complicate the accurate localization of targets, especially

when they are moving at high relative velocities [1][6]. However, in this thesis, we will not consider the impact of range migration as the targets are moving at slow speeds.

3.2.1.1 Range Uncertainty

The uncertainty in radar range measurement primarily stems from the estimation of time delay (t_0), represented mathematically in the context of radar signal processing as

$$s[n; t_0] = s[nT_s - t_0] \tag{3.2}$$

where T_s is the sampling interval in fast time. The variance of the time-delay estimation, $\sigma^2_{t_0}$, is fundamentally linked to the system's noise characteristics and signal properties through the equation [18][1][44]

$$\sigma^2_{t_0} \geq \frac{2E}{N_0} \frac{1}{B^2_{rms}} = \frac{1}{\text{SNR} \cdot B^2_{rms}} \tag{3.3}$$

Here, E denotes the energy of the signal s, N_0 is the noise power spectral density, and B_{rms} is the root mean square (rms) bandwidth of the signal. The term $\frac{2E}{N_0}$ represents the peak signal-to-noise ratio (SNR) at the output of a matched filter. This relationship indicates that the variance in time-delay estimation improves with increases in both SNR and bandwidth. Given that range R is proportional to the time delay t_0 through the relation [2]

$$R = \frac{ct_0}{2} \tag{3.4}$$

where c is the speed of light, the precision in range estimation can also be derived by scaling the time-delay variance. Taking the square root to obtain a standard deviation measure, the precision in range measurement is expressed as

$$\sigma_R \geq \frac{c}{2\sqrt{\text{SNR} \cdot B^2_{rms}}} \tag{3.5}$$

This formulation shows that range precision improves with higher SNR and wider bandwidth. Recalling that range resolution is typically defined as [1]

$$\Delta R = \frac{c}{2B} \tag{3.6}$$

where B is an appropriate measure of bandwidth, this precision can further be related to the inherent resolution limit of the radar system

$$\sigma_R \geq \frac{\Delta R}{\sqrt{\text{SNR}}} \tag{3.7}$$

Connecting this to the context of Linear Frequency Modulated (LFM) waveforms, the range uncertainty in such systems can be further elucidated. Specifically, the CRLB for range measurement under the influence of a known range rate, and

using an LFM waveform with a rectangular envelope, is mathematically expressed as [1]

$$\sigma_n^2 \geq \frac{3c^2}{8\pi^2 B_n^2 \mathrm{SNR}_n} \tag{3.8}$$

where, σ_n^2 represents the lower bound on the variance of the range estimate at the nth radar node, c is the speed of light, B_n is the swept bandwidth of the LFM waveform, and SNR_n is the Signal-to-Noise Ratio at the nth node. The swept bandwidth B_n of nth is a crucial parameter, as it directly impacts the precision of range measurement—the wider the bandwidth, the finer the resolution, and the lower the minimum variance.

The inverse relationship between measurement precision and both SNR and bandwidth is highlighted in Equation 3.8. As SNR increases or bandwidth widens, the lower bound on variance decreases, leading to higher precision in range estimation. This relationship is fundamental, guiding trade-offs between waveform design parameters and achievable accuracy, which is essential for optimizing radar performance across different operational scenarios.

3.2.1.2 Doppler Uncertainty

The CRLB for radar frequency estimation, especially when multiple unknown parameters are involved, such as the initial phase and amplitude of a signal is rigorously derived and widely acknowledged in the literature [46][1][18]:

$$\rho_n^2 \geq \frac{3}{2\pi^2 T^2 L(L^2-1)\mathrm{SNR}_n} \approx \frac{3}{2\pi^2 T^2 L^3 \mathrm{SNR}_n} \tag{3.9}$$

where, T is the slow-time sampling interval (PRI), L denotes the number of measurements, and SNR_n is the signal-to-noise ratio associated with the nth measurement. The CRLB here demonstrates a cubic decrease with respect to L, indicating that the precision of frequency measurements significantly improves as the number of measurements increases [1]. This improvement has several dimensions: the L component boosts the Signal-to-Noise Ratio (SNR) by integrating multiple samples, which consolidates signal power and diminishes noise impact. Furthermore, the squared term (L^2) improves frequency resolution, quadratically because the variance of the frequency estimate decreases with the square of the increase in observation time ($L \times$ PRI), leading to more precise Doppler shift measurements. This dual enhancement of both SNR and resolution underscores the benefit of extensive data collection in radar systems for achieving precise frequency estimates [1].

3.3 Measurement Model

In this section, we discuss the design of the measurement model for each radar node in the network. As mentioned earlier, for simplicity we bypassed the standard FFT

procedure for target position and Doppler estimation, instead directly using the generated measurements.

3.3.1 Target Parameters

Target parameters are key characteristics of a detected target, derived from the echo signals reflected off the target. These parameters include the target's range, bearing, location, radial and cross-range velocity, overall velocity vector, size, shape, symmetry, and material properties. The primary challenge in radar operations is to estimate these parameters accurately, representing the true state of the target based on observed data.

Once a target is detected against a background of noise and clutter, the next step is parameter estimation. The assumption here is that the signal-to-noise ratio (SNR) is sufficiently high and there is no significant clutter, indicating that detection has already been successfully achieved. This allows for the reliable extraction of detailed characteristics from the detected signals [46][45].

In the context of this thesis, we focus on estimating the position and velocity of targets in two-dimensional space, hence four distinct parameters. The specific parameters under consideration are represented as

$$\boldsymbol{\theta} = [x, y, \dot{x}, \dot{y}] \tag{3.10}$$

where x and y denote the target's position along the x and y directions, respectively, and $\dot{x}$ and $\dot{y}$ represent the velocity components along these directions.

3.3.2 True Range Measurements

To compute the true range measurements from each radar node, we employ the Euclidean distance formula to ascertain the direct line-of-sight distance to the target. This calculation assumes an unobstructed path between the radar node and the target, negating any potential occlusion from obstacles or variations in terrain.

The true range measurement ($r_n(\boldsymbol{\theta})$) between the target and the nth radar node is calculated using [6][17]

$$r_n(\boldsymbol{\theta}) = \sqrt{(x_n - x)^2 + (y_n - y)^2} \quad \forall n = 1, 2, \cdots, N \tag{3.11}$$

where $\mathbf{p}_n = [x_n, y_n]^T$ is the position of the nth radar node and $\mathbf{p} = [x, y]^T$ is the position of the target. This approach guarantees that the range measurements accurately represent the spatial relations under optimal conditions.

3.3.3 True Doppler Shift Measurements

The true Doppler shift measurement ($f_n(\boldsymbol{\theta})$) from the nth radar node is determined using [6][17]

$$f_n(\boldsymbol{\theta}) = \frac{\mathbf{v}}{\lambda} \frac{(\mathbf{p}_n - \mathbf{p})}{|\mathbf{p}_n - \mathbf{p}|} \tag{3.12}$$

where $\mathbf{v}$ represents the target's actual velocity, and λ is the wavelength of the radar signal. The term $\frac{(\mathbf{p}_n - \mathbf{p})}{|\mathbf{p}_n - \mathbf{p}|}$ represents the unit vector pointing from the radar node position $\mathbf{p}_n$ to the target position $\mathbf{p}$. This unit vector effectively captures the directional component of the target's movement relative to the radar node. The Doppler shift is crucial as it indicates not only the speed but also the direction of the target relative to the radar, with positive values indicating motion towards the radar and negative values indicating motion away. This Doppler frequency shift directly affects the observed frequency of the radar return and is crucial for tracking and velocity estimation in radar systems.

3.3.4 Measurement Model for nth radar node

For each burst of L pulses, a single set of range and Doppler measurements is obtained, representing the target's position and doppler shift at that moment. In our simulations, we assume the target moves at a constant velocity and remains nearly stationary over the observational time ($M \times CPI$), allowing for M range and Doppler measurements within this short period. If the target moves rapidly, the PRI must be high enough to capture data so that the target does not move much within a CPI. The M range and Doppler shift measurements over a single observational time for $\hat{\boldsymbol{\theta}}$ are presented in Equation 3.13 [6]. Here, $\hat{\boldsymbol{\theta}}$ represents the estimated values of $\boldsymbol{\theta}$, which are the target's parameters, specifically its position and velocity in 2D.

$$\mathbf{z}_n = \begin{bmatrix} \hat{r}_{n,1} \\ \hat{f}_{n,1} \\ \hat{r}_{n,2} \\ \hat{f}_{n,2} \\ \vdots \\ \hat{r}_{n,M} \\ \hat{f}_{n,M} \end{bmatrix} = \begin{bmatrix} r_{n,1}(\boldsymbol{\theta}) \\ f_{n,1}(\boldsymbol{\theta}) \\ r_{n,2}(\boldsymbol{\theta}) \\ f_{n,2}(\boldsymbol{\theta}) \\ \vdots \\ r_{n,M}(\boldsymbol{\theta}) \\ f_{n,M}(\boldsymbol{\theta}) \end{bmatrix} + \begin{bmatrix} e_{r_{n,1}} \\ e_{f_{n,1}} \\ e_{r_{n,2}} \\ e_{f_{n,2}} \\ \vdots \\ e_{r_{n,M}} \\ e_{f_{n,M}} \end{bmatrix} \tag{3.13}$$

where $\hat{r}_{n,m}$ and $\hat{f}_{n,m}$ represent the position and Doppler frequency shift estimates of the target at nth radar node and mth measurement, and $e_{r_{n,m}}$ and $e_{f_{n,m}}$ are range and doppler frequency shift errors of mth measurement at nth radar node. These measurements are dependent on the target's characteristics and the location of the nth radar node. The M measurements combined at the nth radar node in the radar network is represented as [6][17]

$$\mathbf{z}_n = \boldsymbol{\mu_n}(\boldsymbol{\theta}) + \boldsymbol{e_n} \tag{3.14}$$

where $\mathbf{z}_n$ represents the vector of observed measurements at the nth node, encompassing both range and Doppler shifts. The term $\boldsymbol{\mu_n}(\boldsymbol{\theta})$ denotes the true

values of these measurements, dependent on the parameter vector $\boldsymbol{\theta}$, which typically includes the target's movement and positional data relative to the radar. The $\boldsymbol{e_n}$ vector signifies the Gaussian measurement errors, characterized by a zero mean and a covariance structure defined by $\boldsymbol{\Sigma_n}$, which is given as [6][17]

$$\boldsymbol{\Sigma}_n = blkdiag(\boldsymbol{\Sigma}_{n,1}, \boldsymbol{\Sigma}_{n,2}, \ldots, \boldsymbol{\Sigma}_{n,M}) \tag{3.15}$$

where $\boldsymbol{\Sigma_{n,m}}$ for the nth node at mth measurement is given as

$$\boldsymbol{\Sigma_{n,m}} = \begin{bmatrix} \sigma^2_{n,m} & \gamma_{n,m} \\ \gamma_{n,m} & \rho^2_{n,m} \end{bmatrix} \tag{3.16}$$

with $\sigma^2_{n,m}$ and $\rho^2_{n,m}$ from equations 3.8 and 3.9 respectively and the correlation between range and Doppler is given by $\gamma_{n,m}$, where

$$\gamma_n = \eta \sigma_n \rho_n \tag{3.17}$$

The variances indicate the expected variability or noise level within each type of measurement, providing a metric of their reliability. The $\gamma_{n,m}$, on the other hand, reveals the extent to which variations in range measurements are linked to variations in Doppler measurements, offering insights into their interdependencies.

The covariance matrix $\boldsymbol{\Sigma}_{n,m}$ is symmetric and it is Positive Semi-Definite (PSD) [47], ensuring consistent variation between range and Doppler. The PSD nature of the matrix guarantees that the variances are positive and adheres to the Cauchy-Schwarz inequality. This implies that the covariance γ does not surpass the geometric mean of the variances of σ and ρ, mathematically expressed as $\gamma^2 \leq \sigma^2 \times \rho^2$. This inequality ensures that the correlation between range and Doppler does not reach an unrealistically high level. The covariance matrix provides valuable insights into the joint behavior of range and Doppler measurements [47][46].

In a similar way, measurements can be obtained for all nodes in the radar network. Each radar node's measurements depend on its position and the target's characteristics, represented by $\boldsymbol{\theta}$. In Chapter 4, we will explore how the measurement model is used to estimate the target's parameters, denoted as $\boldsymbol{\theta}$.

Methodologies for Parameter Estimation: Decentralized and Distributed Approaches

4

In this chapter, we will explore the methodologies utilized to determine the target parameters, denoted as $\boldsymbol{\theta}$. Our approach is twofold: first, we employ a Decentralized Approach, where all radar nodes transmit their individual range and Doppler shift measurements to a central fusion center. Here, a state-of-the-art estimator is applied to compute the estimated parameters $\hat{\boldsymbol{\theta}}$.

The second methodology involves a Distributed Approach. Unlike the decentralized method, each radar node in this approach communicates only with a number of its immediate neighbors. These nodes collaboratively estimate the target's location and direction of movement, gradually achieving consensus on the target parameters after k iterations, reaching a solution asymptotically. This distributed methodology is innovative to this research and, to the best of the author's knowledge, has not been previously implemented for radar networks in this format. Before delving into the methodologies for estimating $\hat{\boldsymbol{\theta}}$, we introduce the triangulation property, which determines the minimum number of radar nodes required to localize the target.

4.1 Triangulation Property

Since omnidirectional monostatic radars are being used, there is no directional information about the target, making it impossible to estimate its exact location from a single radar node [4]. Additionally, obtaining a reliable velocity estimate from a single monostatic radar is challenging.

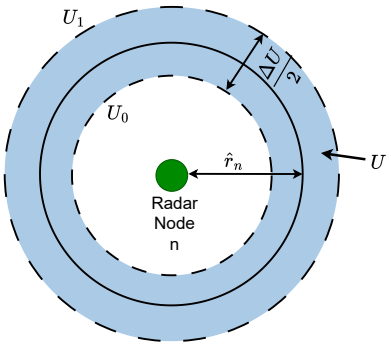

Figure 4.1: Omnidirectional Monostatic Target Range Estimation [4].

From Figure 4.1, we observe that in monostatic radar target estimation, there is an equal probability of the target being located anywhere along a circle centered at radar node n, with a radius equal to $\hat{r}_n$ [4][22]. The measure of target location uncertainty is the area of this gray-shaded region, denoted as U. This uncertainty arises from the noise and other factors described in Chapter 3. When there is more noise, the gray region expands, indicating greater uncertainty; with less noise, the region contracts. The dashed circles, U_0 and U_1 represent the lower and upper bounds. The target's possible location $\mathbf{p}$ is contained within the area between these boundaries, determined by the established uncertainty and range estimates.

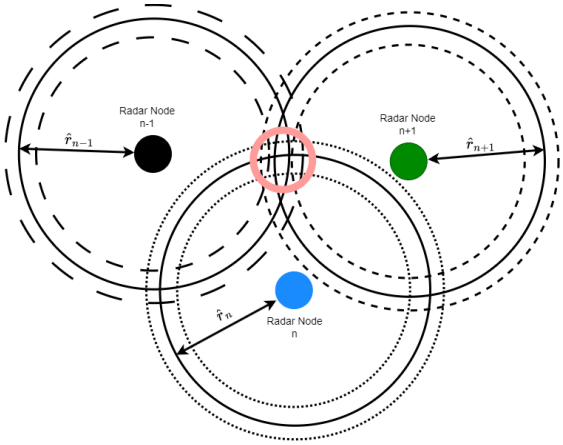

Figure 4.2: Triangulation Property: Requires at least three omnidirectional monostatic radar nodes for precise target location estimation [5].

To estimate the target's position in 2D space, data from at least three radar nodes is necessary, a principle known as the triangulation property [5][48]. Triangulation is fundamental to many geometry-based algorithms used in wireless sensor networks. As shown in Figure 4.2, the target is likely located at the intersection of the range estimates from the three radar nodes (indicated within the red circle). Therefore, data from a minimum of three radar nodes is required to determine the target's position.

4.2 Decentralized Approach

We begin with the Decentralized Approach, wherein each radar node gathers M measurements of target range and Doppler shift during a single observation period as shown in section 3.3.4. These measurements are then transmitted to a central node or fusion center. At this hub, the collected data is used to estimate the overall target's position and velocity within a two-dimensional space. This method is widely adopted in current radar systems literature due to its robustness and effectiveness [4][12].

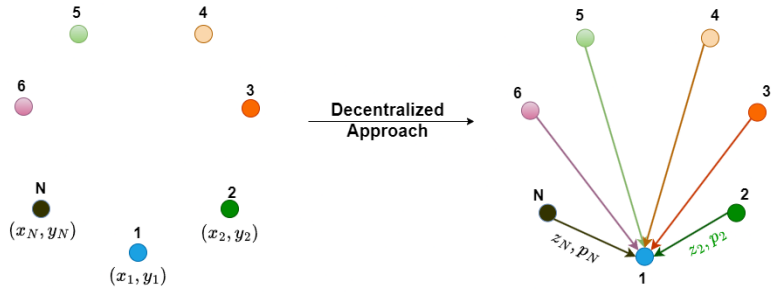

Figure 4.3: Illustration of the Decentralized Approach: Estimated range and Doppler shift measurements from each radar node are transmitted to a central node for the estimation of $\boldsymbol{\theta}$.

Referencing figure 4.3, each radar node transmits its collected measurements of range and Doppler shift, indicated as $\mathbf{z}_n$, along with its position in 2D space, $\mathbf{p}_n$. It is essential to understand that these measurements are influenced by the configuration of the radar network and the relative position and velocity of the target relative to the radar nodes. This relationship is fundamental since radar systems specifically measure range and radial velocity, which are directly affected by the spatial dynamics between the radar node and the target.

4.2.1 Maximum Likelihood Estimator

In Section 3.3.4, the measurement model is characterized as non-convex. A significant challenge with non-convex functions is their tendency to have multiple local minima points where the function value is lower than at all neighboring points. Optimizing such functions can result in becoming trapped in one of these local minima, potentially missing the global minimum that represents the optimal solution [35]. Despite this, the Maximum Likelihood Estimator (MLE) is employed to solve for $\boldsymbol{\theta}$ [46][6][4]. At high SNRs, it is shown that the estimated $\hat{\boldsymbol{\theta}}$ converges to the true value $\boldsymbol{\theta}$ [45][17][6]. MLE involves estimating the parameters of a presumed probability distribution based on observed data. The essence of MLE is to maximize a likelihood function, which adjusts the parameters so that the observed data becomes most probable under the chosen statistical model [46]. Renowned for its intuitive appeal and flexibility, MLE has emerged as a predominant method for statistical inference across various applications. This approach allows for effective and efficient estimation of $\boldsymbol{\theta}$, essential for accurate target parameter determination in our study.

