5-23-2017

UAS Collision Warning and Passive Sensor Fusion Algorithms for Multiple Acoustic Transient Emitter Localization

Wenbo Dou
michael.wenbo.dou@gmail.com

UAS Collision Warning and Passive Sensor Fusion Algorithms for Multiple Acoustic Transient Emitter Localization

Wenbo Dou, Ph.D.

University of Connecticut, 2017

ABSTRACT

This dissertation considers two important topics in the area of estimation, target tracking and sensor fusion. The first topic is closest point of approach (CPA) prediction for unmanned aircraft systems (UAS) collision warning and the second topic is passive sensor fusion for multiple acoustic transient emitter localization.

To operate within a controlled airspace, UAS must have the capability to sense and avoid collisions with non-cooperative aircraft. This dissertation presents an inexpensive system design and develops an algorithm for estimating the CPA between the ownship and the intruder and a collision warning scheme using only bistatic range and range rate measurements from a multistatic radar.

Since it is vital for soldiers to be able to accurately localize sources of hostile fire in the battlefield for situational awareness and threat assessment, this dissertation develops both centralized and distributed passive sensor fusion algorithms to accurately estimate the number of acoustic transient emitters and their locations using bearing and time of arrival measurements.

UAS Collision Warning and Passive Sensor Fusion Algorithms for Multiple Acoustic Transient Emitter Localization

Wenbo Dou

B.E., Nanyang Technological University, Singapore, 2011

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2017

APPROVAL PAGE

Doctor of Philosophy Dissertation

UAS Collision Warning and Passive Sensor Fusion Algorithms for Multiple Acoustic Transient Emitter Localization

Presented by

Wenbo Dou, B.E. EE.

Major Advisor _____
Yaakov Bar-Shalom

Associate Advisor _____
Peter K. Willett

Associate Advisor _____
Krishna R. Pattipati

University of Connecticut

2017

ACKNOWLEDGMENTS

First of all, I want to thank Professor Yaakov Bar-Shalom for providing me such a priceless opportunity to pursue my Ph.D. degree under the supervision of (without "one of") the greatest researcher in the target tracking field. I feel honored to be his 38-th Ph.D. student. His continuous guidance and support make my Ph.D. journey enjoyable and unforgettable. I remember that when he offered me the Ph.D. opportunity, I did not even know random variables. Now I have mastered some of the state-of-the-art technologies to tackle practical challenging tracking problems. Not only do I benefit from his expertise in research, his serious attitude and his organized and systematic manner of conducting research as well as in other activities is also a valuable asset for my future.

Next, I would like to thank Professor Peter Willett, one of my associate advisors, for his kind help during the last few years. He has contributed to a lot of helpful comments to improve my papers. I enjoyed his comments a lot because I can not only learn the appropriate way of writing technical articles but always also learn new English from a native speaker. He is a great speaker to listen to in the classroom, in a technical seminar (You can verify this by watching a couple of his lectures on YouTube) or in a casual meeting. I had a great time in his advanced signal processing class where I officially learned particle filtering for the first time. He provided me the opportunity to serve as the student associate editor for IEEE Aerospace and Electronic Systems Magazine, which enriches my professional experience.

I am also grateful to Professor Krishna Pattipati, my other associate advisor. I took three, the most from any professor, courses from him during my graduate study. I enjoyed

his teaching in class and fully mastered the knowledge through his homework questions and computer assignments. His neural network and data mining classes offer me the chance to learn the field of machine learning right in this artificial intelligence era. His linear programming class provides me with a better understanding of assignment problems, which is the foundation of the data association problem presented in this dissertation. He provides me a chance to work on research problems using machine learning techniques, which help a lot in my job search.

It is my great honor to work with the above three UCONN ECE System rockstar professors for my research. My sincere thanks also go to Professor Robert Lynch, Professor Peter Luh, Professor Shengli Zhou and Professor Shalabh Gupta from Department of ECE and Professor Elizabeth Schifano from Department of Statistics for their time and help in teaching me the knowledge required to complete my dissertation.

I also want to extend my thanks to Lance Kaplan and Jemin George for hosting me at the US Army Research Lab in the summer of 2014. They have been offering continuous guidance and feedback to my work on the multiple transient emitter localization problem from the beginning.

I would like to thank my best friend Zhiheng Xu and his wife Bing Yan, who not only help pass my Resume to my advisor and but also make my adaption to life in United States so simple.

I want to thank my current and past labmates Qin Lu, Kaipei Yang, Radu Visina, Katherine Domrese, Xin Zhang, Shuo Zhang, Richard Osborne, Xiufeng Song, Ramona Georgescu, Sora Choi, Ting Yuan, Kevin Romeo and Djedjiga Belfadel; fellow students Yaowen Yu, Weiji Han, Yujia Li, Danxu Zhang, Ashwin Billava, Adam Markman and Pujitha Mannaru; colleagues Xiao Xiao, Marcus Baum, Karl Granstrom and Balakumar Balasingam for helping and working with me.

I would like to thank my grandfathers Guangshun Dou and Chuanpu Jiang, my parents Chunshan Dou and Longhua Jiang, my girlfriend Xiaoyan Tang and all the other family members for their love and support.

Last but not least, I want to thank the financial support by ARO Grant W991NF-10-1-0369.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

This dissertation considers two topics motivated by situational awareness required for unmanned aircraft systems (UAS) and battlefield soldiers, respectively. The first topic is UAS collision warning and the second topic is multiple acoustic transient emitter localization.

1.1.1 UAS collision warning

In recent years, UAS have been increasingly deployed for military, civilian and commercial applications, such as environmental monitoring, law enforcement, and disaster relief instead of manned aircraft because these applications are usually dull, dirty and/or dangerous, thus posing risks of losing lives to human pilots. At present, UAS are operated in segregated volumes of the national airspace system (NAS) to protect civilian aircraft

operating in the surrounding airspace because UAS cannot autonomously sense and avoid (SAA) other UAS, manned aircraft or obstacles, such as buildings. SAA capabilities — the capability of an unmanned aircraft to remain a safe distance from and to avoid collisions with other airborne aircraft — are crucial for UAS to be seamlessly integrated into the NAS.

An SAA system enables UAS to sense the surroundings to be aware of potential hazardous conflicts from other targets. For each target, the UAS has to evaluate the risk of near miss or collision. In the case of a collision risk, the UAS SAA system needs to determine an appropriate avoidance maneuver. Eventually, the UAS will return to its planned course after such a risk is mitigated.

The Traffic Alert and Collision Avoidance System (TCAS), which relies on Automatic Dependent Surveillance - Broadcast (ADS-B), is a well proven *cooperative* sense and avoid technology for aircraft. Aircraft equipped with TCAS transponders have access to each other's information, such as position, altitude, speed and course. However, non-cooperative targets, which do not broadcast the above information, could lead to a catastrophic incident.

Non-cooperative sensing modalities that have been investigated in the literature include monostatic radar, lidar, electro-optical systems, acoustic systems and infrared sensors. In general, radar has the advantages of long sensing range and being able to work in all weather conditions; however, it is limited by the constrained size, weight and power (SWaP) and cost requirements on UAS.

Bistatic or multistatic radar systems provide a possible miniaturized solution if we use ground-based transmitters and equip UAS with receivers. Low cost is also achievable if we adopt inexpensive antenna arrays for the transmitters, in which case the angular resolution is poor and the bistatic range and range rate of the target of interest are the only measurements available. This dissertation develops an algorithm for estimating the closest point

of approach (CPA) between the ownship and the intruder and a collision warning scheme using only bistatic range and range rate measurements from a multistatic radar.

1.1.2 Multiple acoustic transient emitter localization

Hostile fire localization using soldier-wearable acoustic gunfire detection systems [31] is a critical component of situational awareness and threat assessment in the asymmetric battle field of future conflicts. The ability to accurately localize sources of hostile fire in difficult terrain or urban environments is vital for the survivability of dismounted soldiers and plays a crucial role in timely counter-fire.

An acoustic transient emitter can be any source that yields a short-lived acoustic event. This dissertation considers the problem of multiple acoustic transient emitter (target) localization using a network of passive sensors. It is assumed that the targets are stationary during the time window of interest and the number of the targets is unknown. A sensor can be a single microphone or a microphone array. The sensors can measure line of sight (LOS) angles to the emitters by detecting acoustic signals and record the time of arrivals of the detected signals. Missed detections can occur due to sensors' limited sensing range or obstacles along the LOS. False alarms can result from reflected sounds or other acoustic events in the environment.

Existing works on acoustic transient emitter localization have mainly focused on a single emitter. In a multiple emitter scenario, if the acoustic events are separated significantly in time, measurements from the same event will be close in time whereas measurements from different events will be separated significantly in time. Therefore, the targets can be localized one by one, i.e., localization of a single emitter is performed multiple times.

A more challenging situation that has not been investigated, but is considered in this

dissertation, is that the acoustic events are close in time. In this case, we assume that it is not possible to identify the targets directly based on the measurements, which is the case most of the time in real applications. This is because the target-measurement association is unknown, that is, each sensor does not know from which target (or clutter) a particular measurement originates. Before estimating the position of any target, one has to associate the measurements (that are from a common target) from all the sensors. Therefore, the quality of target-measurement association is critical to the overall localization performance.

Two types of multi-sensor processing configurations are considered in this dissertation. In a centralized configuration, the individual sensors report their measurements to a central node (fusion center) over a communication network. At the central node, the individual sensor measurements are fused to yield the final solution (estimation of the number of targets and the target positions), which is then relayed back to individual sensors for their own knowledge. In a distributed configuration, there is no central node and the sensors are connected such that there is a (possibly multi-hop) path between any two sensors. The sensors will gradually reach consensus on the final solution relying solely on one-hop communication of measurements and intermediate solutions.

1.1.3 List of publications to date

Journal publications:

1. Wenbo Dou, Yaakov Bar-Shalom and Peter Willett

   Bistatic measurement fusion from multistatic configurations for air collision warning. Journal of Advances in Information Fusion, 10, (2): 163–182, Dec. 2015.