The likelihood function $p[\mathbf{z}; \boldsymbol{\theta}]$ quantifies the probability of observing the parameter vector $\boldsymbol{\theta}$ given the measurement vector $\mathbf{z}$. The log-likelihood function, which is crucial for parameter estimation in statistical models, transforms this likelihood into a log-scale to simplify calculations, particularly when dealing with products of probabilities.

The log-likelihood function for the observed data, given the parameter vector $\boldsymbol{\theta}$, is expressed mathematically as

$$\ln[p(\mathbf{z};\boldsymbol{\theta})] = \frac{1}{2}\sum_{n=1}^{N}\left[\mathbf{z}_n - \boldsymbol{\mu}_n(\boldsymbol{\theta})\right]^T \boldsymbol{\Sigma}_n^{-1}\left[\mathbf{z}_n - \boldsymbol{\mu}_n(\boldsymbol{\theta})\right] + \frac{N}{2}\ln(2\pi) + \frac{1}{2}\ln|\boldsymbol{\Sigma}| \quad (4.1)$$

Here, N represents the total number of radar nodes in the system. The first term represents the sum of the squared deviations of the measurements $\mathbf{z}_n$ from their expected values $\boldsymbol{\mu}_n(\boldsymbol{\theta})$, weighted by the inverse of the covariance matrix $\boldsymbol{\Sigma}_n^{-1}$. The second term adjusts for the scaling effect of the multi-dimensional Gaussian distribution, and the last term accounts for the determinant of the covariance matrix, reflecting the spread and correlation of the data dimensions. This formulation effectively combines all these factors to assess the plausibility of the observed data under the assumed model parameterized by $\boldsymbol{\theta}$.

To determine the parameter vector $\boldsymbol{\theta}$ that maximizes the likelihood of the observed data, we undertake an optimization process. The goal is to find the value of $\boldsymbol{\theta}$ that results in the highest probability of observing the given measurement vector $\mathbf{z}$, under the statistical model specified. This is achieved by maximizing the log-likelihood function. However, because optimization routines commonly focus on minimization, the problem is often reframed as minimizing the negative of the log-likelihood function. The optimization problem can be formulated as follows:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left[-\ln[p(\mathbf{z};\boldsymbol{\theta})] \right] \quad (4.2)$$

Here, $\hat{\boldsymbol{\theta}}$ represents the estimated parameters that minimize the negative log-likelihood, thereby maximizing the log-likelihood itself. This optimization effectively seeks the parameters that make the observed data most probable under the assumed model. The summation aggregates the log-likelihoods computed for each observation, ensuring that the solution reflects the best fit across all data points.

4.2.2 Matlab solver

We employ the MATLAB solver '*fmincon*' to solve the optimization function, which is outlined in Equation 4.2. Fmincon is a Matlab solver [49] for finding the minimum of constrained nonlinear multi-variable function. Here we do not have any inequalities so we set the lower and the upper bound to be unbounded that is -inf and inf respectively. Here, the optimization problem is addressed using the interior point algorithm, an inbuilt feature of the fmincon solver [50]. The interior-point algorithm is often chosen in Matlab's '*fmincon*' for solving large-scale constrained optimization problems due to several advantageous features. Firstly, it is designed to handle large-scale problems efficiently, making it ideal for scenarios with a significant number of variables and constraints. This efficiency extends to its adept handling of sparse matrices, which are common in various optimization scenarios, allowing for more manageable computations as the problem size grows. Additionally, the algorithm is favored for its robust global convergence to global minima in nonlinear convex problems [51] [52]. However, our problem

involves a nonconvex problem where it converges to true values only at high SNRs.

In an unconstrained optimization setting using the interior-point method, the primary focus is on efficiently using Newton's method to approach the minimum of the objective function. Despite its design for constraints, the interior-point algorithm leverages its robust handling of the Hessian and gradient calculations to find a global minimum effectively. The artificial barrier is a technique that adds a penalty to the objective function to deter solutions near the boundaries of feasibility in constrained optimization problems. This penalty increases significantly as a solution approaches the constraint limits, acting like a protective barrier to ensure that the solution remains within acceptable bounds. However, in unconstrained cases, where there are no such limits to consider, this artificial barrier is not necessary, but the iterative, Newton-based approach is maintained. The algorithm outlines the steps to estimate $\hat{\boldsymbol{\theta}}$ using the '*fmincon*' optimization function. Here, $g(\boldsymbol{\theta})$ is the function we aim to minimize, specifically the log-likelihood, $ln[p(\mathbf{z};\boldsymbol{\theta})]$.

4.2.2.1 Fmincon Algorithm Steps

1. **Initial Guess and Setup**: The algorithm starts with an initial guess $\boldsymbol{\theta}^{(0)}$ and typically initializes parameters such as a barrier parameter b (if artificially introducing a soft boundary to stabilize steps early in the process), and a tolerance ϵ for convergence.
2. **Iteration**: At each iteration, the algorithm performs the following steps:
   (a) **Calculate Gradient and Hessian**: Compute the gradient $\nabla g(\boldsymbol{\theta})$ and the Hessian $\nabla^2 g(\boldsymbol{\theta})$ of the objective function at the current point.
   (b) **Search Direction**: Determine a search direction d by solving the Newton system:
   $$\nabla^2 g(\boldsymbol{\theta}) \cdot d = -\nabla g(\boldsymbol{\theta}) \tag{4.3}$$
   (c) **Line Search**: Perform a line search to find an appropriate step size α along the direction d that sufficiently reduces $g(\boldsymbol{\theta})$. We do not explicitly set the step size α in fmincon when using the interior-point method, as MATLAB manages it adaptively via an internal line search strategy. One can influence the behavior of this process through various settings that adjust the optimization's precision and convergence criteria.
3. **Update**: Update the current point:
   $$\boldsymbol{\theta}^{(l+1)} = \boldsymbol{\theta}^{(l)} + \alpha \cdot d \tag{4.4}$$
   where $\boldsymbol{\theta}^{(l)}$ is the value at iteration l and $\boldsymbol{\theta}^{(l+1)}$ is the updated value.
4. **Convergence Check**: Check for convergence using a criterion based on the gradient norm:
   $$\|\nabla g(\boldsymbol{\theta}^{(l+1)})\| < \epsilon \tag{4.5}$$
   If the criterion is met, the algorithm stops; otherwise, it returns to the Step 2.

4.2.3 Cramér Rao Lower Bound

To verify whether the estimates $\hat{\boldsymbol{\theta}}$ are theoretically optimal, given that the measurements are generated with specific SNR and bandwidth conditions, we assess them against the CRLB for each parameter individually. The CRLB serves as a standard for evaluating how well an estimator performs. By measuring an estimator's actual performance against the CRLB, we can see how closely it approaches the ideal estimation process. When an estimator matches the CRLB, it means it's as accurate and consistent as possible, given the specific conditions like SNR and bandwidth. This shows that the estimator is performing at its theoretical best.

The CRLB sets a minimum threshold for the Mean Square Error (MSE) of any unbiased estimator when estimating unknown parameters [53]. For a given parameter of interest, $\boldsymbol{\theta}$, and its unbiased estimate $\hat{\boldsymbol{\theta}}$, the estimate must meet the following condition [46]

$$E_{\boldsymbol{\theta}}\{(\hat{\theta}_i - \theta_i)(\hat{\theta}_j - \theta_j)^T\} \geq [\mathbf{I^{-1}}(\boldsymbol{\theta})]_{i,j} \tag{4.6}$$

In this subsection, we simplify our notation. We denote $\boldsymbol{\theta}$ as $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]^T = [x, y, \dot{x}, \dot{y}]^T$, where $\mathbf{I}(\boldsymbol{\theta})$ represents the Fisher Information Matrix (FIM). The FIM when dealing with a Gaussian random vector is described as follows [46]

$$\left[[\mathbf{I}(\boldsymbol{\theta})]_{i,j}\right]_n = \sum_{i,j} \left[\frac{\partial \boldsymbol{\mu_n}(\boldsymbol{\theta})}{\partial \theta_i}\right]^T \boldsymbol{\Sigma_n^{-1}} \left[\frac{\partial \boldsymbol{\mu_n}(\boldsymbol{\theta})}{\partial \theta_j}\right] \tag{4.7}$$

Since, the covariance matrix $\boldsymbol{\Sigma}$ is structured as a block diagonal matrix, it can be expressed as the sum of individual matrices. And $\frac{\partial \boldsymbol{\mu_n}(\boldsymbol{\theta})}{\partial \theta}$ is a 2×4 Jacobian Matrix ($\mathbf{J}_n(\boldsymbol{\theta})$) for the nth radar node [6][17].

The partial derivatives of the range with respect to each parameter of interest are outlined below. These derivatives are crucial for calculating how changes in each parameter affect the calculated range.

$$\frac{\partial r_n(\boldsymbol{\theta})}{\partial x} = \frac{x - x_n}{\sqrt{(x_n - x)^2 + (y_n - y)^2}} \tag{4.8a}$$

$$\frac{\partial r_n(\boldsymbol{\theta})}{\partial y} = \frac{y - y_n}{\sqrt{(x_n - x)^2 + (y_n - y)^2}} \tag{4.8b}$$

$$\frac{\partial r_n(\boldsymbol{\theta})}{\partial \dot{x}} = 0 \tag{4.8c}$$

$$\frac{\partial r_n(\boldsymbol{\theta})}{\partial \dot{y}} = 0 \tag{4.8d}$$

The partial derivatives of the Doppler shift with respect to all the parameters of interest are outlined below:

$$\frac{\partial f_n(\boldsymbol{\theta})}{\partial x} = -\frac{-\dot{x} \ \times \ r + (\dot{x}(x_n - x) + (y_n - y))\frac{x_n - x}{\sqrt{(x_n-x)^2+(y_n-y)^2}}}{\lambda \ (\sqrt{(x_n - x)^2 + (y_n - y)^2}))^2} \tag{4.9a}$$

$$\frac{\partial f_n(\boldsymbol{\theta})}{\partial y} = -\frac{-\dot{y} \ \times \ r + (\dot{x}(x_n - x) + (y_n - y))\frac{y_n - y}{\sqrt{(x_n-x)^2+(y_n-y)^2}}}{\lambda \ (\sqrt{(x_n - x)^2 + (y_n - y)^2}))^2} \tag{4.9b}$$

$$\frac{\partial f_n(\boldsymbol{\theta})}{\partial \dot{x}} = -\frac{x - x_n}{\lambda \ (\sqrt{(x_n - x)^2 + (y_n - y)^2})} \tag{4.9c}$$

$$\frac{\partial f_n(\boldsymbol{\theta})}{\partial \dot{y}} = -\frac{y - y_n}{\lambda \ (\sqrt{(x_n - x)^2 + (y_n - y)^2})} \tag{4.9d}$$

So, the Jacobian Matrix is given as

$$\mathbf{J}_n(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial r_n(\boldsymbol{\theta})}{\partial x} & \frac{\partial r_n(\boldsymbol{\theta})}{\partial y} & \frac{\partial r_n(\boldsymbol{\theta})}{\partial \dot{x}} & \frac{\partial r_n(\boldsymbol{\theta})}{\partial \dot{y}} \\ \frac{\partial f_n(\boldsymbol{\theta})}{\partial x} & \frac{\partial f_n(\boldsymbol{\theta})}{\partial y} & \frac{\partial f_n(\boldsymbol{\theta})}{\partial \dot{x}} & \frac{\partial f_n(\boldsymbol{\theta})}{\partial \dot{y}} \end{bmatrix} \tag{4.10}$$

$$\text{FIM} = \mathbf{I}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \mathbf{J}_n(\boldsymbol{\theta})^T \boldsymbol{\Sigma}^{-1} \mathbf{J}_n(\boldsymbol{\theta}) \tag{4.11}$$

The FIM is a 4D matrix, the inverse of which is the CRLB estimating $\boldsymbol{\theta}$ [46].

$$\text{CRLB} = \mathbf{I}^{-1}(\boldsymbol{\theta}) \tag{4.12}$$

In Chapter 5, Section 5.1, we explore how changes in specific parameters impact the uncertainties in the estimation of $\hat{\boldsymbol{\theta}}$ and assess their alignment with the theoretical CRLB. We demonstrate that the measurements we generated match the theoretical CRLB, which is the lowest variance bound for an unbiased estimator, confirming that our estimator adheres to this theoretical standard. Later in Chapter 5 Section 5.1, we utilize the Root Cramer-Rao Lower Bound (Root CRLB) to verify if the estimated target parameters ($\hat{\boldsymbol{\theta}}$) match the theoretical RCRLB. This analysis helps confirm that the measurements generated align with the theoretical CRLB, ensuring the accuracy and reliability of our estimation methods.

4.3 Distributed Approach

In the Decentralized approach, there is a risk of a single point of failure since all measurements from individual nodes are sent to a fusion center. Additionally, latency increases due to the potential long distance between nodes and the fusion center, and as the number of nodes grows, the computational burden to estimate $\boldsymbol{\theta}$ on the fusion center escalates.

To address these issues, a distributed approach is introduced, where each radar node communicates with its neighboring nodes, sharing data and estimating

$\boldsymbol{\theta}$ until a consensus is reached as shown in Figure 4.4. In this setup, all radar nodes collaborate to estimate the target's position. The failure of a single node has minimal impact, eliminating the risk of a single point of failure. Latency is reduced since each node only communicates locally with its neighbors, and the computational load on each node is lighter, as the data they process is smaller, allowing for faster estimation calculations.

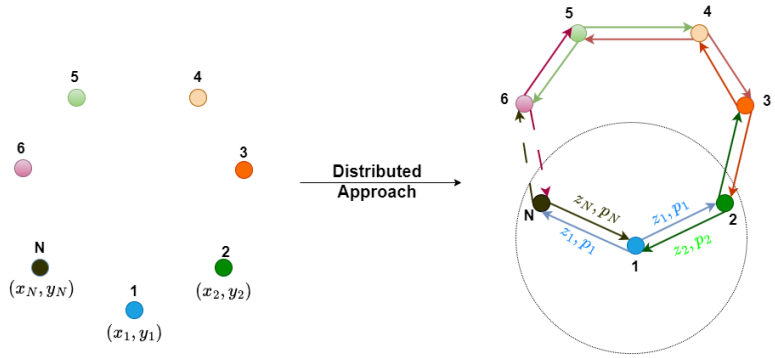

Figure 4.4: Illustration of the Distributed Approach: Each radar node communicates within a specific radius, gathering measurements from neighboring nodes to estimate $\boldsymbol{\theta}$

4.3.1 Network Communication

In the distributed approach, each radar node establishes communication links with its neighbors to share measurements. This data transmission can be visualized using a graph where nodes represent radar nodes and edges represent the communication links between them. The resulting structure is captured in an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, where each entry $A_{nj} = 1$ if $j \in \mathcal{N}_n$ if node j is in the neighborhood of node n i.e., $j \in \mathcal{N}_n$, where $\mathcal{N}_n \subseteq [1, N]$ and $A_{nj} = 0$ if $j \notin \mathcal{N}_n$. Since $j \in \mathcal{N}_n$ if and only if $n \in \mathcal{N}_j$, the adjacent matrix is symmetric i.e., $\boldsymbol{A} = \boldsymbol{A}^T$. To satisfy the triangulation property we need to have cardinality of $\mathcal{N}_n$ greater than 1, i.e. $|\mathcal{N}_n| > 1$.

Various network graphs, such as the Erdős-Rényi model [54] and K-Nearest Neighbors [55], are used for communication. However, in this thesis, we opt for the Unit Disc Graphs (UDG) model to define the communication links between radar nodes [56][48]. The UDG model is chosen because it considers both the position of the nodes and their communication range. It is a simple and widely used graph model in Wireless Sensor Networks (WSNs) because it effectively accounts for node positions and transmission radius [48][40]. In the UDG model, two nodes can communicate if the distance between them is less than a specified transmission radius.

4.3.2 Reformulated Cost Function

Once the neighbors of node n ($\mathcal{N}_n$) are established, the amount of data each node handles is significantly reduced compared to the decentralized approach. The plot

illustrating this is shown in Figure 5.14 in Chapter 5 Section 5.2.2. Each radar node then estimates its own $\hat{\boldsymbol{\theta}}_n$ by exchanging information with its neighbors. The log-likelihood function for the nth radar node is expressed as

$$\ln[p_n(\mathbf{Z}_n;\boldsymbol{\theta}_n)] = \Bigg[[\mathbf{z}_n - \boldsymbol{\mu}_n(\boldsymbol{\theta}_n)]^T \boldsymbol{\Sigma}_n^{-1} [\mathbf{z}_n - \boldsymbol{\mu}_n(\boldsymbol{\theta}_n)] + \sum_{j \in \mathcal{N}_n} [\mathbf{z}_j - \boldsymbol{\mu}_j(\boldsymbol{\theta}_n)]^T \boldsymbol{\Sigma}_j^{-1} [\mathbf{z}_j - \boldsymbol{\mu}_j(\boldsymbol{\theta}_n)] \Bigg] \tag{4.13}$$

Here, each radar node takes into account its own measurement as well as the measurements of its neighboring nodes for the estimation of $\boldsymbol{\theta}_n$ i.e., $\mathbf{Z}_n = \{\mathbf{z}_n\} \cup \{\mathbf{z}_j \mid j \in \mathcal{N}_n\}$.

Since each radar node estimates the target's parameters $\hat{\boldsymbol{\theta}}_n$ by communicating with its neighbors, the estimates will vary across nodes. This is because each radar node observes the target from a different angle, resulting in varying radial velocity and range measurements. While triangulation generally leads to accurate position estimates, velocity estimates can differ significantly between each radar node. Traditionally, a fusion center would be used to reconcile these differences by averaging the $\hat{\boldsymbol{\theta}}_n$ estimates and redistributing the averaged data to all nodes, but this would reintroduce the fusion center, which we aim to avoid.

To address this issue, we introduce an auxiliary variable $\boldsymbol{\vartheta}$, which acts as an edge constraint between nodes. This allows nodes to reach consensus by communicating directly with their neighbors, eliminating the need for a central fusion center. We can think of the auxiliary variable $\boldsymbol{\vartheta}_{nj}$ as a localized "fusion center" between the nth and jth nodes, facilitating consensus through neighbor-to-neighbor communication.