2. Wenbo Dou, Yaakov Bar-Shalom, Lance Kaplan and Jemin George

Distributed fusion algorithm for passive localization of multiple transient emitters.
Journal of Advances in Information Fusion, accepted for publication in March 2017.

3. Wenbo Dou, Jemin George, Lance Kaplan, Richard W. Osborne, and Yaakov Bar-Shalom
   Evaluation of fusion algorithms for passive localization of multiple transient emitters.
   Journal of Advances in Information Fusion, accepted with minor revision in May 2017.

Conference publications:

1. Wenbo Dou, Jemin George, Lance Kaplan, Richard W. Osborne and Yaakov Bar-Shalom
   Assignment and EM approaches for passive localization of multiple transient emitters.
   In Proceedings of SPIE #9842, Signal Processing, Sensor/Information Fusion, and Target Recognition XXV, May 2016.

2. Wenbo Dou, Peter Willett and Yaakov Bar-Shalom
   Configuration selection for fusion of range and Doppler measurements from multistatic radars for air collision warning.
   In Proceedings of 18th International Conference on Information Fusion, July 2015.

3. Wenbo Dou, Peter Willett and Yaakov Bar-Shalom
   Fusion of range-only measurements from multistatic configurations for air collision warning.
   In Proceedings of IEEE Aerospace Conference, March 2015.

4. Wenbo Dou, Yaakov Bar-Shalom and Peter Willett, and Xiufeng Song

   Initialization and tracking using Doppler-biased multistatic time-of-arrival measurements with linear frequency modulated waveforms

   In Proceedings of 17th International Conference on Information Fusion, July 2014.

5. Wenbo Dou, Yaakov Bar-Shalom and Peter Willett

   Tracking filter initialization with Doppler-biased multistatic time-of-arrival measurements.

   In Proceedings of SPIE #9092, Signal and Data Processing of Small Targets, June 2014.

1.2 Literature review

1.2.1 Related collision warning algorithms

There have been numerous works on the UAS collision avoidance problem [2]. Most have emphasized avoidance algorithms [10] [53], while sensing and estimation methods have been less extensively explored. In [37], a monostatic radar configuration in a two-dimensional (2-D) plane with range and bearing measurements is considered for collision avoidance. In [58], collision warning in a 2-D plane using a monostatic radar with range and azimuth measurements is discussed. A confidence ellipsis at a given time instant is mathematically derived and a confidence corridor is constructed by the regions covered by all confidence ellipses at all time instants within a time interval of interest. A warning decision is based on whether any target aircraft falls within this confidence corridor. The collision warning problem in a multistatic radar configuration has not yet been reported in

the literature.

Target localization is possible using a multistatic radar with time of arrival (TOA) measurements [11] [39] [54] [57]. In [54], target localization is considered in a multistatic ultra wideband radar. The problem is formulated as estimation of target position, which is solved using three methods. Taylor series method is shown to have smaller estimation errors than either least-squares or spherical-interpolation method in a system with one transmitter and four receivers. In [39], two methods are presented to estimate the position of a target in a multistatic passive radar. The spherical-intersection method is shown to be better than the spherical-interpolation method in a system with four transmitters and one receiver. In [11], target localization is investigated in a multistatic passive radar system with one receiver when the receiver position is subject to random errors. An approximated maximum likelihood optimization problem is formulated and solved by a semidefinite relaxation combined with bisection method. In [57], target localization based on both time of arrival and angle of arrival measurements in a multistatic radar system is formulated and a weighted least square method is proposed to estimate the target location.

1.2.2 Related data association algorithms

Related algorithms in the centralized processing configuration

The problem of data association has been studied extensively in tracking multiple targets. Methods including multiple hypothesis tracking [7], joint probabilistic data association filter [30] and probability hypothesis density filter [38] are recursive algorithms that require persistent measurements and provide solutions to a dynamic data association problem. Therefore, they cannot be employed to solve the static data association prob-

lem considered in the situation of multiple acoustic transient emitter localization as in this dissertation.

There are two different philosophies — hard data association and soft data association (see [5], Section 2.4.3) — in solving the static data association problem considered in this dissertation. Hard data association either assigns a measurement to one and only one target or condemn it as a false alarm, in other words, the probability of a measurement coming from a target is either 0 or 1 (discrete). In contrast, soft data association assigns the event that a measurement originates from a target to a (continuous) probability, which can be any value between 0 and 1.

The hard data association for S lists of measurements with one list from each sensor[1], assuming a Bernoulli target-originated measurement model (BTMM) that the number of measurements from each target received at each sensor is a Bernoulli random variable with parameter (mean) equal to the probability of detection, leads to an S-dimensional (S-D) assignment problem, which can be formulated as a discrete constrained optimization problem aiming to find out the set of S-tuples of measurements that minimizes the overall association cost. The number of possible S-tuple set for T targets and S sensors in the absence of missed detections and false alarms is $(T!)^{S-1}$, from which it can be seen that S-D assignment problem is non-deterministic polynomial-time hard with $S \geq 3$. Therefore, it is of great interest and importance to find robust suboptimal algorithms.

The Lagrangian relaxation based approach [18], which is termed as the S-D algorithm in this dissertation, provides a measure of how close the final solution is to the (unknown) optimal solution in terms of the association cost. The application of the S-D assignment algorithm on a multiple transient emitter localization problem using a small number of

[1]In a multi-sensor localization application, as in this dissertation, the number of lists is the same as the number of sensors.

sensors was presented in [46]. Although it does not explore the entire space of the S-tuple sets, it needs to calculate the cost of all candidate S-tuples. The cost calculation involves finding the maximum likelihood estimate of the target locations and can take most of the computational time. The number of candidate S-tuples for T targets and S sensors in the absence of missed detections and false alarms is T^S, which increases exponentially with the number of sensors. Since more sensors generates more accurate estimates in the fusion center, computationally efficient algorithms are required when a large number of sensors are deployed.

The S_0-D+SEQ(2-D) algorithm [61], which performs the S-D assignment algorithm on S_0 lists of measurements before applying the modified Auction algorithm [48] for 2-D assignments on the remaining lists sequentially $S - S_0$ times, is a more efficient algorithm than the S-D assignment. The number of candidate associations increases quadratically (rather than combinatorially/exponentially) with the number of sensors. Because of the ghosting problem [5], the S_0-D step requires, in general, at least 3 lists to achieve reliable association.

In [33], the problem of multiple emitter localization using multiple sensors is formulated as a cardinality (number of targets) selection problem which assumes a Poisson target-originated measurement model (PTMM) that the number of measurements from each target received at each sensor is a Poisson random variable with parameter (mean) equal to the probability of detection. The measurements at a single sensor from all targets and the clutter are modeled as a Poisson point process (PPP) [17]. For each possible selected cardinality, one solves a sub-problem based on the learning expectation-maximization (EM) algorithm [19] to select the best cardinality based on an information criterion [1] [51]. During every iteration of the EM algorithm, each measurement will be assigned a probability of having originated from a target, which is an example of the soft data association.

Related algorithms in the distributed processing configuration

Some related problems have been studied in the distributed configuration. The average consensus problem that individual sensors eventually obtains the average of initial values stored in all sensors have been extensively studied. The fastest linear iteration was investigated in [59] and a solution robust to unreliable communication links was presented in [60].

The problem of a single emitter localization, which is equivalent to an nonlinear optimization problem, can be solved in a distributed manner by reformulating the problem into an equivalent linear-equality-constrained nonlinear optimization and solving it using the alternating direction method of multipliers (ADMM) algorithm. [50] and [42] present a bridge-sensor based distributed ADMM and a coloring-scheme based distributed ADMM, respectively.

Distributed EM algorithms have attracted a lot of attentions in sensor network applications for density estimation, data clustering and target tracking. An incremental distributed EM algorithm presented in [44] is the first known scheme for density estimation and clustering in distributed sensor network. A distributed EM algorithm based on the averaging consensus filtering was developed in [34] for particle filter based target tracking. A distributed EM algorithm based on ADMM was proposed in [29] for distributed data clustering.

TABLE 1.3.1: Target motion parameter observability summary.

Scenario	Sensor configuration	Number of transmitters	Observability
2-D	bistatic	1	marginally observable
2-D	multistatic	2	observable
3-D	bistatic	1	unobservable
3-D	multistatic	2	marginally observable
3-D	multistatic	3	observable

1.3 Contribution and Methodology

1.3.1 UAS collision warning

The collision warning problem in a multistatic radar configuration has not yet been reported in the literature. In this dissertation, a strategy for collision warning in a three-dimensional (3-D) space was presented, assuming a constant velocity motion of an aircraft of interest (target/intruder), to estimate the position and velocity of the target so as to predict the CPA. Several configurations listed in Table 2.1.1 are investigated from a parameter observability point of view. In a 2-D scenario, it is assumed that the target is at the same altitude as the ownship. Two transmitters in a 2-D multistatic configuration with the same-altitude assumption and a 3-D multistatic configuration with three transmitters are practically useful configurations.

The collision warning is formulated as a hypothesis testing problem using a generalized likelihood function. Monte Carlo simulation shows the likelihood-based collision warning algorithm using three transmitters has no missed detection of a collision and has no false alarms when the intruder and ownship altitude separation is beyond 100 m. The likelihood-based collision warning algorithm using two transmitters with the same-altitude assumption has no missed detection of a collision, either. It is, however, prone to false alarms when the

CPA angle is near 180^{o}.

Since air collision is deadly, no missed detections can be tolerated. It is also necessary to account for the physical dimensions of aircraft by adding a safety margin to compensate for the errors arising in the point modeling of aircraft. Simulation results show that the likelihood-based collision warning algorithm with a safety margin of 100 m has no missed detections of collision but becomes more conservative with false alarms occurring in more situations.