The new cost function at the nth radar node with edge constraints can be formulated as [14][29]

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_n = \arg\min_{\boldsymbol{\theta}_n} & \Big[-\ln[p_n(\mathbf{Z}_n;\boldsymbol{\theta}_n)] \Big] \\ \text{s.t.} \quad & \boldsymbol{\theta}_n = \boldsymbol{\vartheta}_{nj}, \quad \boldsymbol{\theta}_j = \boldsymbol{\vartheta}_{nj} \quad \forall j \in \mathcal{N}_n \end{aligned} \tag{4.14}$$

In this formulation, each radar node n seeks to minimize the negative log-likelihood of its own measurements and its neighbors, subject to the constraint that its parameter estimate $\hat{\boldsymbol{\theta}}_n$ matches the auxiliary variable $\boldsymbol{\vartheta}_{nj}$, which is shared with its neighbor j. Similarly, node j's estimate $\hat{\boldsymbol{\theta}}_j$ must match $\boldsymbol{\vartheta}_{nj}$, ensuring consensus between neighboring nodes without the need for a central fusion center. The radar node communication network is designed to be fully connected and symmetric, ensuring that consensus is achieved among all nodes in the network.

4.3.3 Alternating Direction Method of Multilpliers

While many distributed processing approaches, such as proximal methods or gossip algorithms, can be used in this distributed framework, this thesis employs the Alternating Direction Method of Multipliers (ADMM) to solve the presented estimation problem [29]. This procedure will yield to distributed estimation algorithm where local iterates $\hat{\boldsymbol{\theta}}_n(k)$ will converge to MLE $\hat{\boldsymbol{\theta}}$, where k is the update ADMM iteration number [57][14]. The ADMM is particularly well-suited for distributed optimization due to several key advantages. First, its decomposability allows complex problems to be divided into smaller subproblems, making it ideal for large-scale data or variables typical in distributed environments [13] [58]. Second, ADMM's flexibility in handling various constraints and objectives, including different forms of regularization, makes it applicable to a wide range of optimization problems [29]. Third, it is highly scalable, efficiently addressing the extensive optimization tasks commonly encountered in big data applications, which is crucial for performance in distributed computing systems [14][14]. Additionally, ADMM's robustness to parameter settings ensures that it performs well even with suboptimal parameter choices, reducing the need for extensive tuning and making it practical for diverse applications. However, the effectiveness of ADMM is contingent on the problem structure, particularly benefiting when the problem can naturally be partitioned in a way that aligns with the algorithm's splitting strategy. This combination of decomposability, flexibility, scalability, and robustness makes ADMM a compelling choice for distributed optimization tasks, especially in fields such as radar systems and signal processing [29].

4.3.3.1 Augmented Lagrangian

Augmented Lagrangian is used to formulate subproblems of ADMM. The augmented Lagrangian method for handling equality constraints was originally called the method of multipliers ([36], [59]). Augmented Lagrangian is an extension of the classical Lagrangian method by adding a penalty term to the Lagrangian function. It is a hybrid approach that combines features of both the Lagrangian multiplier method and penalty methods. The Augmented Lagrangian method enforces constraints more strictly due to the penalty component, without requiring an infinitely large multiplier. The Augmented Lagrangian is given by

$$\mathcal{L}_a\left[\boldsymbol{\Theta}, \boldsymbol{\vartheta}, \boldsymbol{\psi}, \boldsymbol{\Phi}\right] = \sum_{n=1}^{N} \left(-\ln\left[p_n\left(\mathbf{Z}_n; \boldsymbol{\theta}_n\right)\right] + \sum_{j \in \mathcal{N}_n} \left(\boldsymbol{\psi}_{nj}^T \left(\boldsymbol{\theta}_n - \boldsymbol{\vartheta}_{nj}\right) + \left\| \frac{\boldsymbol{\Phi}}{2} (\boldsymbol{\theta}_n - \boldsymbol{\vartheta}_{nj}) \right\|_2^2 \right) \right) \tag{4.15}$$

where, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N]$ represents the estimates from all radar nodes, and $\mathbf{Z}_n$ contains measurements of the nth radar node and those of its neighboring nodes i.e., $\mathbf{Z}_n = \{\mathbf{z}_n\} \cup \{\mathbf{z}_j \mid j \in \mathcal{N}_n\}$, while $\boldsymbol{\vartheta} = [\boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2, \ldots, \boldsymbol{\vartheta}_N]$ represents the auxiliary variables, where $\boldsymbol{\vartheta}_n = [\boldsymbol{\vartheta}_{n1}, \boldsymbol{\vartheta}_{n2}, \ldots, \boldsymbol{\vartheta}_{nj}]$ $\forall j \in \mathcal{N}_n$ is the edge constraint between nodes nth radar node and its neighbors $\mathcal{N}_n$. Similarly, $\boldsymbol{\psi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_N]$

represents the Lagrangian multipliers, where $\boldsymbol{\psi}_n = [\boldsymbol{\psi}_{n1}, \boldsymbol{\psi}_{n2}, \ldots, \boldsymbol{\psi}_{nj}]\ \forall n \in \mathcal{N}_n$. The Augmented Lagrangian at nth radar node for equation 4.14 is given by

$$\mathcal{L}_a\left[\boldsymbol{\theta}_n, \boldsymbol{\vartheta}_n, \boldsymbol{\psi}_n, \boldsymbol{\Phi}\right] = \left(-\ln\left[p_n\left(\mathbf{Z}_n; \boldsymbol{\theta}_n\right)\right] + \sum_{j \in \mathcal{N}_n} \left(\boldsymbol{\psi}_{nj}^T \left(\boldsymbol{\theta}_n - \boldsymbol{\vartheta}_{nj}\right) + \left\| \frac{\boldsymbol{\Phi}}{2} (\boldsymbol{\theta}_n - \boldsymbol{\vartheta}_{nj}) \right\|_2^2 \right) \right) \tag{4.16}$$

The Lagrangian multipliers ($\boldsymbol{\psi}$) are used to find the local maxima or minima of the objective function, subject to equality constraints. The matrix $\boldsymbol{\Phi} \in \mathbb{R}^{4\times4}$ is a penalty matrix with diagonal entries corresponding to position (x and y) and velocity ($\dot{x}$ and $\dot{y}$) penalty terms. Position and velocity penalty terms are different because these quantities have varying sensitivities. The penalty terms for position x and y are equal, as are those for velocity $\dot{x}$ and $\dot{y}$.

The objective function's non-convex nature, stemming from the true measurement equations for range and Doppler shift (see Equations 3.11 and 3.12), means that convergence to the global optimum cannot be assured. However, when applying the ADMM to non-convex optimization problems that exhibit a positive semi-definite (PSD) Hessian, the algorithm capitalizes on local convex regions within the objective function. This local convexity, evident from areas where the function curves upward or remains flat, ensures stability in ADMM's iterative updates. By leveraging these locally convex properties, ADMM can navigate more effectively through the non-convex landscape of the optimization problem, avoiding the pitfalls of local minima and making steady progress toward finding an optimal solution [42][29].

Enhanced convergence with non-convex functions is achieved by incorporating higher penalty terms in ADMM [42] [38]. The Augmented Lagrangian framework used in ADMM leverages these penalty terms alongside dual variables to enforce uniformity across distributed variables, ensuring consensus. By imposing large penalty parameters, ADMM aggressively penalizes discrepancies among nodes, fostering uniform convergence and facilitating consensus.

At higher signal-to-noise ratios (SNRs), the low uncertainty ensures that all radar nodes asymptotically converge to the true value as iterations progress [46][6]. The Augmented Lagrangian's penalty terms not only support the enforcement of equality constraints but also enhance convergence speed by smoothing the optimization process and mitigating the risk of ill-conditioning. Further details on selecting penalty terms and their effects on convergence are discussed in Chapter 5, Subsection 5.2.3.

4.3.3.2 Distributed ADMM Update Equations

Here we will look into ADMM update at the nth radar node to reach convergence to the optimal point and reach consensus. ADMM iteratively updates the primal variables $\boldsymbol{\theta}_n$ and $\boldsymbol{\vartheta}_{nj}$ and the dual variable $\boldsymbol{\psi}$ as

- Update $\boldsymbol{\theta}_n$ (Target Estimate at nth radar node)

$$\boldsymbol{\theta}_n(k+1) = \underset{\boldsymbol{\theta}_n}{\operatorname{argmin}} \Bigg[-\ln\left[p_n\left(\mathbf{Z}_n;\boldsymbol{\theta}_n\right)\right] + \sum_{j\in\mathcal{N}_n} \Big(\boldsymbol{\psi}_{nj}^T(k)\left(\boldsymbol{\theta}_n - \boldsymbol{\vartheta}_{nj}(k)\right) + \left\| \frac{\boldsymbol{\Phi}}{2}\left(\boldsymbol{\theta}_n - \boldsymbol{\vartheta}_{nj}(k)\right)\right\|_2^2 \Big) \Bigg] \tag{4.17a}$$

- Update $\boldsymbol{\vartheta}_{nj}$ (Edge Contraint to reach consensus in a distributed way)

$$\boldsymbol{\vartheta}_{nj}(k+1) = \frac{1}{2}\left[\boldsymbol{\Phi}^{-1}\left(\boldsymbol{\psi}_{nj}(k) + \boldsymbol{\psi}_{jn}(k)\right) + \boldsymbol{\theta}_n(k+1) + \boldsymbol{\theta}_j(k+1)\right] \tag{4.17b}$$

- Update $\boldsymbol{\psi}_{nj}$ (Lagrange Multiplier)

$$\boldsymbol{\psi}_{nj}(k) = \boldsymbol{\psi}_{nj}(k-1) + \boldsymbol{\Phi}\left[\boldsymbol{\theta}_n(k) - \boldsymbol{\vartheta}_{nj}(k)\right] \tag{4.17c}$$

4.3.3.3 Convergence Monitoring

To ensure that the updates in the ADMM process are behaving as expected and do not diverge, we monitor the convergence using both primal and dual residuals. These residuals help us assess whether the solution is stabilizing and approaching the optimal point.

Primal residual measures the difference between the updated primal variables and the auxiliary variables, ensuring that the estimates $\boldsymbol{\theta}_n$ and $\boldsymbol{\vartheta}_{nj}$ are converging toward consensus [29]. This approach checks how well the radar nodes are aligning their estimates with their neighbors.

$$r_n(k+1) = \sqrt{\sum_{j\in\mathcal{N}_n} \left\|\boldsymbol{\theta}_n(\boldsymbol{k+1}) - \boldsymbol{\vartheta}_{nj}(\boldsymbol{k+1})\right\|_2^2} \tag{4.18}$$

The dual residual measures the change in the dual variables $\boldsymbol{\psi}_{nj}$ from one iteration to the next, providing insights into how well the constraints are being enforced over time in the ADMM process [29].

$$s_n(k+1) = \sqrt{\sum_{j\in\mathcal{N}_n} \left\|\boldsymbol{\psi}_{nj}(k+1) - \boldsymbol{\psi}_{nj}(k)\right\|_2^2} \tag{4.19}$$

Currently, our distributed approach uses a convergence monitor that tracks both position and velocity estimates together under a single stopping criterion. The algorithm is as follows

Algorithm 1: Distributed ADMM for Localization (nth node)

Input: Initialize $\boldsymbol{\psi}_{nj}(0)$, $\boldsymbol{\theta}_n(0)$ and $\boldsymbol{\vartheta}_{nj}(0)$ randomly $j \in \mathcal{N}_n$; Define $\boldsymbol{\Phi}$ based on SNR. Set $k = 0$;

while *($r_n(k+1) > \epsilon^{pri}$)* **do**
- Update $\boldsymbol{\theta}_n(k+1)$ using (4.17a);
- Transmit $\boldsymbol{\psi}_{nj}(k)$ and $\boldsymbol{\theta}_n(k+1)$ to neighbors $j \in \mathcal{N}_n$;
- Update $\boldsymbol{\vartheta}_{nj}(k+1)$ using (4.17b);
- Transmit $\boldsymbol{\vartheta}_{nj}(k)$ to neighbors $j \in \mathcal{N}_n$;
- Update $\{\boldsymbol{\psi}_{nj}(k)\}_{j\in\mathcal{N}_n}$ using (4.17c);
- Increment k;

end

Output: $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_n$

The simulated results from Algorithm 1 will be presented in Section 5.2.1.

4.3.3.4 Computationally Efficient Distributed Algorithm

We have implemented distinct primal residuals as the stopping criteria for the position and velocity estimation to optimize computational efficiency. Each radar node refines estimates for position and velocity concurrently at the outset. Once either the position or velocity estimates stabilize—typically by iteration k_q—we lock in those values and reallocate computational resources to exclusively refine the remaining estimates (position if velocity has stabilized, and vice versa) from iteration k_q to k. This strategy significantly reduces the computational burden on each radar node by discontinuing the computation of the first stabilized estimates (position or velocity) beyond their point of convergence. The modified cost function is given by

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_n = \arg\min_{\boldsymbol{\theta}_n} & \Big[-\ln[p_n(\mathbf{Z}_n;\boldsymbol{\theta}_n)] \Big] \\ \text{s.t.} \quad & \boldsymbol{\theta}_{n_p} = \boldsymbol{\vartheta}_{nj_p}, \quad \boldsymbol{\theta}_{n_j} = \boldsymbol{\vartheta}_{nj_p} \quad \forall j \in \mathcal{N}_n, \\ & \boldsymbol{\theta}_{n_v} = \boldsymbol{\vartheta}_{nj_v}, \quad \boldsymbol{\theta}_{v_j} = \boldsymbol{\vartheta}_{nj_v} \quad \forall j \in \mathcal{N}_n \end{aligned} \tag{4.20}$$

Here, $\boldsymbol{\theta}_n = [\boldsymbol{\theta}_{n_p}, \boldsymbol{\theta}_{n_v}]^T$ represents the parameter vector, splitting into position and velocity components. We apply the same ADMM update equations as before, but with a modified stopping criterion to verify whether consensus on the position and velocity estimates has been reached. If consensus is achieved for either parameter, we fix that variable as constant and continue solving for the remaining variable. This approach is a practical adaptation typical of engineering solutions. The separate primal residuals for position and velocity are given by

$$r_{n_p}(k+1) = \sqrt{\sum_{j\in\mathcal{N}_n} \left\| \boldsymbol{\theta}_{n_p}(\boldsymbol{k}+\mathbf{1}) - \boldsymbol{\vartheta}_{nj_p}(\boldsymbol{k}+\mathbf{1}) \right\|_2^2} \tag{4.21}$$

$$r_{n_v}(k+1) = \sqrt{\sum_{j \in \mathcal{N}_n} \|\boldsymbol{\theta}_{n_v}(\boldsymbol{k+1}) - \boldsymbol{\vartheta}_{nj_v}(\boldsymbol{k+1})\|_2^2} \tag{4.22}$$

Due to the presence of two primal residuals for position and velocity, Algorithm 1 has been modified to include two separate convergence monitors, one for position and the other for velocity. This modification is presented as Algorithm 2.

Algorithm 2: Computationally Efficient Distributed ADMM for Localization (nth node)

Input: Initialize $\boldsymbol{\psi}_{nj}(0)$, $\boldsymbol{\theta}_n(0)$ and $\boldsymbol{\vartheta}_{nj}(0)$ randomly $j \in \mathcal{N}_n$; Define $\boldsymbol{\Phi}$ based on SNR; Set $k = 0$;

while $(r_{n_p}(k+1) > \epsilon_p^{pri}$ & & $r_{n_v}(k+1) > \epsilon_v^{pri})$ **do**
- Update $\boldsymbol{\theta}_n(k+1)$ using (4.17a);
- Transmit $\boldsymbol{\psi}_{nj}(k)$ and $\boldsymbol{\theta}_n(k+1)$ to neighbors $j \in \mathcal{N}_n$;
- Update $\boldsymbol{\vartheta}_{nj}(k+1)$ using (4.17b);
- Transmit $\boldsymbol{\vartheta}_{nj}(k)$ to neighbors $j \in \mathcal{N}_n$;
- Update $\{\boldsymbol{\psi}_{nj}(k)\}_{j \in \mathcal{N}_n}$ using (4.17c);
- **if** $(r_{n_p} \leq \epsilon_p^{pri})$ **then**
  - $\boldsymbol{\theta}_{n_p}(k) \leftarrow \boldsymbol{\theta}_{n_p}(k+1)$;
  - Set $\boldsymbol{\theta}_{n_p}$ using stored value for further iterations;
- **else if** $(r_{n_v} \leq \epsilon_v^{pri})$ **then**
  - $\boldsymbol{\theta}_{n_v}(k) \leftarrow \boldsymbol{\theta}_{n_v}(k+1)$;
  - Set $\boldsymbol{\theta}_{n_v}$ using stored value for further iterations;
- **end**
- Increment k;

end

Output: $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_n$

The simulated results from Algorithm 2 will be presented later in Chapter 5 Section 5.2.1.1.

In this chapter, it is demonstrated that measurements from at least three radar nodes are required due to the triangulation property essential for localizing the target. The implementation of the decentralized approach, as referenced in radar literature in Chpater 2 Section 2.1.2.2, is explored; it is solved using the state-of-the-art estimator, Maximum Likelihood Estimator, with the MATLAB solver *'fmincon'*. Additionally, the novel topic of this thesis, the distributed approach, is introduced. This approach estimates the position and velocity of the target in 2D space by facilitating communication with the nearest neighbors, which are established using Unit Disc Graphs.

5 Simulations and Results

In this chapter, we conduct several simulations to analyze both decentralized and distributed approaches. Initially, we explore the decentralized approach, examining how the position and velocity parameters are influenced by variables such as the number of radar nodes and the geometries of these nodes. Upon identifying an optimal scenario, we then shift our focus to the distributed approach. Here, we assess the performance of the algorithm we developed and explore strategies to expedite consensus.

5.1 Decentralized Approach

In the decentralized approach, all measurements are collected at a central node where the estimation of target parameters ($\boldsymbol{\theta}$) occurs. This method serves as a baseline for testing various configurations in our experiments, which help define the setup for the validation of the proposed distributed algorithm. These tests include examining the impact of radar node placement, the number of nodes, and target movement on parameter estimation accuracy. Based on the outcomes of these experiments, we then establish a simulation framework for our distributed algorithm.

5.1.1 Influence of Node Geometry on Target Estimation

In this section, we discuss how the arrangement of radar nodes affects the ability to estimate target position and velocity in 2D space. Among various configurations, linear, circular, and semi-circular arrays are commonly used. The Uniform Linear Array (ULA) is straightforward and provides focused detection along its axis but lacks uniform coverage across all directions because its sensitivity decreases for targets that are off-axis [60]. The Uniform Circular Array (UCA), while more complex and larger, offers consistent detection capabilities in all directions, making it ideal for comprehensive 360-degree coverage [60, 61]. The Semi-Circular Array is an adaptable solution that provides wide coverage when full circular deployment isn't feasible, making it suitable for environments with physical or spatial constraints.

We examine the three configurations of radar nodes: circular, linear along the x-axis, and semicircular geometries, with the target moving along the y-axis. The specific simulation parameters used in these configurations are detailed in Table 5.1. These setups and their respective directions are illustrated in Figure 5.1.