A second, Bayesian, approach formulates the CPA distance as a random variable and estimates its probability density function (pdf) as a fitted Rician distribution. Then it defines the collision event by considering the physical dimensions of the aircraft and calculates the probability of collision, based on which a warning decision can be made. The performance of the Bayesian collision warning algorithm is consistent with that of a likelihood-based algorithm.

1.3.2 Multiple acoustic transient emitter localization

In this dissertation, we develop passive sensor fusion algorithms in both centralized and distributed multi-sensor processing configurations.

Centralized algorithms

For the centralized configuration, we considered both BTMM and PTMM as possible measurement models.

BTMM leads to a multi-dimensional assignment problem formulation. In [46], the S-D assignment algorithm was developed for a small number of sensors. Since this algorithm is no longer computationally efficient when a large number of sensors are used, we develop

a sequential m-best 2-D assignment (Smb) algorithm. The time of arrival measurements play a critical role in avoiding ghost targets which commonly occur when only LOS measurements are associated.

PTMM leads to a cardinality selection problem formulation. In [33], both the range and bearing measurements are assumed available and the initialization in the EM-based algorithm uses a finite set including target locations that are close to the truth. Since the range measurement and prior information for a "good" initialization is not always available in the real world, we develop a measurement-driven initialization approach relying on LOS and time of arrival measurements in order for the centralized EM algorithm to converge to the maximum likelihood solution.

The developed algorithms will be evaluated using the data generated based on BTMM, which is a more realistic model.

Distributed algorithms

For the distributed configuration, we solve the localization problem of a single emitter before we develop a distributed algorithm for the case of multiple emitters. We developed two types of distributed algorithms for single emitter localization. One results from a distributed implementation of an iterative least squares algorithm based on average consensus. The other is an average consensus based distributed ADMM algorithm which results from a constrained nonlinear optimization formulation.

For the multiple emitter case, we only consider the PTMM as the measurement model. We develop a distributed EM algorithm to solve the data association and position estimation problems. Existing distributed EM algorithms have assumed a *linear* generative model for their respective applications, which does not apply to a *nonlinear* generative model due

to the incomplete position measurement being bearings and time of arrivals in the emitter localization scenario considered in this dissertation. Moreover, the parameters in existing distributed EM algorithms are initialized to be either fixed values or random values. Such an initialization approach can not guarantee the convergence to the maximum likelihood solution. Similarly as in the centralized EM algorithm, a measurement-driven initialization is presented. We extend the average consensus problem from one value at each sensor to the case that each sensor has a different number of values. We developed a distributed set consensus algorithm based on track association to obtain the same initialization at each sensor for the distributed EM algorithm.

Chapter 2

Bistatic Measurement Fusion from Multistatic Configurations for Air Collision Warning

2.1 Introduction

Sense-and-Avoid (SAA) capabilities are required for unmanned aircraft systems (UAS) to operate within the national airspace, since the proliferation of UAS has increased the risk of aircraft collision. The air traffic control radar beacon system works well to coordinate cooperative aircraft. Active sensing methods have to be employed for UAS to be functional against non-cooperative targets. The limitations on the size, weight and power of UAS suggest an implementation with ground-based transmitters and UAS-mounted receivers.

There have been numerous works on the UAS collision avoidance problem [2]. Most have emphasized avoidance algorithms [10] [53] [55], while sensing and estimation methods have been less extensively explored. In [37], a monostatic radar configuration in a

two-dimensional (2-D) plane with range and bearing measurements is considered for collision avoidance. In [58], collision warning in a 2-D plane using a monostatic radar with range and azimuth measurements is discussed. A confidence ellipsis at a given time instant is mathematically derived and a confidence corridor is constructed by the regions covered by all confidence ellipses at all time instants within a time interval of interest. A warning decision is based on whether any target aircraft falls within this confidence corridor. The collision warning problem in a multistatic radar configuration has not yet been reported in the literature.

Target localization is possible using a multistatic radar with time of arrival (TOA) measurements [11] [39] [54] [57]. In [54], target localization is considered in a multistatic ultra wideband radar. The problem is formulated as estimation of target position, which is solved using three methods. Taylor series method is shown to have smaller estimation errors than either least-squares or spherical-interpolation method in a system with one transmitter and four receivers. In [39], two methods are presented to estimate the position of a target in a multistatic passive radar. The spherical-intersection method is shown to be better than the spherical-interpolation method in a system with four transmitters and one receiver. In [11], target localization is investigated in a multistatic passive radar system with one receiver when the receiver position is subject to random errors. An approximated maximum likelihood optimization problem is formulated and solved by a semidefinite relaxation combined with bisection method. In [57], target localization based on both time of arrival and angle of arrival measurements in a multistatic radar system is formulated and a weighted least square method is proposed to estimate the target location. TOA measurements can be used to estimate the position but not the velocity, range rate measurements are needed for the velocity.

In our previous work [26] [27], a strategy for collision warning in a three-dimensional

TABLE 2.1.1: Target motion parameter observability summary.

Scenario	Sensor configuration	Number of transmitters	Observability
2-D	bistatic	1	marginally observable
2-D	multistatic	2	observable
3-D	bistatic	1	unobservable
3-D	multistatic	2	marginally observable
3-D	multistatic	3	observable

(3-D) space was presented, assuming a constant velocity motion of an aircraft of interest (target/intruder), to estimate the position and velocity of the target so as to predict the closest point of approach (CPA). Since an inexpensive system is the goal, only bistatic range and range rate measurements are available. Several configurations listed in Table 2.1.1 are investigated from a parameter observability point of view. In general 3-D scenarios, the target motion parameter is shown to be unobservable in a bistatic configuration (that is: one transmitter and one receiver, not co-located) and a change of course of the receiver (the "observability platform maneuver" that is the saving grace for angle-only target motion analysis (TMA)) merely improves the observability marginally. In a multistatic configuration, one has marginal observability using two transmitters, but good observability with three. In a 2-D scenario which assumes that the target is at the same altitude as the ownship, the target parameter is still only marginally observable in a bistatic configuration. The observability is improved by a small maneuver of the ownship but it is still unappealing. On the other hand, one can have very good observability of the target motion parameter with two transmitters in a 2-D multistatic configuration with the same-altitude assumption, which turns out to be another practically useful configuration in addition to a 3-D multistatic configuration with three transmitters. Simulation results and comparison with the CRLB show that the ML estimate of the target parameter can be considered as statistically efficient in both useful configurations.

The collision warning is formulated as a hypothesis testing problem using a generalized likelihood function. Monte Carlo simulation shows the likelihood-based collision warning algorithm using three transmitters has no missed detection of a collision and has no false alarms when the intruder and ownship altitude separation is beyond 100 m. The likelihood-based collision warning algorithm using two transmitters with the same-altitude assumption has no missed detection of a collision, either. It is, however, prone to false alarms when the CPA angle is near 180^{o}.

This chapter extends the previous work [26] [27] by (i) taking the physical dimensions of aircraft into consideration in the likelihood-based collision warning algorithm; (ii) investigating the statistical efficiency of the closest point of approach (CPA) time estimate in the likelihood-based collision warning algorithm; (iii) adding a Bayesian approach for collision warning. Since air collision is deadly, no missed detections can be tolerated. It is also necessary to account for the physical dimensions of aircraft by adding a safety margin to compensate for the errors arising in the point modeling of aircraft. Simulation results show that the likelihood-based collision warning algorithm with a safety margin of 100 m has no missed detections of collision but becomes more conservative with false alarms occurring in more situations.

The likelihood based collision warning algorithm makes decisions by using an estimated CPA time, and Monte Carlo simulations have shown that the CPA time estimate can be considered as unbiased and statistically efficient for the simulated scenarios. The Bayesian approach formulates the CPA distance as a random variable and estimates its probability density function (pdf) as a fitted Rician distribution. Then it defines the collision event by considering the physical dimensions of the aircraft and calculates the probability of collision, based on which a warning decision can be made. The performance of the Bayesian collision warning algorithm is consistent with that of a likelihood-based

algorithm.

The remaining sections of this chapter are organized as follows. Section 2.2 describes and formulates the general 3-D problem and considers a special 2-D problem. Section 2.3 analyzes several possible configurations for collision warning including both 3-D and 2-D scenarios and shows that two of them seem to be practically useful. Section 2.4 presents the ML estimator based on which two different collision warning algorithms are described in Section 2.5. Section 2.6 investigates the efficiency of the ML estimator of the target motion parameter and the efficiency of the CPA time estimate used in the likelihood-based collision warning algorithm, and also shows the performances of both collision warning algorithms and Section 2.7 draws conclusions.

2.2 Problem Formulation

Assume a target of interest (intruder) is moving in 3-D with a constant velocity. The 3-D target position in Cartesian coordinates at time k is therefore

$$\boldsymbol{\xi}(\mathbf{x}, k) = \mathbf{x}_0 + kT\dot{\mathbf{x}}_0 \quad k = 0, 1, \ldots \tag{2.2.1}$$

where

$$\mathbf{x} = [\mathbf{x}_0', \dot{\mathbf{x}}_0']' = [x, y, z, \dot{x}, \dot{y}, \dot{z}]' \tag{2.2.2}$$

is the unknown target motion parameter which is a vector of dimension $n_{\mathbf{x}} = 6$ consisting of the target's position $\mathbf{x}_0$ and velocity $\dot{\mathbf{x}}_0$ in Cartesian coordinates at time $k = 0$ (or without loss of generality at any chosen reference time); and T is the sampling period. There are