Parameter	Value
Radar Nodes (N)	10
Initial Position (m)	(200, 0)
Constant Velocity (m/s)	20
Measurements (M)	32
Number of Pulses (L)	32
Monte Carlo Simulations	1000
Geometry Radius (m)	3000

Table 5.1: Simulation Parameters for Different Geometries and Directions

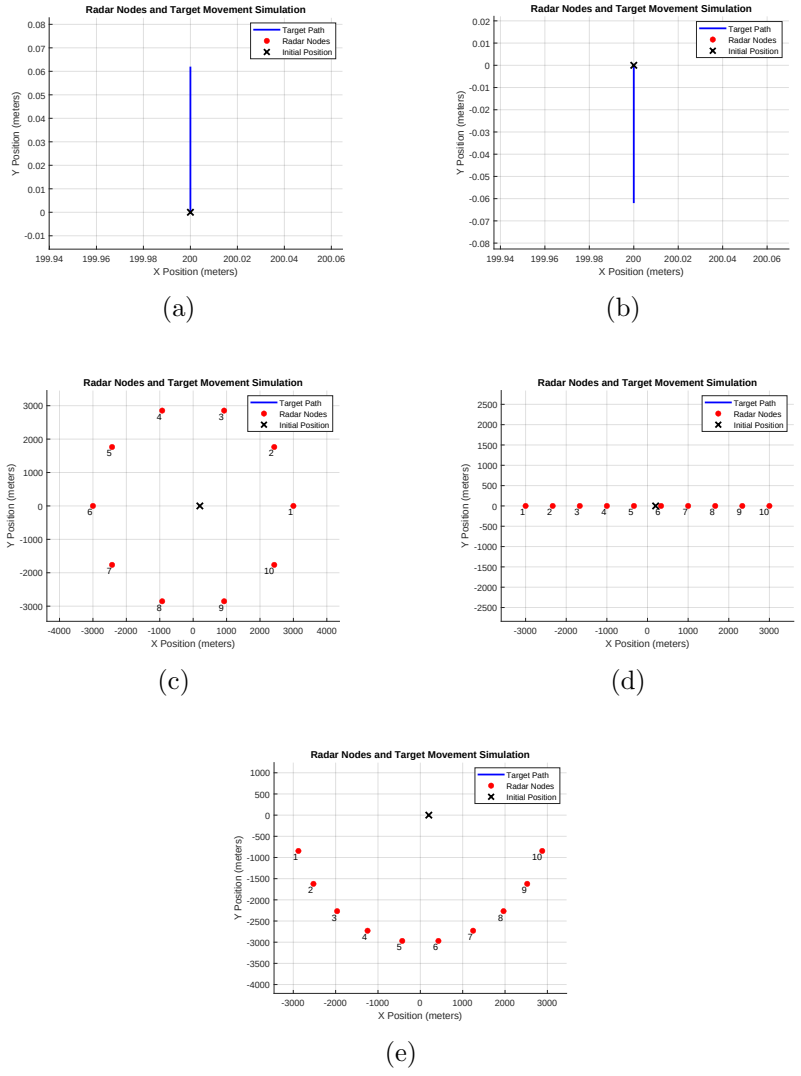

Figure 5.1: Simulation Configurations Across Geometries and Movement Directions: Target movement from x-axis a) 90°; b) 270°. Radar nodes are arranged in c) Circular Geometry; d) Straight Line Geometry; and e) Semi Circular Geometry.

From Figure 5.2, it is apparent that a directional bias is present in the straight-line geometry. This configuration would lead to a consistent high uncertainty in estimating $\hat{y}$, primarily because the straight-line setup of radar nodes does not capture or measure motion perpendicular to the x-axis in an effective manner. When the target moves along the y-axis, this particular motion fails to induce a significant Doppler shift within the straight-line radar configuration. Consequently, since $\hat{y}$ does not affect the distance between the radar and the target, Doppler radar techniques cannot detect this well, resulting in significant uncertainty.

This bias is less pronounced in circular and semicircular geometries. In a circular arrangement, radars are more evenly distributed compared to a semicircular setup, providing broader coverage and deeper insights into target movements. This is particularly advantageous in a radar network with multiple nodes, as the circular geometry enhances detection capabilities across various directions of target motion.

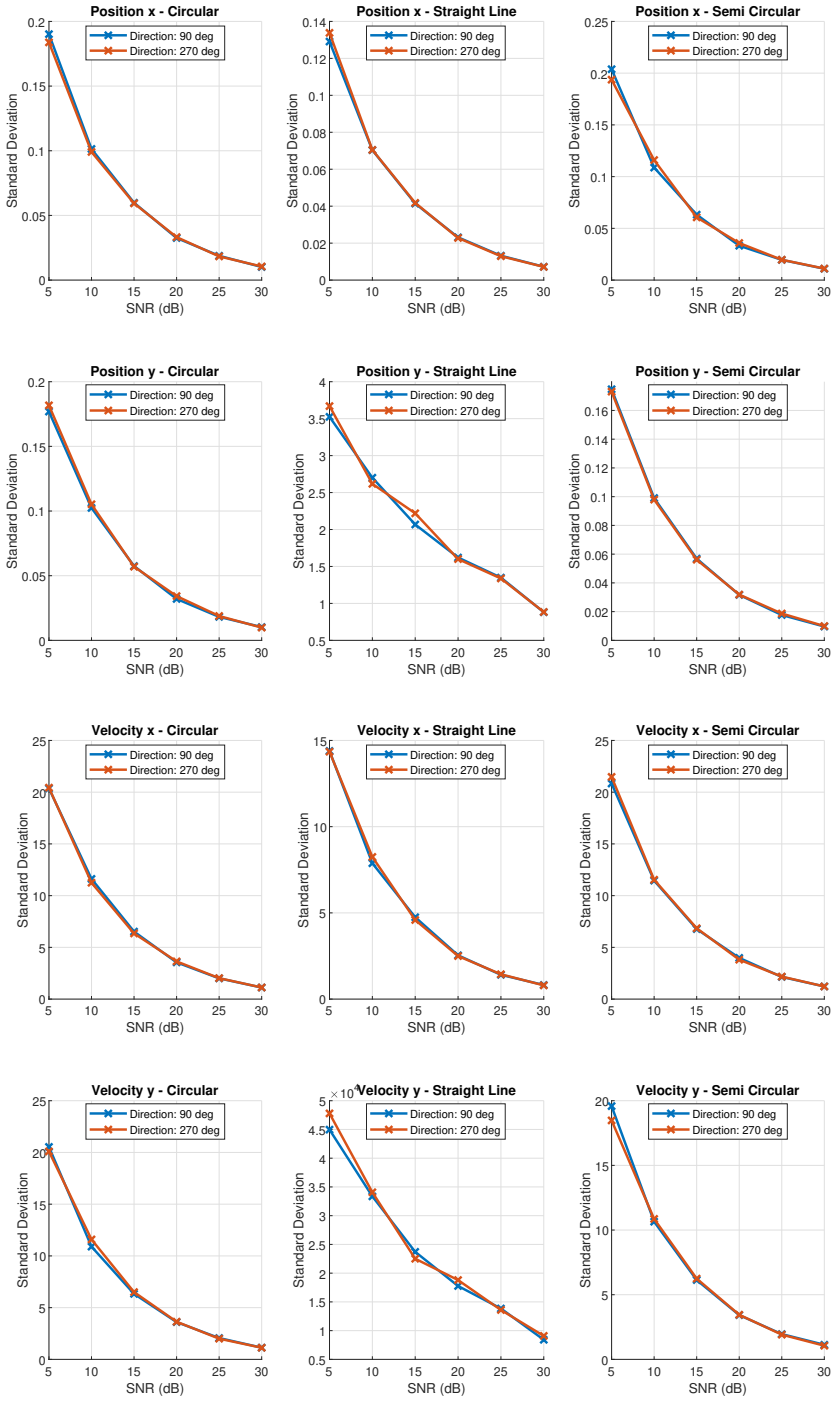

Figure 5.2: Variation in estimation accuracy for target movements at 90° and 270° across circular, straight line, and semicircular geometries, highlighting increased y-velocity error in straight line configurations when targets move perpendicular to nodes aligned along the x-axis.

5.1.2 Node Count Impact on Target Estimation

In this section, after establishing a circular geometry for the radar nodes due to its neutral impact on the direction of target movement, we examine the influence of varying the number of nodes within a Uniform Circular Array (UCA) on the estimation of the target's position and velocity in two-dimensional space. As outlined in Section 4.1, a minimum of three radar nodes is necessary for target localization. We explore how changes in the node count affect the accuracy of position and velocity estimates. Figure 5.3 illustrates the placement of nodes with counts of 5, 10, and 20. The target moves towards a direction of 135° from the x-axis, starting from the coordinate (1000, 1000)m, with the simulation parameters detailed in Table 5.2. This trajectory, as depicted in Figure 5.3d, will be maintained in subsequent simulations unless a change of direction is specifically noted.

Parameter	Value
Geometry	Circular
Initial Position (m)	(1000, 1000)
Constant Velocity (m/s)	20
Measurements (M)	32
Number of Pulses (L)	32
Monte Carlo Simulations	1000
Direction	135° along x-axis
Geometry Radius (m)	3000

Table 5.2: Simulation Parameters for Different Node Count

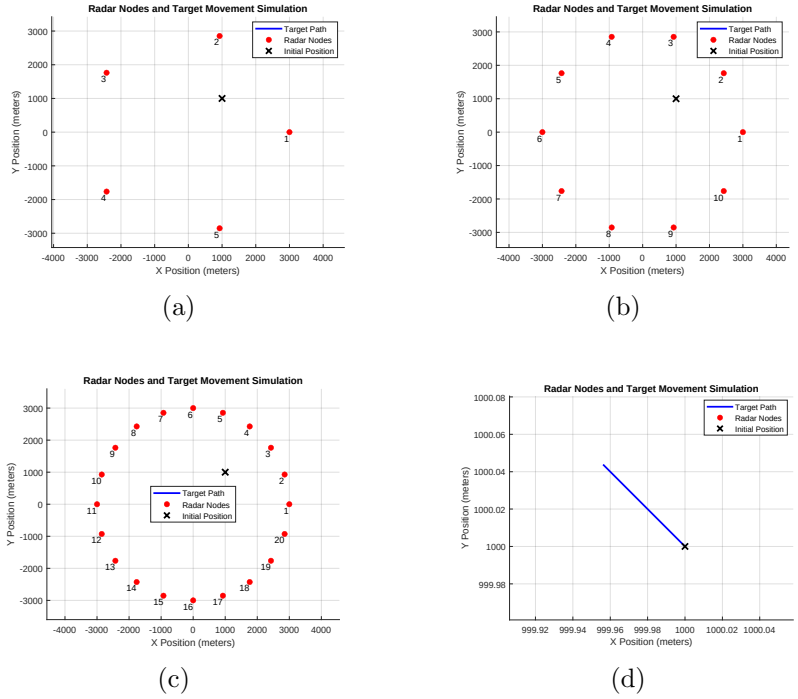

Figure 5.3: Simulation setting for circular geometry with fixed radius and different node count Radar Node Count: a) 5 Radar Nodes; b) 10 Radar Nodes; c) 20 Radar Nodes with d) Target Movement in 135° along x-axis in all simulations.

The augmentation of radar nodes, as shown in Figure 5.4, significantly diminishes the uncertainty associated with estimating $\hat{\boldsymbol{\theta}}$. This increase in nodes not only provides a greater volume of data within the same processing timeframe, but also offers diverse perspectives on the target, thereby enhancing the accuracy of its localization. Furthermore, the Root Cramńer-Rao Lower Bound (RCRLB) for all node counts aligns with the estimated uncertainty of the target's parameters. This observation suggests that, despite its simplicity, the simulator we developed to generate range and Doppler measurements adheres to the theoretical CRLB as shown in Chapter 4 Section 4.2.3, providing a reliable baseline for our evaluations.

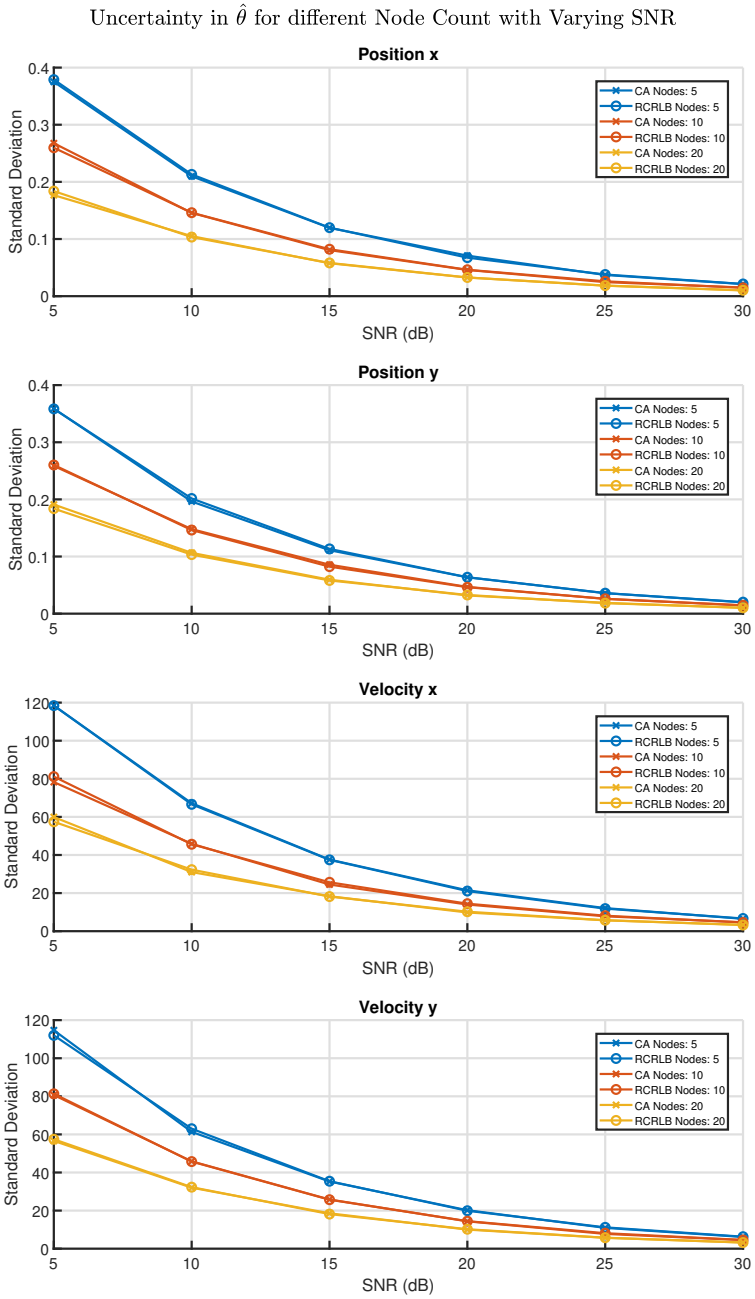

Figure 5.4: $\hat{\boldsymbol{\theta}}$ Uncertainty by Node Count and SNR: Impact on target estimation precision with radar node counts of 5, 10, and 20.

5.1.3 Effect of Transmitted Pulse Count per CPI on Target Estimation

In this analysis, we focus on the number of transmitted pulses, denoted as L, within a single Coherent Processing Interval (CPI). Table 5.3 presents the parameters to run this simulation. Each CPI is processed collectively to yield a singular range and Doppler estimate for each target detected by the radar node. Considering a higher number of pulses (L) with a fixed PRI results in more time spent observing the target, which can improve the accuracy of the velocity estimate. However, this extended observation time also introduces greater uncertainty in the range estimate, as the target may continue to move during this extended interval.

Understanding this trade-off is crucial for optimizing radar performance. Increasing L enhances velocity measurement precision but at the expense of range accuracy. Therefore, this analysis is essential for achieving a balanced approach in obtaining more reliable range and Doppler estimates, enabling more effective radar operation and target tracking.

Parameter	Value
Radar Nodes (N)	10
Initial Position (m)	(1000, 1000)
Constant Velocity (m/s)	20
Measurements (M)	32
Geometry	Circular
Monte Carlo Simulations	1000
Geometry Radius (m)	3000

Table 5.3: Simulation Parameters for Different Pulses sent in a Single CPI

Increasing the number of pulses from 16 to 128 does not significantly alter the position uncertainty for a slow-moving target, as evidenced in Figure 5.5. This observation suggests that for certain target speeds, the benefits of additional pulses in reducing uncertainty may plateau. This is because the target does not move much during the observation period, so adding more pulses does not have much effect on position uncertainty. However, one should notice a significant improvement in velocity uncertainty with more pulses. The duration of observing the target directly impacts how precisely we can estimate its velocity—the longer we observe, the more accurate the velocity estimate becomes. This illustrates a trade-off in radar settings: extending observation times enhances velocity measurements without substantially affecting position accuracy.

However, if the target were moving quickly, increasing the number of pulses could negatively affect range measurements. A fast-moving target travels a significant distance within the longer CPI, which can increase errors in range estimation.