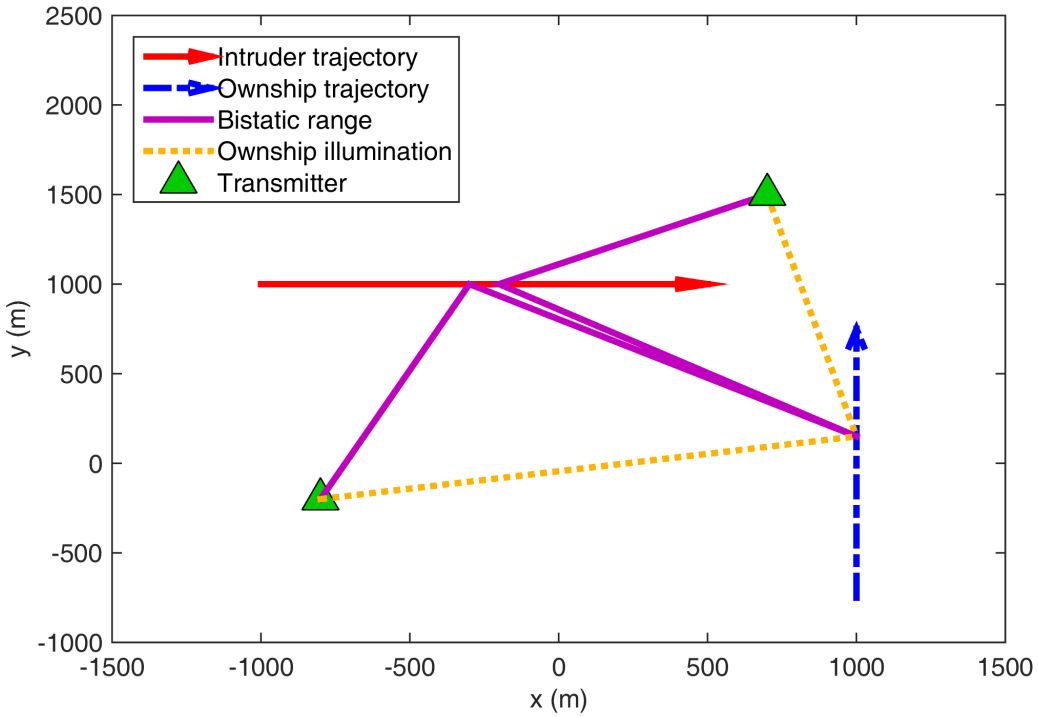


FIGURE 2.2.1: A multistatic configuration in the X-Y plane. The time differences of arrival (actual measurements) between the direct path (ownship illumination) and the indirect path (bistatic range) multiplied by the speed of light is added to the direct path distance to yield an equivalent bistatic range measurement.

N_{Tx} ($N_{\mathrm{Tx}} \geq 1$) transmitters at known locations $\mathbf{u}_i = [x_{u_i}, y_{u_i}, z_{u_i}]', i = 1, \ldots, N_{\mathrm{Tx}}$. At time k ($k > 0$), a moving receiver (the ownship) with known position $\mathbf{s}(k)$ and velocity $\dot{\mathbf{s}}(k)$ can obtain measurements consisting of the bistatic range [8] illustrated in Figure 2.2.1 and the bistatic range rate from the ith transmitter located at $\mathbf{u}_i$ given by

$$\mathbf{z}_i(k) = \mathbf{h}_i(\mathbf{x}, k) + \mathbf{w}_i(k) \quad i = 1, \ldots, N_{\mathrm{Tx}} \tag{2.2.3}$$

where

$$\mathbf{h}_i(\mathbf{x},k) = \begin{bmatrix} r_i(k) \\ \dot{r}_i(k) \end{bmatrix}$$
$$= \begin{bmatrix} \|\boldsymbol{\xi}(\mathbf{x},k) - \mathbf{s}(k)\| + \|\boldsymbol{\xi}(\mathbf{x},k) - \mathbf{u}_i\| \\ \dfrac{[\boldsymbol{\xi}(\mathbf{x},k) - \mathbf{s}(k)]'\,[\dot{\mathbf{x}}_0 - \dot{\mathbf{s}}(k)]}{\|\boldsymbol{\xi}(\mathbf{x},k) - \mathbf{s}(k)\|} + \dfrac{[\boldsymbol{\xi}(\mathbf{x},k) - \mathbf{u}_i]'\,\dot{\mathbf{x}}_0}{\|\boldsymbol{\xi}(\mathbf{x},k) - \mathbf{u}_i\|} \end{bmatrix} \tag{2.2.4}$$

and $\mathbf{w}_i(k)$ are the measurement noises, assumed to be independent and identically distributed zero-mean white Gaussian sequences with known covariance matrix

$$R_i = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{\dot{r}}^2 \end{bmatrix} \tag{2.2.5}$$

The measurement function comprising all the measurements at time k is

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x},k) + \mathbf{w}(k) \quad k = 1,\ldots \tag{2.2.6}$$

where

$$\mathbf{z}(k) = [\mathbf{z}_1(k)' \ldots \mathbf{z}_{N_{\mathrm{Tx}}}(k)']' \tag{2.2.7}$$

$$\mathbf{h}(\mathbf{x},k) = [\mathbf{h}_1(\mathbf{x},k)' \ldots \mathbf{h}_{N_{\mathrm{Tx}}}(\mathbf{x},k)']' \tag{2.2.8}$$

$$\mathbf{w}(k) = [\mathbf{w}_1(k)' \ldots \mathbf{w}_{N_{\mathrm{Tx}}}(k)']' \tag{2.2.9}$$

and

$$R(k) = E[\mathbf{w}(k)\mathbf{w}(k)'] = \begin{bmatrix} R_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & R_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & R_{N_{\mathrm{Tx}}} \end{bmatrix} \tag{2.2.10}$$

Since both intruder and ownship are moving, it is important to avoid any collision between them. The goal is thus to estimate the target parameter $\mathbf{x}$ based on N frames of measurements, and to deliver a warning long enough and confidently enough before a possible collision occurs so as to predict the CPA and, presumably, to do something about it if needed.

2.2.1 Parameter Observability

We need to check the observability of the target motion parameter (2.2.2) to see whether there is sufficient information in the data. Observability requires the invertibility of the Fisher information matrix (FIM), which is given by [5]

$$J = E\left\{[\nabla_{\mathbf{x}} \ln \Lambda(\mathbf{x}; \mathbf{Z})][\nabla_{\mathbf{x}} \ln \Lambda(\mathbf{x}; \mathbf{Z})]'\right\}|_{\mathbf{x}=\mathbf{x}_{\mathrm{t}}} \tag{2.2.11}$$

where $\Lambda(\mathbf{x}; \mathbf{Z})$ is the likelihood function of the parameter based on the measurement set

$$\mathbf{Z} = \mathbf{z}(k)_{k=1}^{N} \tag{2.2.12}$$

and $\mathbf{x}_{\mathrm{t}}$ is the true value of the target motion parameter. In a simulated scenario, $\mathbf{x}_{\mathrm{t}}$ is known. However, in a real scenario where $\mathbf{x}_{\mathrm{t}}$ is unknown and needs to be estimated, the

FIM is evaluated at the estimate.

Since the measurement noises are assumed to be white, we have

$$\Lambda(\mathbf{x};\mathbf{Z}) = \prod_{k=1}^{N} p\left(\mathbf{z}(k)|\mathbf{x}\right) \tag{2.2.13}$$

where

$$\begin{aligned} p\left(\mathbf{z}(k)|\mathbf{x}\right) &= \left|2\pi R(k)\right|^{-\frac{1}{2}} \\ &\cdot \exp\left(-\frac{1}{2}\left[\mathbf{z}(k)-\mathbf{h}(\mathbf{x},k)\right]' R(k)^{-1}\left[\mathbf{z}(k)-\mathbf{h}(\mathbf{x},k)\right]\right) \end{aligned} \tag{2.2.14}$$

The gradient of the log-likelihood function is

$$\nabla_{\mathbf{x}} \ln \Lambda(\mathbf{x};\mathbf{Z}) = -\sum_{k=1}^{N}\left[\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x},k)'\right] R(k)^{-1}\left[\mathbf{z}(k)-\mathbf{h}(\mathbf{x},k)\right] \tag{2.2.15}$$

Substituting (2.2.15) into (2.2.11) yields

$$\begin{aligned} J &= \sum_{k=1}^{N}\left[\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x},k)'\right] R(k)^{-1}\left[\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x},k)'\right]'\Big|_{\mathbf{x}=\mathbf{x}_{\mathrm{t}}} \\ &= \sum_{k=1}^{N}\sum_{i=1}^{N_{\mathrm{Tx}}}\left[\nabla_{\mathbf{x}}\mathbf{h}_i(\mathbf{x},k)'\right] R_i^{-1}\left[\nabla_{\mathbf{x}}\mathbf{h}_i(\mathbf{x},k)'\right]'\Big|_{\mathbf{x}=\mathbf{x}_{\mathrm{t}}} \end{aligned} \tag{2.2.16}$$

If J is not invertible, then the target motion parameter is unobservable. Otherwise, the size of confidence region for the true target position [5] can be used to distinguish between marginal observability and good observability. In this chapter, marginal and good observability are distinguished from each other by the length of the longest semiaxis of 99.9999% probability region. In the application of air collision warning, one could say that the observability is good if the longest semiaxis is, say, less than 100 meters and

that the observability is marginal if the longest semiaxis is, say, more than 100 meters. Mathematically, the length of the longest semiaxis is proportional to the square root of largest eigenvalue of the covariance matrix in (2.2.20).

2.2.2 Confidence Region in the General Case

Suppose one has an unbiased and statistically efficient estimate $\hat{\mathbf{x}}$, that is

$$E\left[\hat{\mathbf{x}}\right] = \mathbf{x}_{\mathrm{t}} \tag{2.2.17}$$

$$P \triangleq E\left[\left[\hat{\mathbf{x}} - \mathbf{x}_{\mathrm{t}}\right]\left[\hat{\mathbf{x}} - \mathbf{x}_{\mathrm{t}}\right]'\right] = J^{-1} \tag{2.2.18}$$

where J^{-1} is the Cramer Rao lower bound (CRLB). The 3-D target position estimate at an arbitrary time t is

$$\hat{\mathbf{x}}_{\mathrm{p}}(t) = \begin{bmatrix} 1 & 0 & 0 & t & 0 & 0 \\ 0 & 1 & 0 & 0 & t & 0 \\ 0 & 0 & 1 & 0 & 0 & t \end{bmatrix} \hat{\mathbf{x}} \triangleq \Phi_{\mathrm{p}}(t)\hat{\mathbf{x}} \tag{2.2.19}$$

and the corresponding covariance is

$$P_{\mathrm{p}}(t) = \Phi_{\mathrm{p}}(t) P \Phi_{\mathrm{p}}(t)' \tag{2.2.20}$$

If one further assumes $\hat{\mathbf{x}}$ is Gaussian, that is,

$$\hat{\mathbf{x}} \sim \mathcal{N}(\mathbf{x}_{\mathrm{t}}, P) \tag{2.2.21}$$

then, because of linear transformation in (2.2.19)

$$\hat{\mathbf{x}}_{\mathrm{p}}(t) \sim \mathcal{N}(\mathbf{x}_{\mathrm{p}}(t), P_{\mathrm{p}}(t)) \tag{2.2.22}$$

The normalized estimation error squared (NEES) for the target position $\mathbf{x}_{\mathrm{p}}(t)$ at t, defined as

$$\epsilon_{\mathrm{p}}(t) = [\mathbf{x}_{\mathrm{p}}(t) - \hat{\mathbf{x}}_{\mathrm{p}}(t)]' \, P_{\mathrm{p}}^{-1}(t) \, [\mathbf{x}_{\mathrm{p}}(t) - \hat{\mathbf{x}}_{\mathrm{p}}(t)] \tag{2.2.23}$$

is chi-square distributed with $n_{\mathbf{x}}/2$ degrees of freedom, that is,

$$\epsilon_{\mathrm{p}}(t) \sim \chi^2_{n_x/2} \tag{2.2.24}$$

Let g be such that

$$P\{\epsilon_{\mathrm{p}}(t) \leq g^2\} = 1 - Q \tag{2.2.25}$$

where Q is a small tail probability. Given the predicted target position $\hat{\mathbf{x}}_{\mathrm{p}}(t)$, the $100(1 - Q)\%$ confidence region [4] for the true position $\mathbf{x}_{\mathrm{p}}(t)$ is defined to be within the ellipsoid given by

$$[\mathbf{x}_{\mathrm{p}}(t) - \hat{\mathbf{x}}_{\mathrm{p}}(t)]' \, P_{\mathrm{p}}^{-1}(t) \, [\mathbf{x}_{\mathrm{p}}(t) - \hat{\mathbf{x}}_{\mathrm{p}}(t)] = g^2 \tag{2.2.26}$$

Alternatively, given the true position $\mathbf{x}_{\mathrm{p}}(t)$, (2.2.26) is also defined to be the $100(1 - Q)\%$ probability region for the predicted target position $\hat{\mathbf{x}}_{\mathrm{p}}(t)$. These two regions as shown in Figure 2.2.2 have identical geometrical sizes since they can be represented by the same

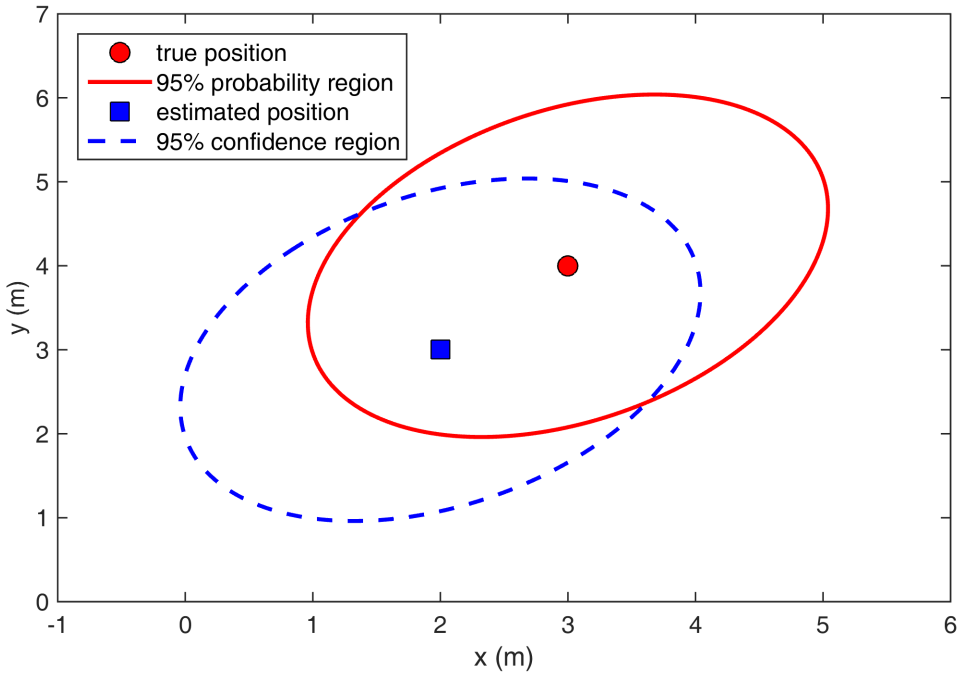


FIGURE 2.2.2: Confidence region and probability region in the X-Y plane. If an estimate is inside 95% probability region around the truth, then the truth must be inside 95% confidence region around this estimate.

equation as in (2.2.26). If either region is large, one has marginal observability of the target position; if any one of the regions is small, one has good observability of the target position.

2.2.3 Confidence Region When Intruder and Ownship at Same Altitude

If the intruder's altitude z is assumed to be known and is equal to that of the ownship, then the 2-D X-Y plane at the altitude z is of interest and everything related to the target can be considered restricted to this 2-D space. Specifically, the target parameter to be estimated

TABLE 2.3.1: Scenario specifications. The last column reflects the results from Section 2.3.

Scenario	Transmitters used	UAS motion	Target altitude	Collision	Semiaxis lengths of 99.9999% probability region (m)
1	Tx1	CV	Unknown	Yes	3×10^9, 2020, 62
2	Tx1	two-segment CV	Unknown	Yes	6468, 1660, 109
3	Tx1 and Tx2	CV	Unknown	Yes	1542, 50, 41
4	Tx1 and Tx2	two-segment CV	Unknown	Yes	1402, 51, 41
5	Tx1,Tx2 and Tx3	CV	Unknown	Yes	50, 42, 11
6	Tx1,Tx2 and Tx3	CV	Unknown	No	48, 43, 12
7	Tx1	CV	Known	Yes	2600, 81
8	Tx1	two-segment CV	Known	Yes	301, 25
9	Tx1 and Tx2	CV	Known	Yes	40, 8

becomes

$$\mathbf{x}^{\mathrm{2D}} = [x, y, \dot{x}, \dot{y}]' \tag{2.2.27}$$

Correspondingly, the 2-D target position at an arbitrary time t is

$$\mathbf{x}_{\mathrm{p}}^{\mathrm{2D}}(t) = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \end{bmatrix} \mathbf{x}^{\mathrm{2D}} \tag{2.2.28}$$

The confidence region for the true target position around its estimate is now an ellipse given by (2.2.26).

2.3 Scenarios and Observability Analysis

From (2.2.26), the size of the confidence region for the true target position around the predicted position is the same as that of the probability region for the predicted target position around the true position. Since it is more convenient for an observability analysis to

obtain the probability region for the predicted target position with the true target motion parameter assumed available than to estimate the true target parameter and obtain the confidence region for it, in this section several scenarios are simulated with the knowledge of the true target motion parameter and the probability region of the estimate in each scenario is obtained without performing any estimation.

A radar system, which consists of three transmitters on the ground and one receiver mounted on an unmanned aircraft system (UAS) — the ownship — is used to warn of a possible collision between the UAS (ownship) and an intruder aircraft. The transmitters are located at (0 m, 1000 m, 0 m), (0 m, – 1000 m, 0 m) and (1000 m, 0 m, 0 m) in Cartesian coordinates, and are denoted by Tx1, Tx2 and Tx3, respectively. The UAS is moving at an altitude of 1500 m.

Eight collision scenarios and one non-collision scenario listed in the Table 2.3.1, differing in the number of transmitters, the motion of the UAS and the dimensionality of target parameter are studied here. Scenarios with the "known target altitude" assumption are referred to as 2-D scenarios. The rest are 3-D scenarios. Two motions of UAS are considered. In a constant velocity (CV) motion, the UAS starts moving from the point (– 4500 m, 0 m, 1500 m) at time $t = 0$ s with a constant velocity $\dot{\mathbf{s}}_0 = [50 \text{ m/s}, 0 \text{ m/s}, 0 \text{ m/s}]'$. In a two-segment CV motion, the UAS starts with a constant velocity $[43 \text{ m/s}, -25 \text{ m/s}, 0 \text{ m/s}]'$ from the point (– 4306 m, 752 m, 1500 m) at time $t = 0$ s for 27 s and then executes a 5° /s coordinated turn for 6 s before changing to another velocity $[50 \text{ m/s}, 0 \text{ m/s}, 0 \text{ m/s}]'$ when it arrives at the location (– 2850 m, 0 m, 1500 m). In all the collision scenarios, the intruder aircraft starts from the position (4500 m, 0 m, 1500 m) at time $t = 0$ s with a constant velocity $\dot{\mathbf{x}}_0 = [-50 \text{ m/s}, 0 \text{ m/s}, 0 \text{ m/s}]'$ and will collide with the UAS at time $t = 90$ s. In the non-collision scenario, the altitude of the intruder aircraft is assumed to be 1600 m, which is 100 m higher than in the collision scenarios, and the CPA occurs at time $t = 90$ s. Bistatic

range and range rate measurements are made from the ownship every 1 s over a period of 60 s, which is 30 s before the CPA time. The noise standard deviations for the range and range rate measurements are assumed to be 8.66 m and 1 m/s, respectively, at all times.

Figures 2.3.1 and 2.3.2 visualize all the 3-D scenarios and plots the 99.9999% probability region, the lengths of the semiaxes of which are also shown in Table 2.3.1, around the collision point or the target CPA in each scenario.