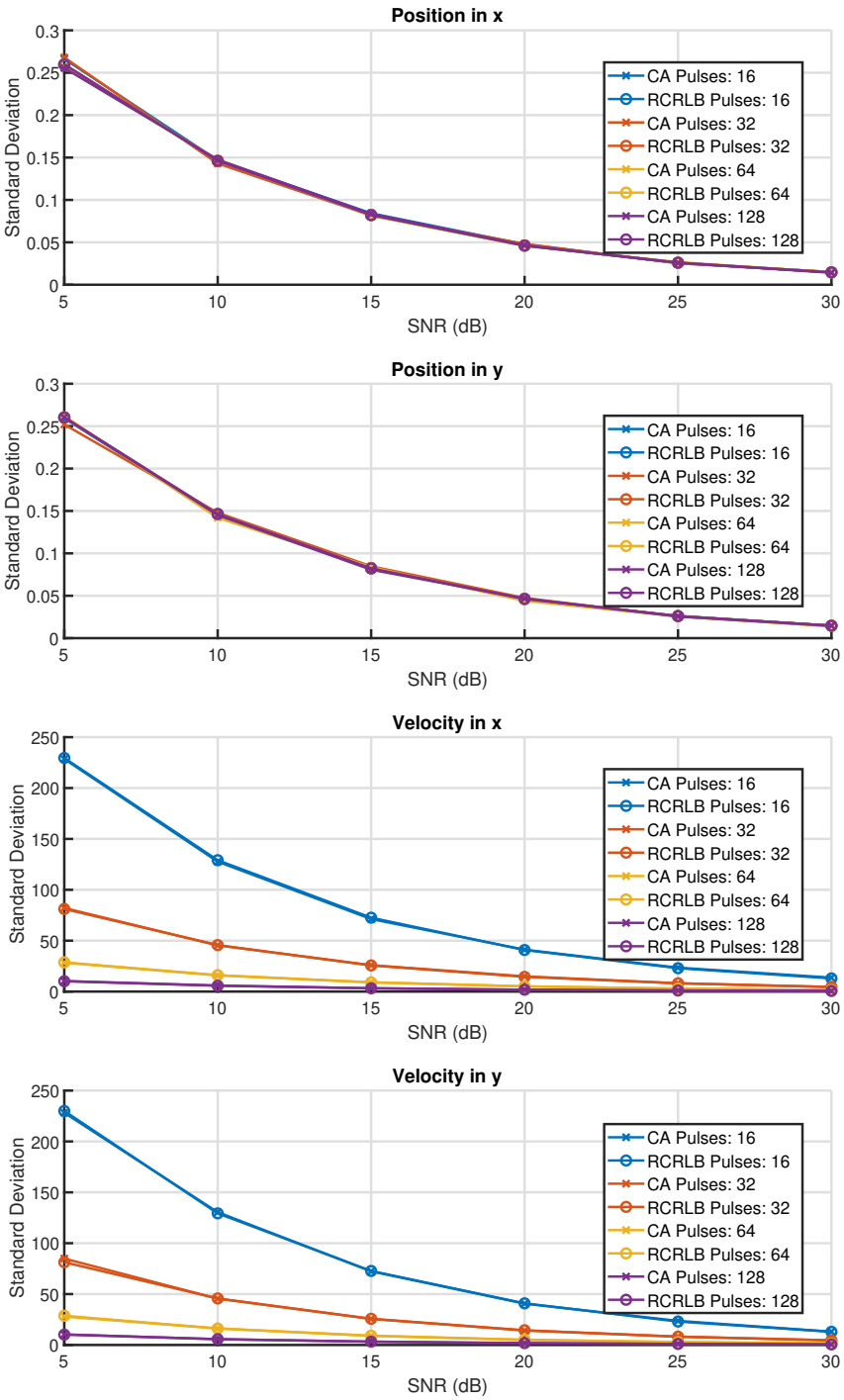

Figure 5.5: Pulse Count Impact on $\hat{\boldsymbol{\theta}}$ Uncertainty: Estimation accuracy variance with 16, 32, 64, and 128 pulses per burst across SNR levels.

5.1.4 Impact of Measurement Count (M) on Target Estimation

In this analysis, we focus on the number of measurements each radar node gathers before transmitting this data to a centralized node to estimate the target's parameters $\boldsymbol{\theta}$. Table 5.4 presents the parameters to run this simulation. The variable M represents the number of Coherent Processing Intervals (CPIs) collected, where each CPI consists of L pulses. From each CPI, we obtain a single range estimate and one Doppler shift estimate. Increasing M effectively means transmitting more data, which can enhance the localization accuracy of the target.

Parameter	Value
Radar Nodes (N)	10
Initial Position (m)	(1000, 1000)
Constant Velocity (m/s)	20
Number of Pulses (L)	32
Geometry	Circular
Monte Carlo Simulations	1000
Geometry Radius (m)	3000

Table 5.4: Simulation Parameters for Measurements Estimated by Each Radar Node Before Transmission to Fusion Center.

The variation in the uncertainty of theta estimation relative to SNR for different values of M, the number of range and Doppler shift frequency measurements collected before transmission to the fusion center, is illustrated in Figure 5.6. We observe that when the target moves slowly, a higher quantity of data generally results in better estimation accuracy. Based on these findings, we initially set M to 32. However, this configuration needs to be reevaluated for fast-moving targets, as accumulating a large number of measurements before transmission can lead to range ambiguity. Despite this, it enhances Doppler velocity estimation because more time is spent collecting data on the target before it is sent to the fusion center.

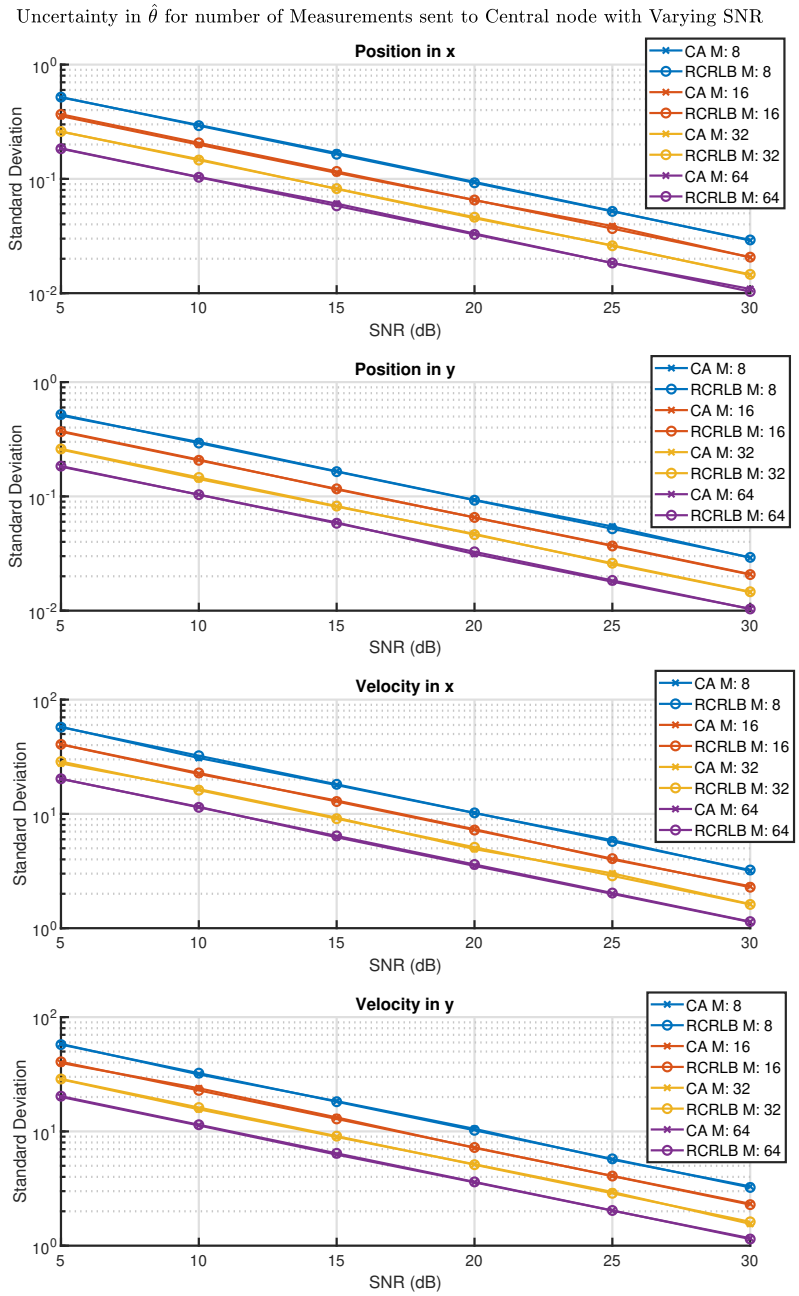

Figure 5.6: Analyzes $\hat{\boldsymbol{\theta}}$ uncertainty with 8, 16, 32, and 64 measurements sent to the central node by each radar node across varying SNR levels.

5.1.5 Effect of Array Radius on Target Estimation

This analysis explores the role of the radar geometry's radius in providing comprehensive views of the target from various angles, enhancing our understanding of the target's behavior within different spatial contexts. Table 5.5 presents the

parameters to run this simulation. The impact of the radius on the uncertainty in target estimation is examined both when the target is positioned outside and inside the circular radar node array. The specific configurations of the radar geometries with varying radii are illustrated in Figure 5.7. Additionally, the movement of the target, oriented at 135° along the x-axis. This setup allows us to assess how changes in the radar array's radius influence detection capabilities and the precision of target estimation under different geometrical radius arrangements.

Parameter	Value
Radar Nodes (N)	10
Initial Position (m)	(1000, 1000)
Constant Velocity (m/s)	20
Measurements (M)	32
Geometry	Circular
Monte Carlo Simulations	1000
Number of Pulses (L)	32

Table 5.5: Simulation Parameters for Different Geometry Radii

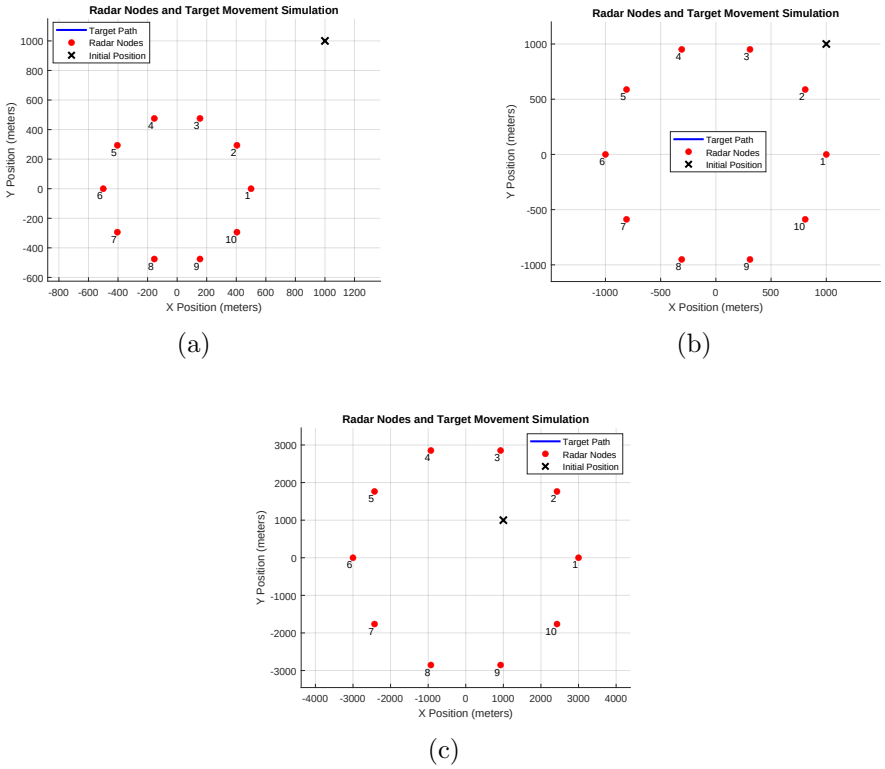

Figure 5.7: Simulation Setup for Variable Circular Geometry Radii: Different circular geometry radii with same initial target position at (1000,1000)m. Radii configurations are: a) 500m; b) 1000m; c) 3000m.

The uncertainty in target estimation is greater when the target is positioned outside the circular radar node array compared to when it is inside, as observed in Figure 5.8. This pattern holds true for the considered trajectory, indicating how the positioning of radar nodes relative to the target's path significantly affects the accuracy of the estimations. When the target is inside a circular array of radar nodes, the nodes surround it and gather range and Doppler frequency shift measurements from multiple angles. This arrangement allows for effective triangulation, accurately capturing the target's position and velocity in both the x and y directions. The favorable geometry reduces uncertainties because the lines of sight from different nodes intersect at wide angles, enhancing the precision of the estimates.

In contrast, when the target is outside the array, all the radar nodes are approximately on the same side relative to the target. Their measurements are thus more aligned and provide less information about certain components of the target's velocity. This leads to higher uncertainties because the lines of sight are more parallel with each other, which reduces the spatial diversity of the radar network and the effectiveness of triangulation, and so increases sensitivity to measurement errors.

In designing a radar network with a circular geometry, a key consideration is the radius of the array with respect to the area where targets are expected to move. A larger radius, with a fixed number of nodes, means the radar nodes are more spread out. This spacing allows the target, once within the array, more time and distance to maneuver while still being tracked by the radar, providing enhanced opportunities to observe the target from various angles. This extended coverage and observation time improve range and Doppler estimations, thereby enhancing the accuracy of target tracking in 2D space. However, the increased distance between nodes and the fusion center introduces a drawback: longer data transmission times and potentially more power, which can impact the promptness and utility of the data in real-time applications.

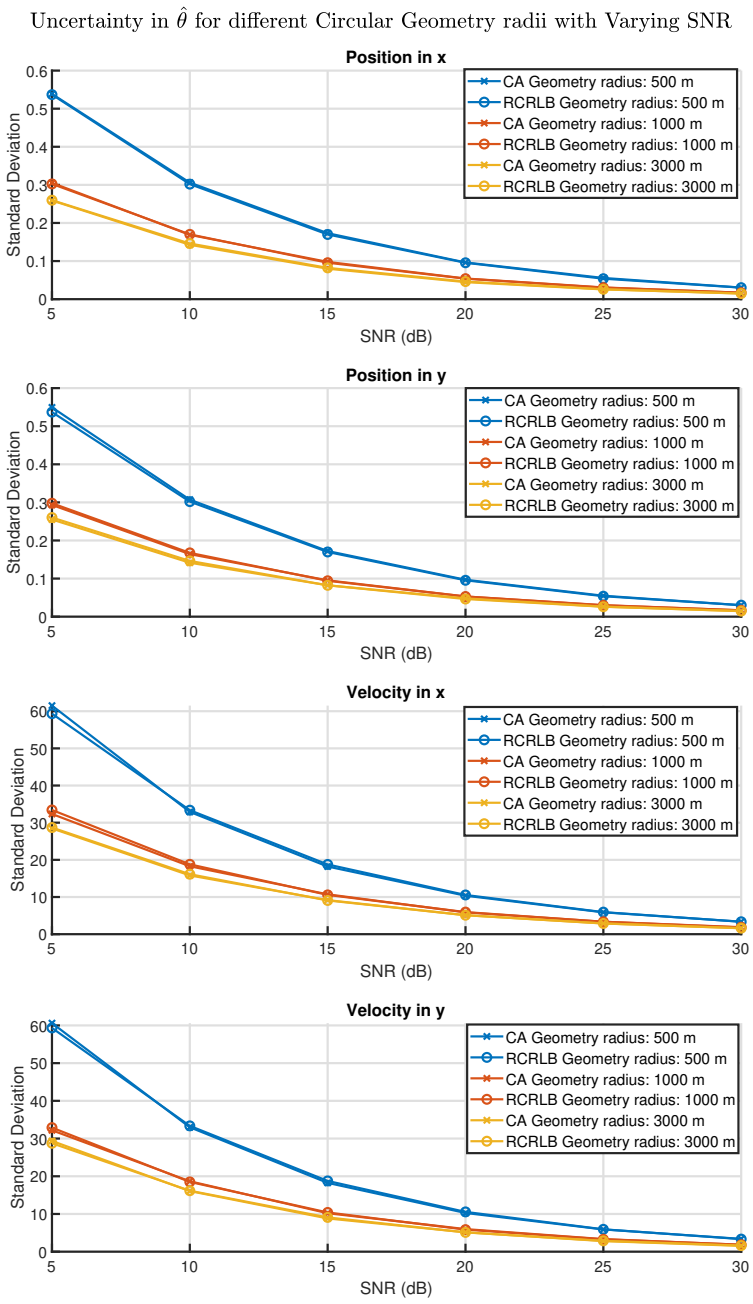

Figure 5.8: Effect of 500m, 1000m, and 3000m geometry radii on estimating $\hat{\boldsymbol{\theta}}$: Analysis with 10 nodes and a target starting at (1000, 1000)m across varying SNR levels.

5.1.6 Key Takeaways in Decentralized Approach

- In Section 5.1.1, we analyze how uncertainties in position and velocity along the x and y directions are affected by varying SNR levels and different antenna array geometries. While the Uniform Linear Array (ULA) is easier to design and implement, it introduces directional bias, as shown in Figure 5.1. To eliminate this bias, we use a Uniform Circular Array (UCA) in further simulations, which is not affected by the target's movement direction.

- In Section 5.1.2, we investigate how varying the number of radar nodes impacts the uncertainties in position and velocity estimates within a 2D space. Increasing the node count provides more data and leads to better estimation accuracy. However, this improvement comes with the trade-off of higher computational demands to process a single estimate. Therefore, for future simulations, we choose to use 10 nodes as an optimal balance between enhanced accuracy and manageable computational requirements.

- In section 5.1.3, we investigate how the number of pulses in a single burst affects the accuracy of range and Doppler shift measurements. For a slow-moving target, increasing the number of pulses significantly reduces velocity uncertainty, leading to more precise measurements. However, for a fast-moving target, we cannot spend as much time collecting pulses because the target moves considerably during a single burst, introducing range uncertainty. This creates a trade-off: using more pulses improves velocity estimates but can degrade range accuracy for fast-moving targets. Balancing this trade-off is essential in radar system design.

- In Section 5.1.4, we examine how the number of measurements (M) collected before sending data to the fusion center affects the final estimation. Collecting more measurements improves the estimation accuracy and reduces uncertainty. However, this also increases bandwidth requirements and computational complexity due to the larger data volume. To balance accuracy with resource constraints, we use 32 measurements in our simulations.

- In Section 5.1.5, we examine how the geometry radius of the circular array affects target estimation. A larger radius allows the target to remain within the array for a longer time, improving estimation accuracy due to extended observation. However, this also increases latency because the fusion center may be farther from some radar nodes, leading to delays in data processing. Conversely, a smaller radius means the radar nodes are closer together, but if the target moves outside the array, uncertainties in estimating all four parameters—position and velocity in both the x and y directions—increase. This highlights a trade-off between improved accuracy and increased latency based on the geometry radius, emphasizing the need to balance array size with operational requirements.

5.2 Distributed Approach

In the distributed approach, each radar node functions within a specified communication range, using the Alternating Direction Method of Multipliers (ADMM) to independently estimate $\boldsymbol{\theta}$ and to achieve consensus among all nodes in the network. Following the simulation setup determined from the results of the Decentralized approach, we execute the distributed approach at higher Signal-to-Noise Ratios (SNRs).

5.2.1 Reaching Consensus

Parameter	Value
Radar Nodes (N)	10
Initial Position (m)	(1000, 1000)
Constant Velocity (m/s)	20
Measurements (M)	32
Geometry	Circular
Monte Carlo Simulations	1000
Number of Pulses (L)	32
Signal to Noise Ratio (SNR)	30dB

Table 5.6: Simulation Parameters for Distributed Approach

The parameters for the simulation setup are outlined in Table 5.6, demonstrating the distributed approach utilized throughout the study. The simulation setup utilizes a Unit Disk Graph (UDG) as the communication network, as detailed in Chapter 4 Section 4.3.1. To effectively localize the target, the triangulation property necessitates at least three radar nodes as explained in Section 4.1. Consequently, each radar node is configured with a communication radius of 3 kilometers. This setup ensures that every node communicates with at least two neighboring nodes in a circular configuration, facilitating efficient data transfer. This arrangement optimizes local information sharing among adjacent nodes, which is instrumental in collectively converging toward the accurate estimation of the target's parameters. Additionally, the system operates with a Signal-to-Noise Ratio (SNR) of 30dB, which enhances the quality of the data exchanged between the radar nodes, facilitating more accurate and reliable target estimation. This framework illustrates the effectiveness of localized node communication in achieving collective accuracy in a distributed radar system.