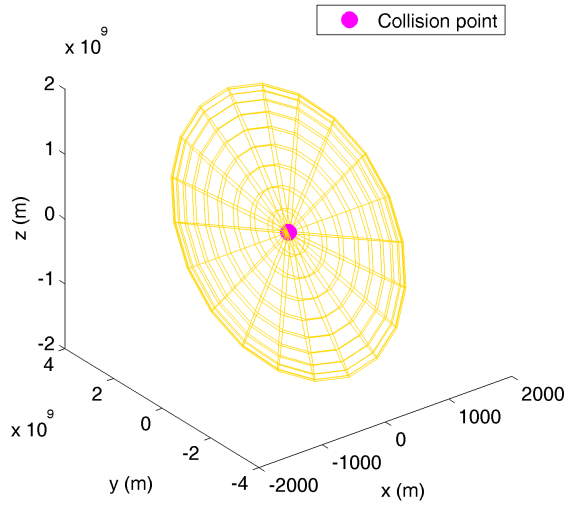
In Scenario 1, the FIM is nearly singular with a condition number[1] of 18.8. The large probability region (which implies a large confidence region) indicates the target parameter is practically unobservable and even an efficient estimator is useless in such a situation.

In Scenario 2, the FIM is not ill-conditioned. The ellipsoid is much smaller than in the first scenario, which indicates the change of course in the ownship trajectory improves the observability. However, the size of the probability (or confidence) region is still quite large so that even an efficient estimator remains practically useless.

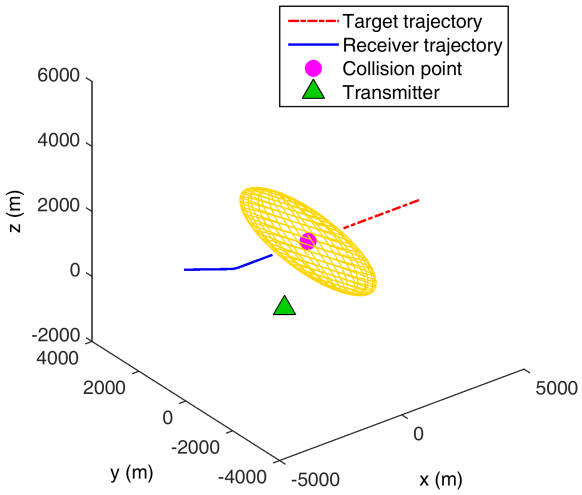
Compared with the 3-D bistatic configuration (Scenarios 1 and 2), adding a second transmitter in Scenarios 3 and 4 reduces the target localization uncertainty, although the size of the probability region is still too large to be useful. Comparison between Figures 2.3.1(c) and 2.3.1(d) indicates that the further reduction of the localization uncertainty resulting from the change of course in the ownship trajectory in the multistatic configuration is not as significant as in the bistatic.

As shown in Figures 2.3.2(a) and 2.3.2(b), the addition of a third transmitter into the multistatic configuration has significantly improved observability, which makes the localization practically useful. Therefore, one needs three transmitters in a 3-D multistatic configuration to build up an efficient estimator based on which a useful collision warning

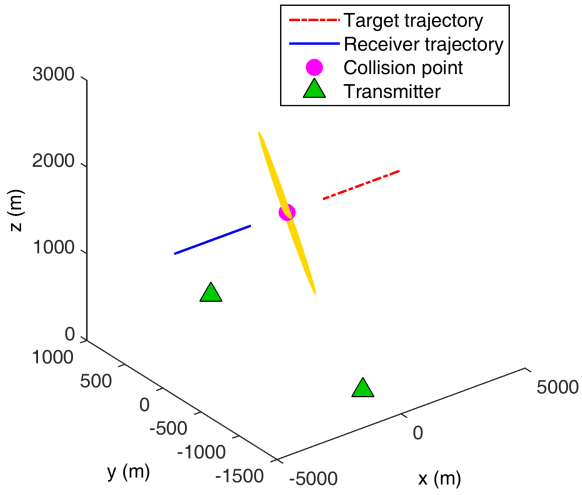
[1]The condition number is $\log_{10} \frac{\lambda_{max}}{\lambda_{min}}$, where λ_{max} and λ_{min} are the largest and smallest eigenvalues of the FIM.



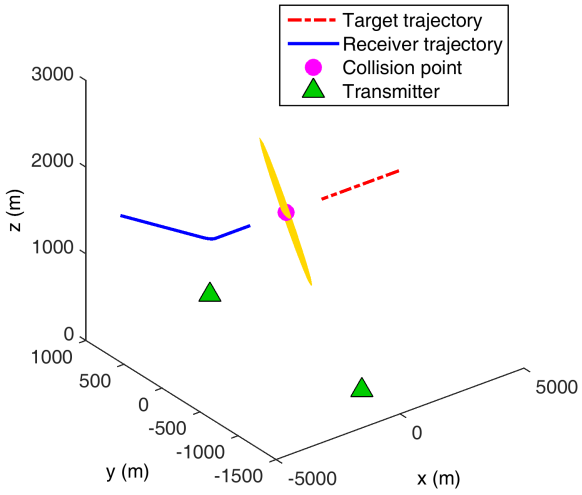
(A) Scenario 1



(B) Scenario 2

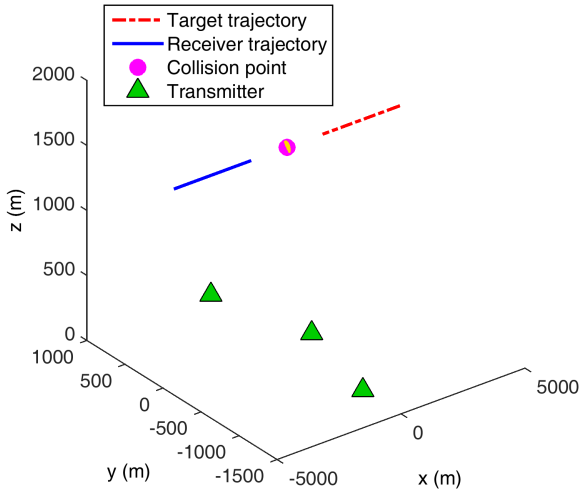


(C) Scenario 3

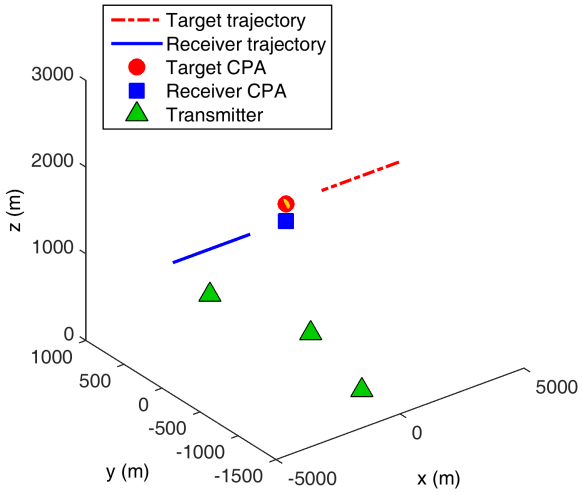


(D) Scenario 4

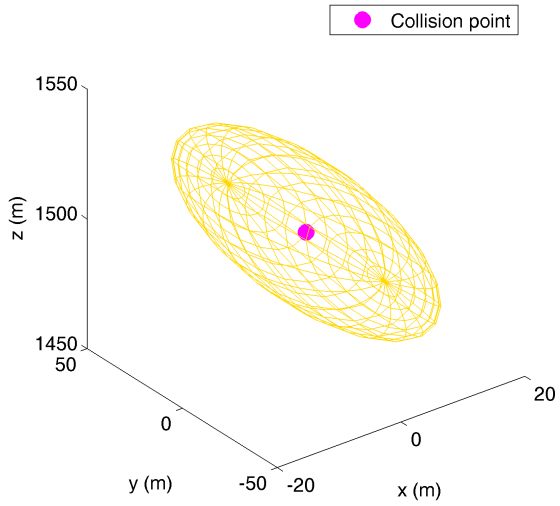
FIGURE 2.3.1: 99.9999% (ellipsoidal) probability region around the collision point in Scenarios 1 to 4. The target motion parameter is practically unobservable in Scenario 1. The target motion parameter is marginally observable in Scenarios 2, 3 and 4.