In the simulation described in Figure 5.9, the convergence of the velocity parameters, $\hat{\dot{x}}$ and $\hat{\dot{y}}$, requires more iterations than the position parameters. Each radar node measures the radial velocity of the target, which is the component of the target's velocity directed along the line of sight from the radar to the target. Since the target's orientation relative to each radar node varies, the measured radial velocity also varies from node to node. Essentially, each radar node perceives the target's motion differently based on its unique viewpoint. This

variation in perceived velocity means that averaging out these measurements to determine the true velocity vectors, $\hat{x}$ and $\hat{y}$, across the network requires more iterations. The radar nodes can only exchange information with their two nearest neighbors due to the 3000m communication radius. This localized communication restricts the rate at which information disperses across the network, slowing down the process of aligning these diverse velocity measurements into a consensus on the target's actual velocity.

However, despite this slower convergence, it is notable that the nodes, by merely communicating with adjacent neighbors, successfully align to the true value. This method of data exchange effectively eliminates the single point of failure inherent in Decentralized systems, enhancing the reliability and robustness of the radar network.

The performance of our algorithm across different radar node geometries is depicted in Figure 5.10: circular, straight line, and semicircular, all within a communication radius of 3000 meters. Observations indicate a similar convergence pattern among all radar nodes in the straight line geometry. This similarity in convergence is due to the alignment and viewing angles of the nodes towards the target, which are more uniform in the straight line setup compared to the circular and semicircular geometries. Additionally, in the straight line geometry, each radar node communicates with a greater number of nodes within the 3000m radius, compared to the other two configurations where the nodes are more spread out. Moreover, it is observed that the number of iterations required to reach stability is nearly the same across all configurations, typically around 60-70 iterations for all nodes.

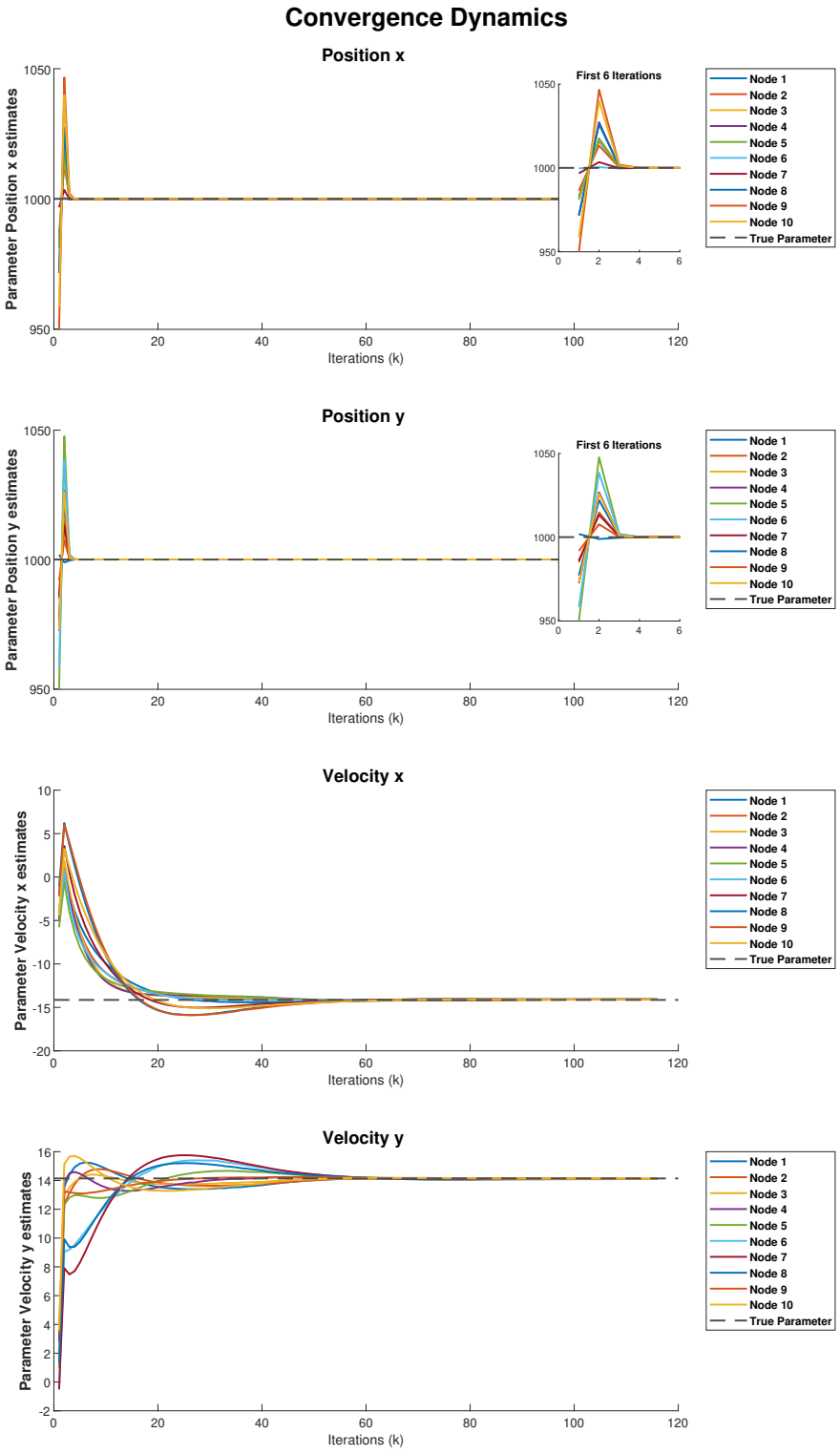

Figure 5.9: Convergence Dynamics in Circular Geometry: Each node's behavior at all k iterations with a communication radius of 3000 meters at 30dB SNR.

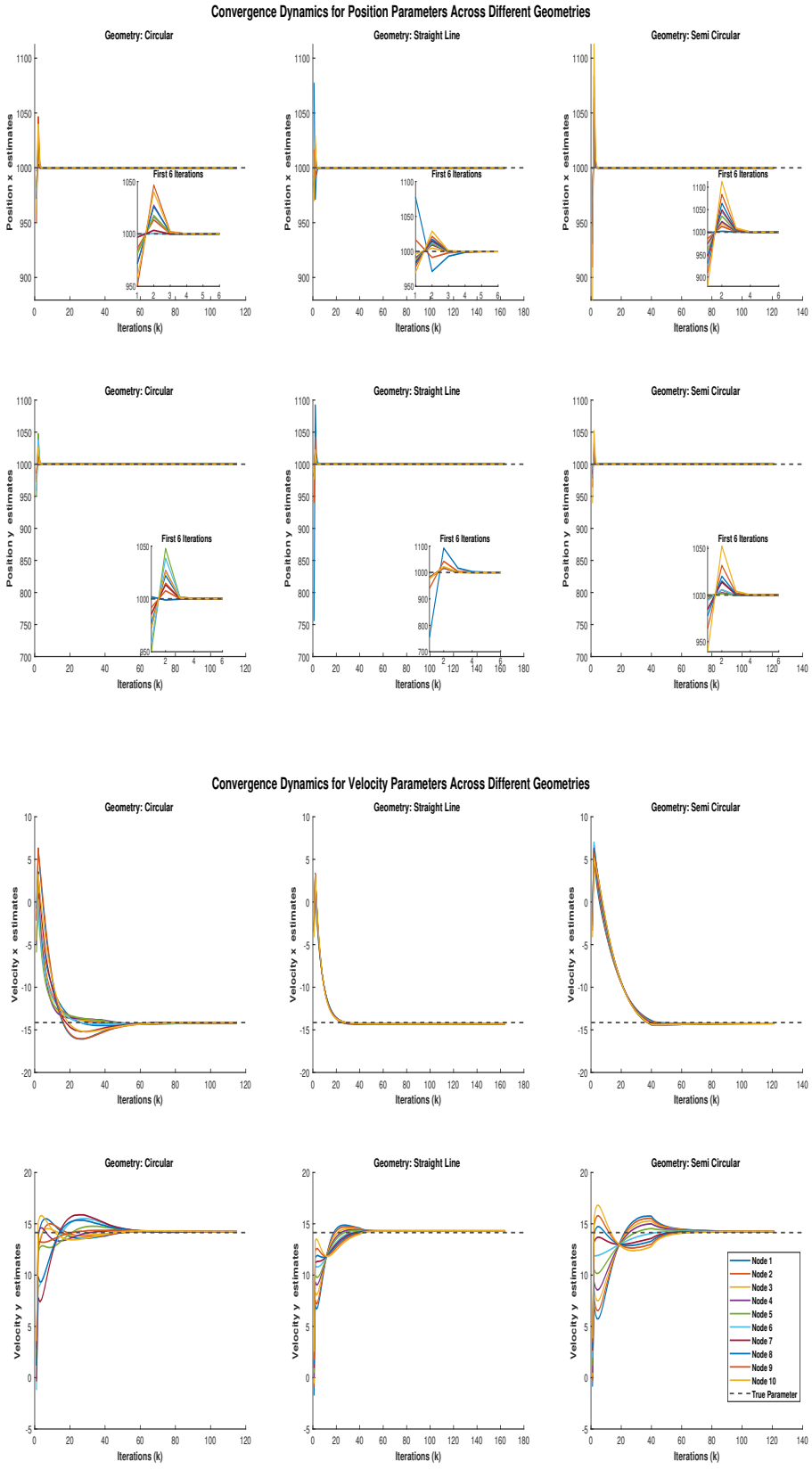

Figure 5.10: Convergence Dynamics Across Geometries: Consensus behavior of radar nodes across Circular, Straight Line, and Semi-Circular geometries with a 3000-meter communication radius and 30 dB SNR.

5.2.1.1 Consensus Achievement with Operational Efficiency

In this section, we display the outcomes using distinct stopping criteria for position and velocity primal residuals, detailed in Algorithm 2 in Chapter 4.3.3.4. Unlike the previous approach that utilized a single stopping criterion for the entire equation, we now apply separate criteria for position and velocity. This division reduces the computational load of the algorithm.

As illustrated in Figure 5.11, the position parameters in the simulation reach consensus more quickly than the velocity parameters. Specifically, the position parameters typically converge within about 10 iterations. In contrast, the velocity parameters, due to the complexity of averaging out diverse measurements from various viewpoints, require significantly more iterations to reach consensus, usually around 70 to 80 iterations. This discrepancy underscores the challenge in aligning velocity data across the network, as each radar node contributes unique radial velocity measurements based on its specific line of sight to the moving target.Once the stopping criteria for the position are met, we fix the position values and subsequently focus solely on the velocity parameters. This approach lightens the algorithm's load by reducing the number of variables to solve for from four to two. While this may not drastically decrease the number of iterations ('k') required to achieve consensus across all the nodes for velocity parameter, it does mean that, beyond a certain point, optimization occurs with only two variables at each iteration instead of four.

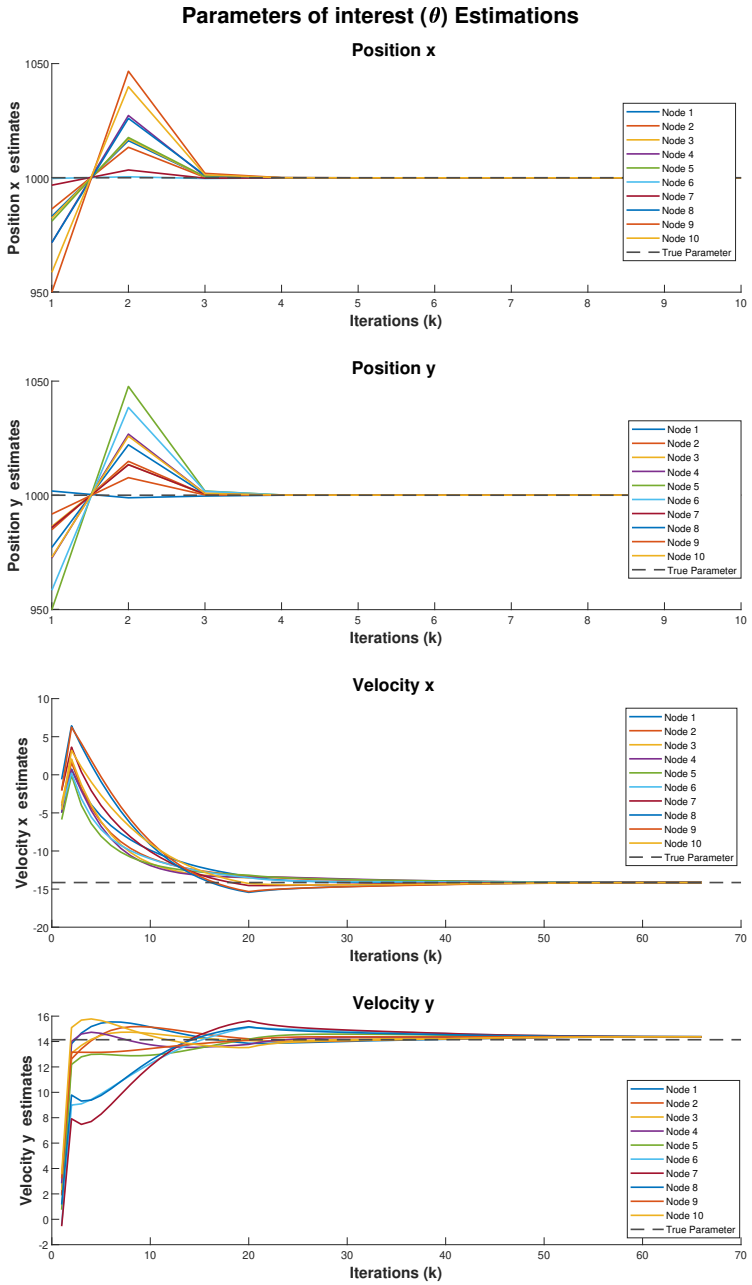

Figure 5.11: Convergence Dynamics with Distinct Parameter Stopping Points: Differing convergence iterations for each parameter, in contrast to simultaneous convergence in Figure 5.9

5.2.2 Impact of Communication Radius on Convergence Speed

In this simulation, we adjust the communication radius to assess its impact on the speed of convergence to the true value within a circular geometry setup. The number of iterations required for convergence is directly influenced by the size of the communication radius. While a larger communication radius allows each radar node to interact with more nodes, potentially speeding up convergence, it also requires the processing of more data, thus increasing the processing load on each radar node. This trade-off suggests that significantly increasing the communication radius may not be as beneficial if the gains in convergence speed are minimal compared to the additional resource expenditure.

Furthermore, the initial guess from which the convergence process starts also plays a crucial role in achieving faster convergence. A more accurate initial guess can significantly reduce the number of iterations needed, making the process more efficient. Thus, optimizing both the communication radius and the accuracy of the initial estimates is key to enhancing the performance of the radar network in simulations. The selection of the initial guess will not be discussed here and left for future investigation.

Increasing the number of nodes that each radar communicates with leads to faster convergence towards the true value, as observed in Figure 5.12. The specific reduction in the number of iterations required is detailed in Figure 5.13, highlighting the significant impact of enhanced network connectivity on the efficiency of consensus among the radar nodes. This simulation was conducted using 100 Monte Carlo runs for each of the different communication radii scenarios. Specifically, we analyzed three settings: a communication radius of 3000m where each radar communicates with 2/10 neighboring nodes, 4500m where each communicates with 4/10 neighboring nodes, and 5500m where each communicates with 6/10 neighboring nodes. Whenever ratios such as 4/10 or 6/10 are mentioned, they refer to the number of nodes each node is communicating with relative to the total number of nodes in the system.

Figure 5.13 provides a deeper analysis of the number of iterations required to achieve convergence for each communication radius. A significant reduction in the number of iterations is noted when the number of neighboring nodes each radar communicates with increases from 2 to 4, from around 100 to 67. However, once each node starts communicating with more than half of all nodes in the network (anything above 5 neighbors in this case), adding more connections does not speed up consensus significantly. This is because the essential information needed to form an accurate estimate is already being captured with the existing connections. Adding more connections beyond this point does not appear to contribute much new information. The iterations it takes beyond this are primarily to reach an accurate consensus among all the radar nodes in the system.

Figure 5.14 illustrates the data volume processed by each node to estimate $\boldsymbol{\theta}$. The figure illustrates the total data volume managed by each radar node, assuming

that each range and Doppler measurement, essential for estimating $\hat{\boldsymbol{\theta}}$, is stored as a 64-bit float, occupying 8 bytes of data. This data is transferred between radar nodes to facilitate the accurate estimation of $\hat{\boldsymbol{\theta}}$.

$$Data\ Size_n = (|\mathcal{N}_n| + 1) \times M \times 2 \times 64 \tag{5.1}$$

where, $(|\mathcal{N}_n| + 1)$ denotes the number of nodes from which data is collected, including the node itself and its neighbors, and M indicates the number of measurements each node collects. As explained in Chapter 4 Section 4.3.1, $(|\mathcal{N}_n| + 1)$ represents the cardinality, indicating the number of nodes the nth node is communicating with. In Equation 5.1, the '2' accounts for the range and Doppler measurements, and '64' refers to the data size in bits (8 bytes) for one measurement. Technically, one node sends 32 measurements (where $M = 32$) to its neighbors, and each measurement consists of one range and one Doppler frequency shift estimate, with each estimate comprising 64 bits. The plot compares the data calculated by each node against the node range, under varying communication radii. It is noted that if each radar node achieves faster consensus by communicating with only half of the total nodes, the amount of data processed significantly decreases, thus reducing the processing load on each radar node. This graph provides an example of how the data load on each radar node is reduced when estimating $\boldsymbol{\theta}$ in a distributed framework.

Mean Squared Error (MSE) calculations, used to analyze convergence dynamics, are depicted in Figures 5.12 and 5.15. It is observed that fluctuations in velocity MSE occur during the convergence process when each radar node communicates with fewer than half of the total nodes.This occurs because velocity estimation in 2D relies on radial velocity measurements from each radar node. When each radar node communicates with only a few others—less than half of the nodes in the system—it can lead to inconsistencies in the velocity estimates across all the radar nodes. This happens because each node bases its calculations on radial velocity measurements from only its nearest neighbors. Without a broader range of data from more nodes, these limited interactions might produce different velocity results compared to nodes that are receiving information from different or additional neighbors. During consensus efforts, the influence of nodes with divergent velocity estimates may cause some spikes in MSE, prompting adjustments in the primal and dual variables within the proposed algorithm to mitigate this error. In contrast, smoother convergence is achieved when each radar node communicates with more than half of the nodes in the network, as this allows for more comprehensive data gathering, thereby enhancing the accuracy of the target's velocity estimation in 2D.

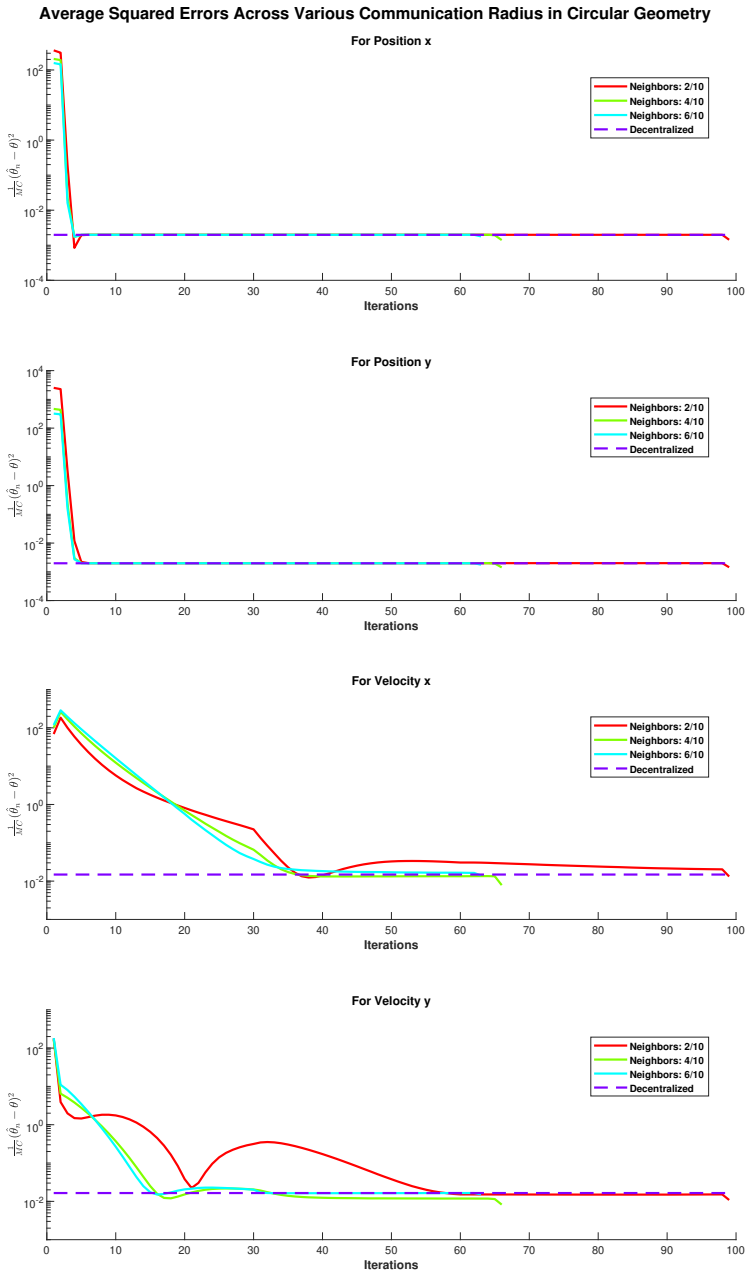

Figure 5.12: Convergence Dynamics of $\hat{\boldsymbol{\theta}}_1$ in circular geometry with communication radii of 3000m, 4500m, and 5500m at 30dB.

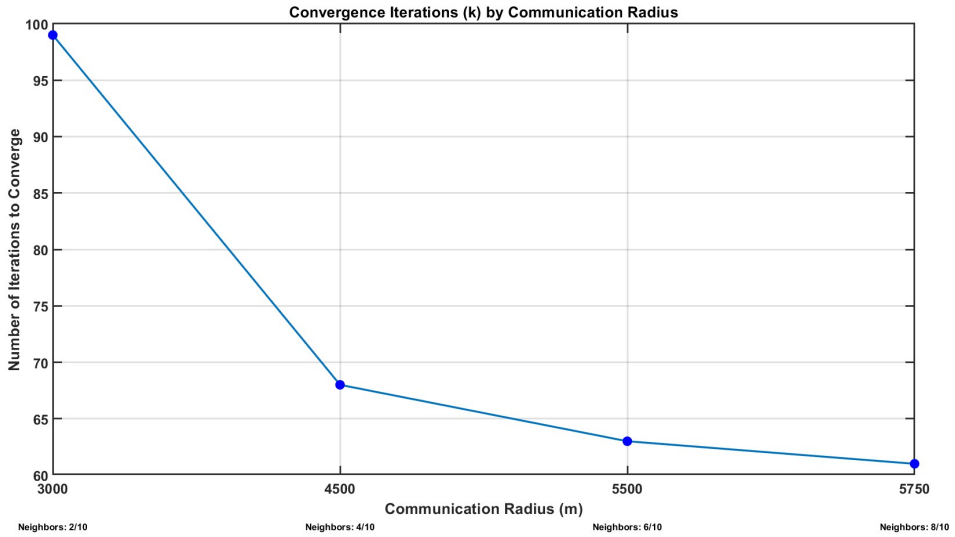

Figure 5.13: Iterations for convergence across different communication radii averaged across all the radar nodes in the system.

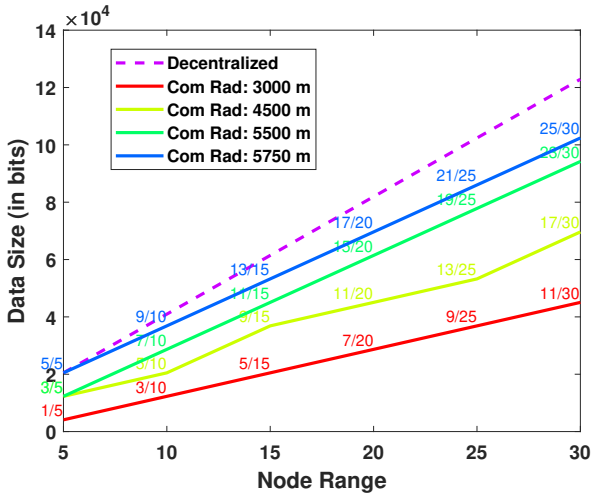

Figure 5.14: Data Size Processed by Each Radar Node vs. Node Range for Different Communication Radii for a circular geometry.

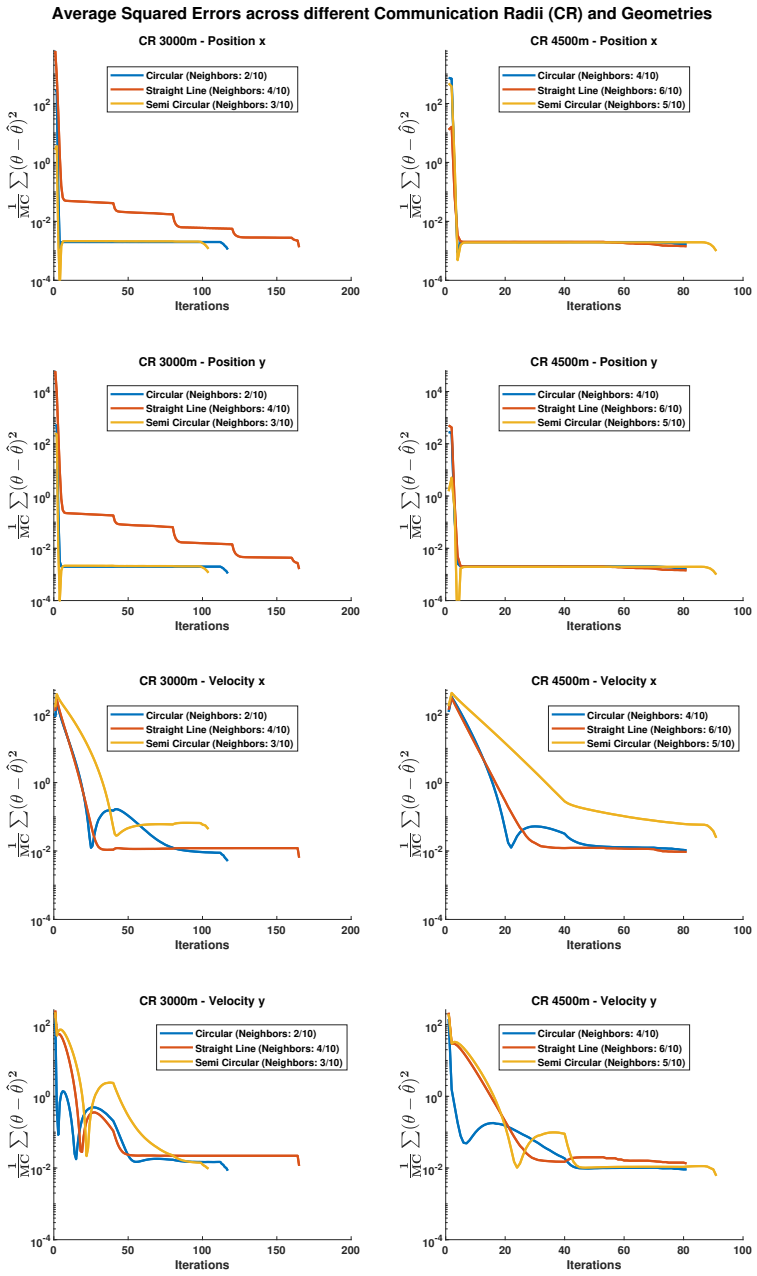

Figure 5.15: Convergence Dynamics of $\hat{\boldsymbol{\theta}}_1$ for Circular, Semi-Circular and Straight Line geometries for Communication Radii of 3000m and 4500m at 30dB SNR.

5.2.3 Impact of Penalty Parameters on Convergence Accuracy

The impact of penalty terms on convergence is explored in Section 4.3.3.1, specifically within Equation 4.15, and plays a pivotal role in the ADMM update equations for the estimation of $\hat{\boldsymbol{\theta}}_n$. These terms are instrumental in steering the solution towards the optimal value during the convergence process. These terms are quite sensitive; a small adjustment, such as altering a penalty term by a factor of 10, can result in inaccurate estimates. This sensitivity underscores the need for careful calibration of these parameters. Table 5.7 in the thesis details the specific penalty parameters we employed in our analysis.

SNR (dB)	Position Penalty	Velocity Penalty
5	10^9	3×10^7
10	10^8	3×10^6
15	8×10^7	2×10^6
20	10^7	3×10^5
25	10^6	3×10^4
30	8×10^5	2×10^4

Table 5.7: Penalty Parameters for Different SNR Values

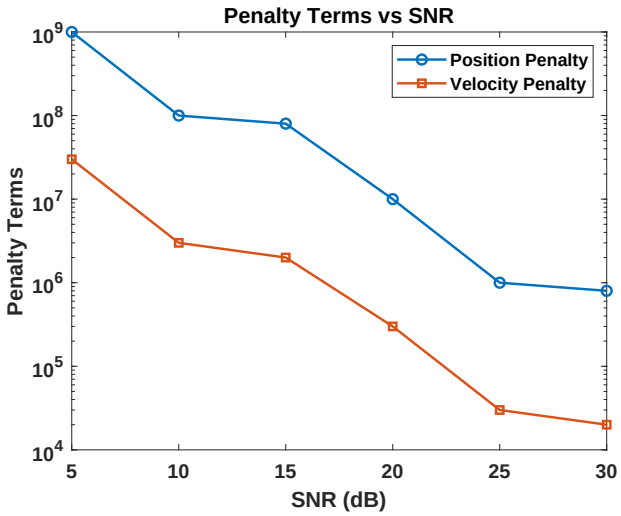

Figure 5.16: Relationship between SNR levels and penalty terms for position and velocity. Same data as Table 5.7.

The relationship between penalty terms and Signal-to-Noise Ratio (SNR) is visually depicted in Figure 5.16, which presents the same data as Table 5.7. This illustration highlights how the penalty terms are adjusted in response to varying levels of SNR. At higher SNRs, the measurements each radar node receives are less affected by noise. This results in higher loglikelihood values, reflecting reduced uncertainty and indicating that the parameters estimated by each node align more precisely with the actual values. Consequently, smaller penalty terms are sufficient

to facilitate consensus among nodes, as less correction is needed to reconcile the estimates. Conversely, at lower SNRs, the increased noise in measurements leads to lower loglikelihood values and greater uncertainty. To compensate for these inaccuracies and achieve agreement across nodes, larger penalty terms are required. Therefore, the adjustment of penalty terms is intricately connected to SNR.

One effective way to enhance the convergence speed of ADMM is by adaptively updating the penalty matrix. In Equation 5.2, ADMM employs an adaptive approach to modifying the penalty matrix, which not only accelerates convergence but also reduces the algorithm's dependency on the initial penalty terms listed in Table 5.7. The key idea is to adjust the penalty matrix based on the primal and dual residuals in ADMM. The primal residual measures the violation of the consensus constraints, while the dual residual reflects the consistency between successive updates of the variables in the optimization process, indicating how well the dual variables are converging toward optimality [29]. Here τ_{inc} is τ, and τ_{dec} is $\frac{1}{\tau}$.

$$\mathbf{\Phi}(k+1) = \begin{cases} \tau_{\text{inc}} \times \mathbf{\Phi}(k), & \text{if } \|r_n(k)\|_2^2 < 10 \cdot \|s_n(k)\|_2^2 \\ \tau_{\text{dec}} \times \mathbf{\Phi}(k), & \text{if } \|s_n(k)\|_2^2 < 10 \cdot \|r_n(k)\|_2^2 \\ \mathbf{\Phi}(k), & \text{otherwise} \end{cases} \tag{5.2}$$

The Distributed ADMM with Adaptive Penalty Matrix for Target Localization at the nth radar node is detailed in Algorithm 3. This algorithm builds upon the foundation laid by Algorithm 1, incorporating a new feature that adaptively changes the penalty matrix, as specified in Equation 5.2.

Algorithm 3: DADMM with Adaptive Penalty Matrix Update for Localization (nth node)

Input: Initialize $\boldsymbol{\psi}_{nj}(0)$, $\boldsymbol{\theta}_n(0)$ and $\boldsymbol{\vartheta}_{nj}(0)$ randomly $j \in \mathcal{N}_n$; Define $\boldsymbol{\Phi}(k)$ based on SNR; Set $k = 0$;

while $(r_n(k+1) > \epsilon^{pri})$ **do**
- Update $\boldsymbol{\theta}_n(k+1)$ using (4.17a);
- Transmit $\boldsymbol{\psi}_{nj}(k)$ and $\boldsymbol{\theta}_n(k+1)$ to neighbors $j \in \mathcal{N}_n$;
- Update $\boldsymbol{\vartheta}_{nj}(k+1)$ using (4.17b);
- Transmit $\boldsymbol{\vartheta}_{nj}(k)$ to neighbors $j \in \mathcal{N}_n$;
- Update $\{\boldsymbol{\psi}_{nj}(k)\}_{j\in\mathcal{N}_n}$ using (4.17c);
- **if** $(\|r_n(k)\|_2^2 < 10 \cdot \|s_n(k)\|_2^2)$ **then**
  - Update $\boldsymbol{\Phi}(k)$ using (5.2);
- **else if** $(\|r_n(k)\|_2^2 > 10 \cdot \|s_n(k)\|_2^2)$ **then**
  - Update $\boldsymbol{\Phi}(k)$ using (5.2);
- **else**
  - $\boldsymbol{\Phi}(k) = \boldsymbol{\Phi}(k-1)$;
- **end**
- Increment k;

end

Output: $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_n$

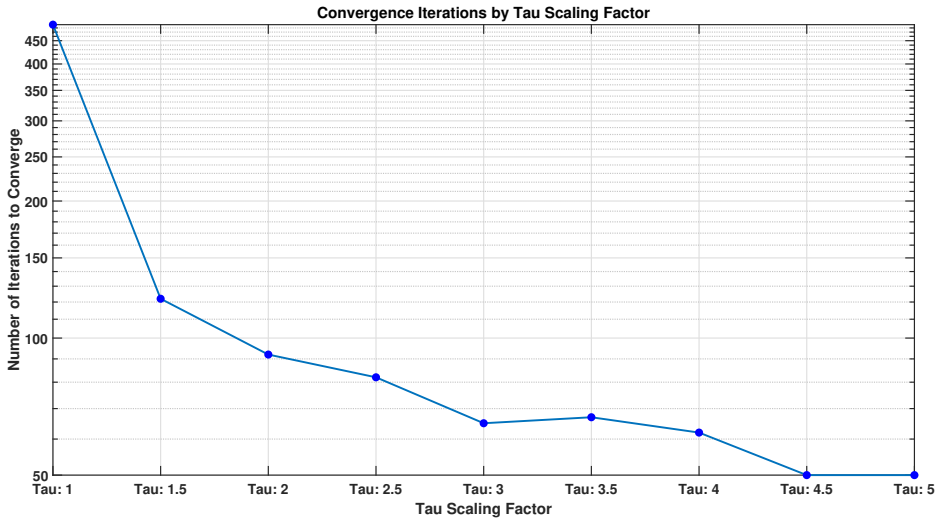

Figure 5.17: Effect of τ on convergence iterations and reaching consensus.