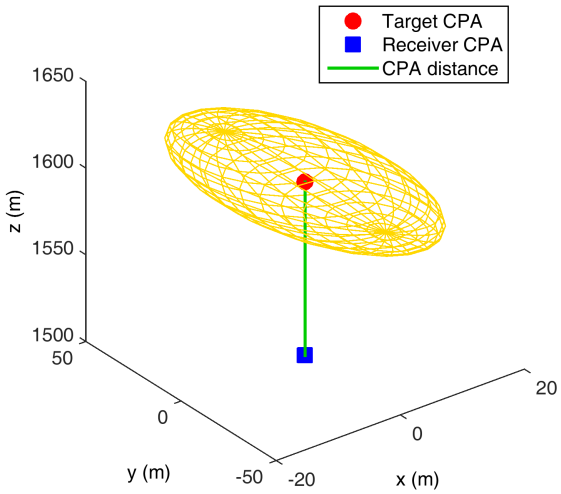
(A) Scenario 5



(B) Scenario 6

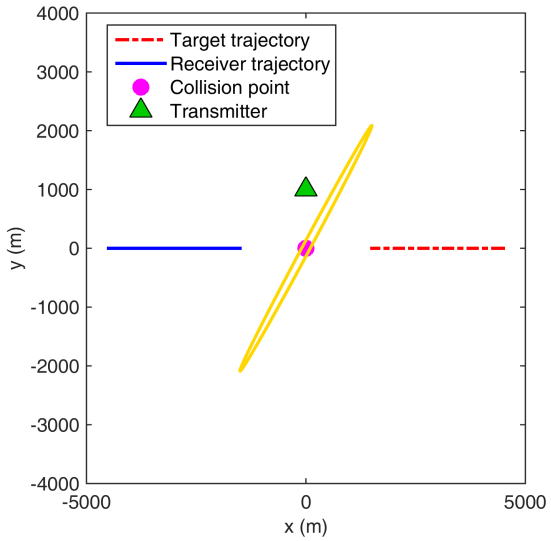


(C) Scenario 5 magnified

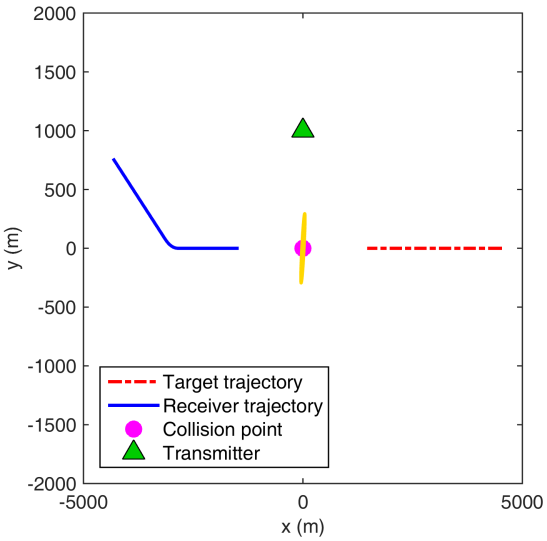


(D) Scenario 6 magnified

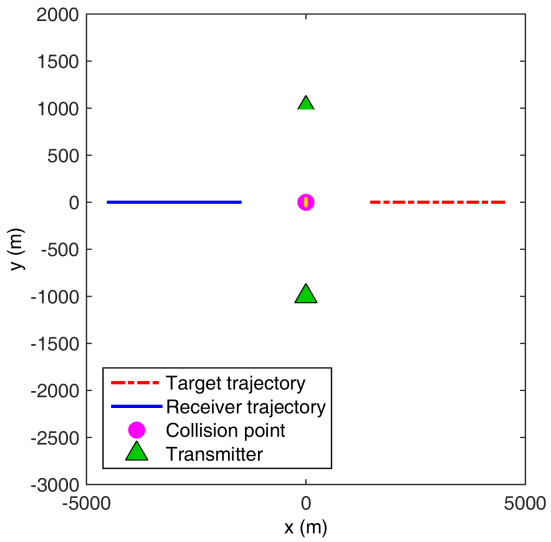
FIGURE 2.3.2: 99.9999% (ellipsoidal) probability region around the collision point or the target CPA in Scenarios 5 and 6. The target motion parameter observability is good in both scenarios.



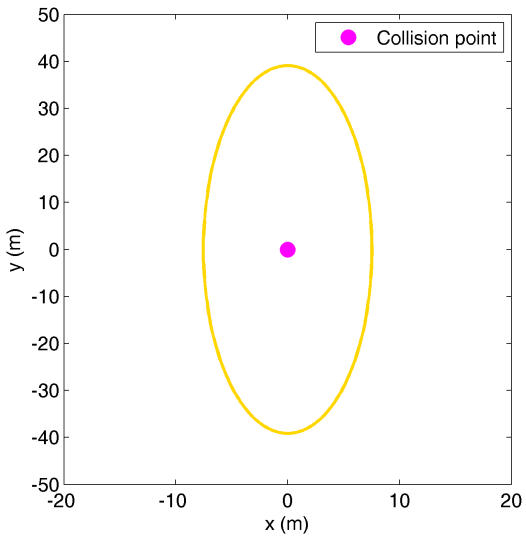
(A) Scenario 7



(B) Scenario 8



(C) Scenario 9



(D) Scenario 9 magnified

FIGURE 2.3.3: 99.9999% (elliptic) probability region around the collision point in 2-D scenarios. The target motion parameter is marginally observable in Scenarios 7 and 8. The target motion parameter observability is good in Scenario 9.

algorithm can be designed.

Figure 2.3.3 visualizes all the 2-D scenarios and plots the 99.9999% probability region around the collision point in each scenario. Compared with 3-D scenarios, the knowledge of target altitude in a 2-D scenario results in a significant reduction in the uncertainty. In Scenario 7, the size of the probability region is still too large to be useful. In Scenario 8, the probability region could be useful, however, it is due to the change of course of the ownship and this maneuver action itself could lead a safety situation to a dangerous collision. In Scenario 9, adding a second transmitter reduces the target localization uncertainty significantly. The size of this region is practically useful. Therefore, with the knowledge of the target altitude one needs two transmitters in a multistatic configuration to build up an efficient estimator based on which a useful collision warning algorithm can be designed.

In the sequel, collision warning is only considered in those two practically useful configurations — 3 transmitters in general 3-D scenarios and 2 transmitters with known target altitude in 2-D scenarios, corresponding to Scenarios 5, 6 and 9.

2.4 The Maximum Likelihood Estimator

The ML estimate of the target motion parameter $\mathbf{x}$ in (2.2.2) is

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg\max_{\mathbf{x}} \Lambda(\mathbf{x}; \mathbf{Z}) \tag{2.4.1}$$

where $\Lambda(\mathbf{x}; \mathbf{Z})$ is given in (2.2.13). The ILS technique [4] was used to find the ML estimate in this case. If we set (2.2.15) to zero, we will notice that there is no closed-form solution. Using a first order series expansion about an estimate $\hat{\mathbf{x}}^j$ at the end of the j-th iteration

leads to an iterative scheme and the $(j+1)$-th estimate is

$$\hat{\mathbf{x}}^{j+1} = \hat{\mathbf{x}}^{j} + \left[(H^{j})' R^{-1} H^{j}\right]^{-1} (H^{j})' R^{-1} \left[\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}^{j})\right] \tag{2.4.2}$$

where

$$\mathbf{z} = [\mathbf{z}(1)', \mathbf{z}(2)', \ldots, \mathbf{z}(N)']' \tag{2.4.3}$$

$$\mathbf{h}(\hat{\mathbf{x}}^{j}) = \left[\mathbf{h}(\hat{\mathbf{x}}^{j}, 1), \mathbf{h}(\hat{\mathbf{x}}^{j}, 2), \ldots, \mathbf{h}(\hat{\mathbf{x}}^{j}, N)\right]' \tag{2.4.4}$$

$$R = \begin{bmatrix} R(1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & R(2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & R(N) \end{bmatrix} \tag{2.4.5}$$

and

$$H^{j} = \begin{bmatrix} [\nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x}, 1)']' \big|_{\mathbf{x}=\hat{\mathbf{x}}^{j}} \\ [\nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x}, 2)']' \big|_{\mathbf{x}=\hat{\mathbf{x}}^{j}} \\ \vdots \\ [\nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x}, N)']' \big|_{\mathbf{x}=\hat{\mathbf{x}}^{j}} \end{bmatrix} \tag{2.4.6}$$

An initial estimate can be obtained by solving (2.2.3) with the noise set to zero based on the measurements for two transmitters at two different time instants.

The ML estimate of the target parameter $\mathbf{x}_{\mathrm{2D}}$ in (2.2.27) in a 2-D scenario can be found using the ILS technique in the same manner.

This dissertation assumes that a fixed number N of frames of measurements are processed together using a batch approach. Therefore, there is no need to use a recursive

algorithm for sequential update. One can sequentially process the measurements using a recursive estimator as more and more measurements are received. For example, the probability region considered in this dissertation will becomes smaller and smaller as more and more measurements are used to in the target parameter estimation. The decision on collision warning can be made earlier before N frames of measurements become available. However, since the problem is highly nonlinear, a recursive estimator would be by necessity suboptimal, either due to linearization or using a particle filter. We will consider this in the future: This is a topic for future investigation.

2.5 Collision Warning Approaches

2.5.1 Collision Warning via Hypothesis Testing Based on a Generalized Likelihood Function

The collision event at time t (t to be determined) is defined by equating the true target position $\mathbf{x}_{\mathrm{p}}(t)$ to the ownship position, namely,

$$\{\text{Collision at } t\} \triangleq \{\mathbf{x}_{\mathrm{p}}(t) = \mathbf{s}(t)\} \tag{2.5.1}$$

Following [4], the likelihood function of collision is the pdf of the predicted target position to time t (the "observation" based on which the collision warning can be made)

conditioned on (2.5.1)

$$\begin{aligned}
&\Lambda\left[\mathbf{x}_{\mathrm{p}}(t)=\mathbf{s}(t);\hat{\mathbf{x}}_{\mathrm{p}}(t)\right]=p[\hat{\mathbf{x}}_{\mathrm{p}}(t)|\mathbf{x}_{\mathrm{p}}(t)=\mathbf{s}(t)]\\
&=\mathcal{N}\left[\hat{\mathbf{x}}_{\mathrm{p}}(t);\mathbf{s}(t),P_{\mathrm{p}}(t)\right]=\left|2\pi P_{\mathrm{p}}(t)\right|^{-1/2}\\
&\cdot\exp\left(-\frac{1}{2}\left[\hat{\mathbf{x}}_{\mathrm{p}}(t)-\mathbf{s}(t)\right]'P_{\mathrm{p}}^{-1}(t)\left[\hat{\mathbf{x}}_{\mathrm{p}}(t)-\mathbf{s}(t)\right]\right)
\end{aligned}\tag{2.5.2}$$

where $\hat{\mathbf{x}}_{\mathrm{p}}(t)$ is given by (2.2.19). The use of the covariance $P_{\mathrm{p}}(t)$ in (2.5.2) is justified based on the discussion presented in Section 2.6, which validates the efficiency of (3.3.6).

Since the time t in (2.5.2) is not known, we estimate the CPA time as

$$\hat{t}_{\mathrm{CPA}}=\arg\max_{t}\Lambda\left[\mathbf{x}_{\mathrm{p}}(t)=\mathbf{s}(t);\hat{\mathbf{x}}_{\mathrm{p}}(t)\right]\tag{2.5.3}$$

The CPA time estimate is found by using the Quasi-Newton method with a cubic line search procedure. The search starts with an initial value, which can be obtained using (2.5.24) by considering the estimated target parameter as deterministic. For the purpose of simulations, the MATLAB function "fminunc" is used.