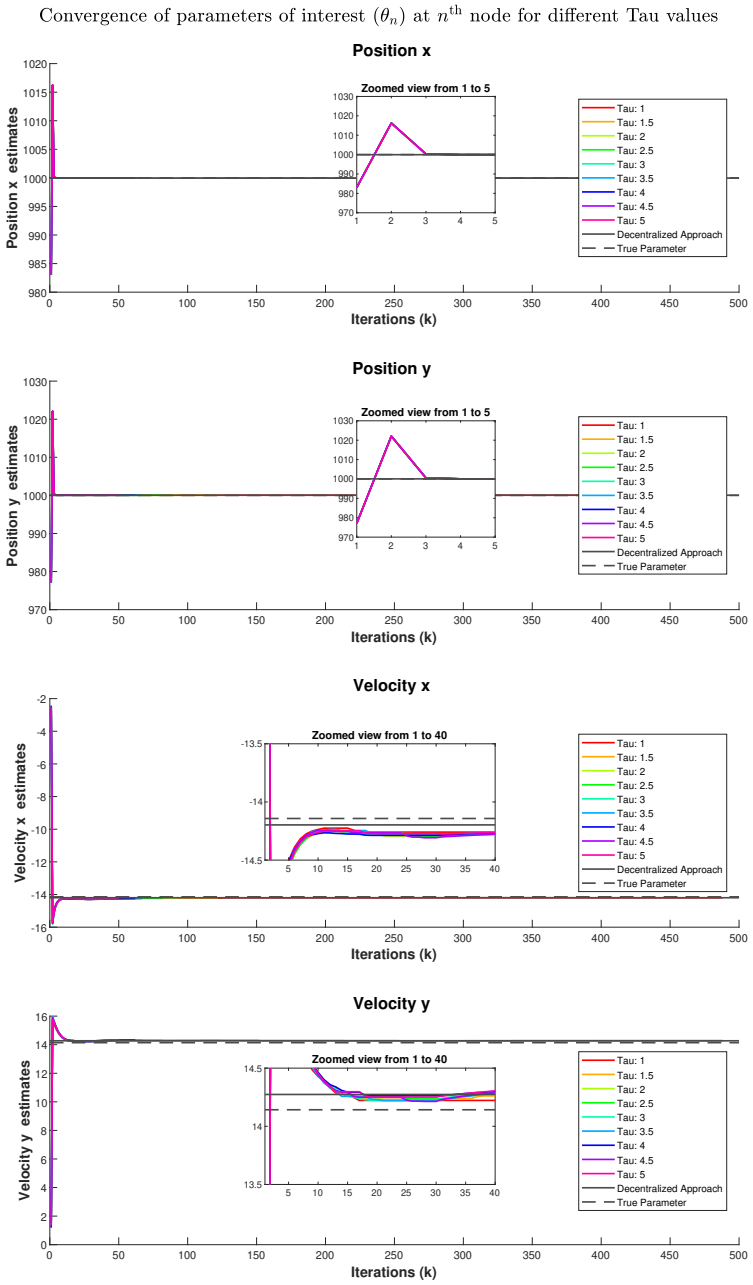

Figure 5.18: Impact of various τ values on convergence and consensus for $\hat{\boldsymbol{\theta}}_1$.

The adaptation of the penalty matrix based on primal and dual residuals substantially enhances convergence, as illustrated in Figures 5.17 and 5.18. This adaptation effectively demonstrates how changes in the multiplier factor, τ, influence the number of iterations required for convergence. Notably, when

τ increases from 1 to 2, the number of iterations needed dramatically reduces from 500 to 90. Further increases in τ beyond 4 yields no significant reduction in convergence iterations. Importantly, the simulations conducted up to now utilized a $\tau = 2$, under which it takes about 80-90 iterations to reach consensus. Meanwhile, Figure 5.18 illustrates that at high Signal-to-Noise Ratios (SNRs), varying τ enables all parameters to converge to their true values, showcasing the effectiveness of the adaptive update strategy across different τ values. From Figure 5.18, it is confirmed that the true value is reached with various values of tau, and the number of iterations required to reach consensus changes accordingly. When tau is set at 1, it takes about 500 iterations to reach consensus, while a tau of 2 requires around 90 iterations; however, both settings ultimately converge to the true value.

5.2.4 Key takeaways in Distributed Approach

- In Section 5.2.1, we transition from a Decentralized approach to a Distributed approach where each radar node communicates only with its nearest neighbors. Through local interactions, the nodes collectively reach consensus after a few iterations, eliminating the need for a central coordinator. Figure 5.10 demonstrates this distributed consensus algorithm applied to all three geometries we initially discussed, showcasing its effectiveness across different array configurations. In Subsection 5.2.1.1, we demonstrate a method where we terminate the iterations for one set of parameters early, allowing us to concentrate on optimizing the remaining parameters. This strategy effectively reduces the computational effort required to reach the optimal solution.

- In Section 5.2.2, we investigate the impact of increasing the communication radius—that is, how many radar nodes each node communicates with—on the number of iterations required for convergence in a distributed optimization setting. We find that when each radar node communicates with more than half of the nodes in the entire system, the number of iterations to converge becomes almost constant. The iterations needed are primarily for achieving consensus (satisfying the constraints), suggesting that beyond this point, increasing the communication radius does not significantly reduce the convergence iterations.

- In Section 5.2.3, we demonstrate the inverse relationship between penalty terms and SNR in distributed optimization. Lower SNR implies more noise and poorer initial estimates, more variance. As a result, we require higher penalty terms to reduce the primal residual and achieve consensus among the radar nodes. Furthermore, we show that convergence can be accelerated by adaptively updating the penalty terms based on monitoring the primal and dual residuals during the iterative process.

6 Conclusion and Future Scope

In this chapter, we outline the main conclusions of this thesis and discuss possible directions for further improvement.

6.1 Conclusion

This section summarizes the contributions of this thesis and reflects on the research objectives introduced in Chapter 1. The thesis presents a Distributed ADMM method for estimating the position and velocity of a single target in 2D space, addressing critical gaps identified in existing methods. Specifically, issues such as single points of failure and excessive data processing loads at the fusion center were highlighted in Chapter 2 as significant challenges to the scalability and robustness of radar networks. The Distributed ADMM method proposed here overcomes these challenges by allowing each node to independently calculate the target's position and velocity while communicating only with adjacent nodes to achieve consensus. This approach is novel and, to the author's knowledge, has not been documented in prior research.

Initially, a basic simulator is developed, detailed in Chapter 3, which generates range and Doppler frequency shift measurements by introducing noise to true values to adhere to CRLB standards. This simple simulator served as a foundation for basic analysis and testing within a decentralized approach to estimate the position and velocity of a target in 2D space. This step was essential to check the simulator's accuracy and ensure it matched existing methods, confirming that the generated measurements were reliable for further experiments and analysis. Building on the simulation scenario established, the Distributed ADMM Algorithm was implemented. This approach eliminates single points of failure and significantly reduces computational demands at each node by limiting communication to a nearby radius, effectively addressing scalability issues. The simulations have confirmed that the algorithm can accurately estimate the target's position and velocity at high SNRs.

To enhance computational efficiency, an algorithm with separate convergence monitors for position and velocity has been introduced. This method effectively reduces unnecessary calculations by applying dual stopping criteria, allowing position estimates to converge quickly and be stored after approximately 10 iterations. Subsequently, the algorithm shifts its focus exclusively to velocity, which requires around 70 iterations to converge in a scenario involving a 3000 m communication radius.

The selection of penalty terms in relation to SNRs has been effectively demonstrated to ensure that consensus is reached without destabilizing the system. The penalty terms decrease by approximately 10 times for every 5dB increase in SNR. Additionally, an adaptive penalty matrix has been implemented, which significantly reduces the iterations required for nodes to converge from about 450 to just 50 at high SNRs by incorporating a multiplication constant $'\tau'$ that adjusts based on the primal and dual residuals.

In summary, this thesis presents a transition from a decentralized to a distributed approach, addressing the issue of a single point of failure while significantly reducing the computational load at each node to ensure smooth and efficient operation. Consensus has been achieved by sharing only range and Doppler frequency shift measurements with neighboring nodes, thus improving scalability and reliability. The optimization of penalty terms relative to SNRs for faster convergence has been explored, and a two-stopping-criteria method has been implemented to streamline local computations and enhance efficiency.

6.2 Future Scope

This project introduces an approach where data distribution and estimations are performed locally at each radar node, offering a foundational framework for further exploration. However, it's important to acknowledge that the measurements generated in this study are based on a range of assumptions and are relatively simplistic in nature. Several avenues for future research have been identified, which could build upon the findings of this thesis. Some of the future recommendations to further this work is mentioned below:

- For future development, instead of introducing errors to the true values, we could generate range and Doppler estimates using the traditional Fast Fourier Transform (FFT) method. Specifically, for Linear Frequency Modulation (LFM) signals, the range profile could be obtained through matched filtering, followed by an FFT performed along the slow time axis. Additionally, introducing elements such as clutter, which were not initially considered, would create a more realistic simulation environment. This method would provide a more conventional and robust baseline for estimating these parameters, thereby enhancing the simulator's capability to reflect real-world complexities.

- Currently, as outlined in Chapter 5, Section 5.2.3, it takes approximately 80-90 iterations to converge when the parameter τ is set to 2 and there are 2 neighboring nodes. Future work could focus on reducing the number of update iterations required to achieve convergence while still reaching consensus on the true values. The distributed computing literature presents various methods aimed at accelerating convergence. One potential improvement could involve making τ adaptive rather than fixed, allowing it to adjust in response to the rate of convergence, which could enhance the convergence process. Additionally, we have not observed how convergence behaves when

the target travels at higher velocities. This aspect could also be explored to further understand and improve the system's performance under different operational conditions.

- Currently, our focus has been primarily on the performance of Distributed ADMM (DADMM) methods, without extensive comparison to other algorithms. There are several alternative algorithms, such as the Primal Dual Method of Multipliers (PDMM) [62], known for its faster convergence, and Geographic Gossip [63], which could be explored further. Future research could compare these methods to DADMM in terms of convergence speed, computational efficiency per iteration, and overall robustness, to better understand their potential advantages and trade-offs, , including whether the theoretical boundaries meet and what implications this has for further optimization.

- Future research could extend the current model to incorporate tracking capabilities, enabling the radar to simultaneously collect data and communicate with neighboring nodes for faster convergence and consensus. Ideally, the time taken to achieve convergence and reach consensus should be shorter than the time required for data collection, facilitating real-time processing and decision-making. This enhancement would be especially beneficial once we refine our simulator to more accurately replicate real-world conditions, such as incorporating realistic noise, clutter, and other environmental factors, as well as scenarios where targets move at varying speeds. Further, by comparing the enhanced model's performance with established state space distributed tracking approaches like Distributed Kalman Filters [64] and Particle Filtering methods, we could gain deeper insights into the efficiency and accuracy of our approach.

- In the future, attention can be directed toward analyzing the data flow within the network and addressing potential congestion in the distributed approach. Investigating how data flow impacts the system's performance, particularly under bandwidth limitations, could provide valuable insights. Key considerations include understanding how such constraints affect the accuracy and speed of estimations and determining the optimal amount of data that should flow through the network during a single iteration. Additionally, power consumption is a critical factor to explore, as efficient data flow can help minimize the energy required for communication and computation at each node. Balancing bandwidth usage, power consumption, and estimation accuracy will be essential for improving the overall efficiency and practicality of distributed radar systems.

Bibliography

[1] M. Richards, J. Scheer, and W. Holm, *Principles of Modern Radar*. No. v. 3 in Principles of Modern Radar, SciTech Pub., 2010.

[2] M. Richards, *Fundamentals Of Radar Signal Processing*. McGraw-Hill Education (India) Pvt Limited, 2005.

[3] MATLAB, "Linear Frequency Modulated Pulse Waveforms." https://nl.mathworks.com/help/phased/ug/linear-frequency-modulated-pulse-waveforms.html.

[4] D. Dhulashia, M. Temiz, and M. A. Ritchie, "Performance of range and velocity estimation in a multistatic radar network with receiver swarms," in *International Conference on Radar Systems (RADAR 2022)*, vol. 2022, pp. 447–452, 2022.

[5] O. Tekdas and V. Isler, "Sensor placement for triangulation-based localization," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 681–685, 2010.

[6] B. Griffin, A. Balleri, C. Baker, and M. Jahangir, "Optimal receiver placement in staring cooperative radar networks for detection of drones," in *2020 IEEE Radar Conference (RadarConf20)*, pp. 1–6, 2020.

[7] T. Jia, K. C. Ho, H. Wang, and X. Shen, "Localization of a moving object with sensors in motion by time delays and doppler shifts," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5824–5841, 2020.

[8] H. Griffiths, "Multistatic, mimo and networked radar: The future of radar sensors?," in *The 7th European Radar Conference*, pp. 81–84, 2010.

[9] H. D. Griffiths, "Bistatic and multistatic radar," 2004.

[10] T. Aittomäki, H. Godrich, H. V. Poor, and V. Koivunen, "Resource allocation for target detection in distributed mimo radars," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 873–877, 2011.

[11] A. E. Mitchell, G. E. Smith, K. L. Bell, and M. Rangaswamy, "Single target tracking with distributed cognitive radar," in *2017 IEEE Radar Conference (RadarConf)*, pp. 0285–0288, 2017.

[12] P. STINCO, M. GRECO, and F. GINI, "Data fusion in a multistatic radar system," in *Synthetic Aperture Sonar and Radar 2010*, SAS SAR, Institute of Acoustics, Nov. 2023.

[13] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed clustering using wireless sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 707–724, 2011.

[14] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.

[15] R. Nakamura and H. Hadama, "Target localization using multi-static uwb sensor for indoor monitoring system," in *2017 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, pp. 37–40, 2017.

[16] C. Cui, J. Xu, R. Gui, W.-Q. Wang, and W. Wu, "Search-free dod, doa and range estimation for bistatic fda-mimo radar," *IEEE Access*, vol. 6, pp. 15431–15445, 2018.

[17] M. Jahangir, C. J. Baker, M. Antoniou, B. Griffin, A. Balleri, D. Money, and S. Harman, "Advanced cognitive networked radar surveillance," in *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6, 2021.

[18] P. Z. Peebles, *Radar Principles*. Wiley, 1998.

[19] X. Liu, J. Yan, and Y. Peng, "A distributed detection scheme for multiple monostatic radars," in *2006 CIE International Conference on Radar*, pp. 1–4, 2006.

[20] S. Jebali, H. Keshavarz, and M. Allahdadi, "Joint power allocation and target detection in distributed mimo radars," *IET Radar, Sonar amp; Navigation*, vol. 15, p. 1433–1447, June 2021.

[21] D. Dhulashia and M. A. Ritchie, "Multistatic radar data fusion for detection with reduced transmit power consumption," in *2023 IEEE Radar Conference (RadarConf23)*, pp. 1–6, 2023.

[22] D. Dhulashia and M. A. Ritchie, "Multistatic radar distribution geometry effects on parameter estimation accuracy," *IET Radar, Sonar & Navigation*, vol. 18, no. 1, pp. 7–22, 2024.

[23] M. Liggins, D. Hall, and J. Llinas, eds., *Handbook of multisensor data fusion*. Electrical Engineering & Applied Signal Processing Series, Boca Raton, FL: CRC Press, 2 ed., Sept. 2008.

[24] D. Sirmans and B. Bumgarner, "Numerical comparison of five mean frequency estimators," *Journal of Applied Meteorology and Climatology*, vol. 14, no. 6, pp. 991 – 1003, 1975.

[25] C. Shi, J. Zhou, and F. Wang, "Lpi based resource management for target tracking in distributed radar network," in *2016 IEEE Radar Conference (RadarConf)*, pp. 1–5, 2016.

[26] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 5, pp. 4997–5002 Vol.5, 2003.

[27] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[28] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, p. 127–239, Jan. 2014.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, p. 1–122, Jan. 2011.

[30] A. A. Zabolotsky and E. A. Mavrychev, "Distributed detection and imaging in mimo radar network based on averaging consensus," in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 607–611, 2018.

[31] P. Braca, S. Marano, and V. Matta, "Enforcing consensus while monitoring the environment in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3375–3380, 2008.

[32] G. Morral, P. Bianchi, and J. Jakubowicz, "On-line gossip-based distributed expectation maximization algorithm," in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 305–308, 2012.

[33] B. Lee, H. K. Song, Y. Suh, K. H. Oh, and H. Y. Youn, "Energy-efficient gossiping protocol of wsn with realtime streaming data," in *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, pp. 219–224, 2014.

[34] D. Drusvyatskiy, "The proximal point method revisited," 2017.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[36] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 303–320, 1969.

[37] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, p. 1750–1761, Apr. 2014.

[38] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," 2018.

[39] G. França and J. Bento, "Distributed optimization, averaging via admm, and network topology," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1939–1952, 2020.

[40] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[41] J. He, M. Xiao, and M. Skoglund, "Fast-converging decentralized admm for consensus optimization," in *2024 IEEE Conference on Artificial Intelligence (CAI)*, pp. 575–580, 2024.

[42] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[43] S. Parwana and D. S. B. R. Kumar, "Analysis of lfm and nlfm radar waveforms and their performance analysis," 2015.

[44] N. Levanon, *Radar principles*. wiley, 1988. "A Wiley-Interscience publication".

[45] H. Van Trees, *Detection, Estimation, and Modulation Theory, Part III: Radar-Sonar Signal Processing and Gaussian Signals in Noise*. Detection, Estimation, and Modulation Theory, Wiley, 2004.

[46] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1997.

[47] J. M. Steele, *The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities*. Cambridge University Press, 2004.

[48] A. K. Paul and T. Sato, "Localization in wireless sensor networks: A survey on algorithms, measurement techniques, applications and challenges," *Journal of Sensor and Actuator Networks*, vol. 6, no. 4, 2017.

[49] MATLAB, "fmincon." https://nl.mathworks.com/help/optim/ug/fmincon.html.

[50] Matlab, "fmincon Interior Point Algorithm." https://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html#brnpd5f.

[51] I. Pólik and T. Terlaky, *Interior Point Methods for Nonlinear Optimization*, pp. 215–276. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[52] R. J. Vanderbei and D. F. Shanno, "An interior-point algorithm for nonconvex nonlinear programming," vol. 13, p. 231–252, Apr. 1999.

[53] H. Godrich, A. M. Haimovich, and R. S. Blum, "Cramer rao bound on target localization estimation in mimo radar systems," in *2008 42nd Annual Conference on Information Sciences and Systems*, pp. 134–139, 2008.

[54] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer New York, 2001.

[55] M. Brede, "Networks—an introduction. mark e. j. newman. (2010, oxford university press.) 772 pages. isbn-978-0-19-920665-0.," *Artificial Life*, vol. 18, pp. 241–242, 2012.

[56] D. Gao, P. Chen, C. H. Foh, and Y. Niu, "Hop-distance relationship analysis with quasi-udg model for node localization in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, Sep 2011.

[57] B. HE and X. YUAN, "On the o(1/n) convergence rate of the douglas-rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

[58] J. Predd, S. Kulkarni, and H. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[59] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," 1969.

[60] H. Ougraz, S. Safi, and M. Frikel, "Performance study of linear and circular arrays based on wideband doa estimation," in *Artificial Intelligence and Smart Environment* (Y. Farhaoui, A. Rocha, Z. Brahmia, and B. Bhushab, eds.), (Cham), pp. 896–902, Springer International Publishing, 2023.

[61] A. R. Kulaib, R. M. Shubair, M. A. Al-Qutayri, and J. W. P. Ng, "Performance evaluation of linear and circular arrays in wireless sensor network localization," in *2011 18th IEEE International Conference on Electronics, Circuits, and Systems*, pp. 579–582, 2011.

[62] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *CoRR*, vol. abs/1702.00841, 2017.

[63] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient averaging for sensor networks," *CoRR*, vol. abs/0709.3921, 2007.

[64] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *2007 46th IEEE Conference on Decision and Control*, pp. 5492–5498, 2007.