The collision warning can be formulated as a hypothesis testing problem as follows. The two hypotheses are, based on (2.5.3)

$$H_0:\mathbf{x}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}})=\mathbf{s}(\hat{t}_{\mathrm{CPA}})\tag{2.5.4}$$

$$H_1:\mathbf{x}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}})\neq\mathbf{s}(\hat{t}_{\mathrm{CPA}})\tag{2.5.5}$$

The (generalized[2]) likelihood function for H_0 is

$$\begin{aligned}\Lambda\left[H_0; \hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}})\right] &= \mathcal{N}\left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}); \mathbf{s}(\hat{t}_{\mathrm{CPA}}), P_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}})\right] \\ &= \mathcal{N}\left[\mathbf{s}(\hat{t}_{\mathrm{CPA}}); \hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}), P_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}})\right]\end{aligned} \tag{2.5.6}$$

For a given level of significance, say 0.0001% (assuming this is the desired confidence to avoid collision, $Q = 10^{-6}$ in (2.2.25)), there are two equivalent procedures to determine whether H_0 should be rejected.

Procedure 1: one computes

$$\epsilon = \left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}) - \mathbf{s}(\hat{t}_{\mathrm{CPA}})\right]' P_{\mathrm{p}}^{-1}(\hat{t}_{\mathrm{CPA}}) \left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}) - \mathbf{s}(\hat{t}_{\mathrm{CPA}})\right] \tag{2.5.7}$$

and

$$\epsilon_{\mathrm{th}} = F_{\chi^2}^{-1}(1 - Q, n_{\mathrm{dof}}) \tag{2.5.8}$$

where $F_{\chi^2}^{-1}$ is the inverse of the cumulative distribution function (cdf) of a chi-square random variable with n_{dof} degrees of freedom. If

$$\epsilon > \epsilon_{\mathrm{th}} \tag{2.5.9}$$

then $\mathbf{s}(\hat{t}_{\mathrm{CPA}})$ is outside the 99.9999% confidence region centered at $\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}})$, then one can say that collision is unlikely ($< 0.0001\%$). Otherwise a collision warning is issued.

[2]This is a generalized likelihood function because it relies on $\hat{t}_{\mathrm{CPA}}$, which is an estimate.

Procedure 2: one computes

$$\epsilon = \left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}) - \mathbf{s}(\hat{t}_{\mathrm{CPA}})\right]' P_{\mathrm{p}}^{-1}(\hat{t}_{\mathrm{CPA}}) \left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}) - \mathbf{s}(\hat{t}_{\mathrm{CPA}})\right] \tag{2.5.10}$$

and estimates the probability of collision as

$$\mathsf{P}_c = 1 - F_{\chi^2}(\epsilon, n_{\mathrm{dof}}) \tag{2.5.11}$$

where F_{χ^2} is the cdf of a chi-square random variable with n_{dof} degrees of freedom. If

$$\mathsf{P}_c > 0.0001\% \tag{2.5.12}$$

then a collision warning is alerted.

These two procedures are equivalent because of the invertibility of the cdf of the chi-square distribution.

So far, both the target and the ownship have been modeled as points of zero size. If one takes the physical dimensions of both the target and the ownship into consideration, a safety margin Δd (which would, typically, be more than the sum of the target and ownship sizes) is needed in the decision making. In this case, the definition of the collision event in (2.5.1) will be modified to be

$$\{\text{Collision at } t\} \triangleq \{\|\mathbf{x}_{\mathrm{p}}(t) - \mathbf{s}(t)\| \leq \Delta d\} \tag{2.5.13}$$

and the hypotheses in (2.5.4) and (2.5.5) will be modified as

$$H_0 : \|\mathbf{x}_{\mathrm{p}}(t) - \mathbf{s}(t)\| \leq \Delta d \tag{2.5.14}$$

$$H_1 : \|\mathbf{x}_{\mathrm{p}}(t) - \mathbf{s}(t)\| > \Delta d \tag{2.5.15}$$

Therefore, H_0 in (2.5.14) is rejected at a level of 0.0001% if $\mathbf{s}(\hat{t}_{\mathrm{CPA}})$ is outside the 99.9999% confidence region centered at $\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}})$ and

$$\Delta d < \min_{\mathbf{x}} \|\mathbf{s}(\hat{t}_{\mathrm{CPA}}) - \mathbf{x}\| \tag{2.5.16}$$

subject to

$$\begin{aligned} &\left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}) - \mathbf{x}\right]' P_{\mathrm{p}}^{-1}(\hat{t}_{\mathrm{CPA}}) \left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}) - \mathbf{x}\right] \\ &= F_{\chi^2}^{-1}(0.999999, n_{\mathrm{dof}}) \end{aligned} \tag{2.5.17}$$

that is, the minimum distance between $\mathbf{s}(\hat{t}_{\mathrm{CPA}})$ and any point on the surface of the 99.9999% confidence region is larger than Δd.

Equivalently, in a similar way to (2.5.10)–(2.5.12), one can also estimate the probability of collision as

$$\mathsf{P}_c = 1 - F_{\chi^2}(\epsilon_{\min}, n_{\mathrm{dof}}) \tag{2.5.18}$$

where

$$\epsilon_{\min} = \min_{\mathbf{x}} \left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}) - \mathbf{x}\right]' P_{\mathrm{p}}^{-1}(\hat{t}_{\mathrm{CPA}}) \left[\hat{\mathbf{x}}_{\mathrm{p}}(\hat{t}_{\mathrm{CPA}}) - \mathbf{x}\right] \tag{2.5.19}$$

subject to

$$\|\mathbf{s}(\hat{t}_{\mathrm{CPA}}) - \mathbf{x}\| \leq \Delta d \tag{2.5.20}$$

2.5.2 Collision Warning Based on a Bayesian Approach

In the Bayesian approach instead of using $\hat{t}_{\mathrm{CPA}}$ as "the collision time", the approach accounts for t_{CPA} as a random variable. Since the CPA distance d_{CPA} (the distance between the target and the ownship at the CPA time) is a function of the CPA time, d_{CPA} is also a random variable. One can define the collision event based on d_{CPA} and estimate the probability of collision based on an estimated pdf of d_{CPA}.

CPA distance as a function of the target parameter

Under the assumption that both the target and the ownship are moving with constant velocities, the CPA time is when the target and the ownship are closest to each other, that is

$$\begin{aligned} t_{\mathrm{CPA}} &= \arg\min_t \|\mathbf{x}_p(t) - \mathbf{s}(t)\| \\ &= \arg\min_t \|(\mathbf{x}_0 + t\dot{\mathbf{x}}_0) - (\mathbf{s}_0 + t\dot{\mathbf{s}}_0)\| \\ &= \arg\min_t \|(\mathbf{x}_0 + t\dot{\mathbf{x}}_0) - (\mathbf{s}_0 + t\dot{\mathbf{s}}_0)\|^2 \\ &= \arg\min_t d^2 \end{aligned} \tag{2.5.21}$$

Taking the derivative of d^2 with respect to t and setting it to zero

$$\frac{dD}{dt} = 2\left[(\mathbf{x}_0 + t\dot{\mathbf{x}}_0) - (\mathbf{s}_0 + t\dot{\mathbf{s}}_0)\right]'\left[\dot{\mathbf{x}}_0 - \dot{\mathbf{s}}_0\right] = 0 \tag{2.5.22}$$

the CPA time is obtained as

$$t_{\mathrm{CPA}} = -\frac{\left[\mathbf{x}_0 - \mathbf{s}_0\right]'\left[\dot{\mathbf{x}}_0 - \dot{\mathbf{s}}_0\right]}{\|\dot{\mathbf{x}}_0 - \dot{\mathbf{s}}_0\|^2} \tag{2.5.23}$$

and the CPA distance is therefore a function of the target parameter $\mathbf{x}$ in (2.2.2)

$$\begin{aligned} d_{\mathrm{CPA}} &= f(\mathbf{x}) = \|\mathbf{x}_p(t_{\mathrm{CPA}}) - \mathbf{s}(t_{\mathrm{CPA}})\| \\ &= \|\mathbf{x}_0 - \mathbf{s}_0 - \frac{[\mathbf{x}_0 - \mathbf{s}_0]'[\dot{\mathbf{x}}_0 - \dot{\mathbf{s}}_0]}{\|\dot{\mathbf{x}}_0 - \dot{\mathbf{s}}_0\|^2}(\dot{\mathbf{x}}_0 - \dot{\mathbf{s}}_0)\| \end{aligned} \tag{2.5.24}$$

In the above, $\|\cdot\|$ is the Cartesian norm.

Estimation of the probability density of d_{CPA}

Assuming a diffuse (non-informative) prior density for the target parameter $\mathbf{x}$, as in [4], the posterior density of $\mathbf{x}$ conditioned on $\hat{\mathbf{x}}_{\mathrm{ML}}$, given by (3.3.6), is approximated as

$$p[\mathbf{x}|\hat{\mathbf{x}}_{\mathrm{ML}}] = \mathcal{N}\left[\mathbf{x}; \hat{\mathbf{x}}_{\mathrm{ML}}, J^{-1}\right] \tag{2.5.25}$$

This Gaussian approximation is reasonable as Section 2.6 shows that the ML estimate can be considered as unbiased and statistically efficient, that is, (2.2.17) and (2.2.18) hold.

One possible way of estimating the density of d_{CPA} is to draw N_s samples of $\mathbf{x}$ from (2.5.25), obtain N_s samples of d_{CPA} and fit a density based on these samples. In this dissertation, we estimate the pdf of d_{CPA} as a Rician distribution. The validity of fitting the Rician distribution is confirmed in Section 2.6. The Rician distribution with noncentrality parameter $\nu \geq 0$ and scale parameter $\sigma > 0$ has the density function

$$p_{d_{\mathrm{CPA}}}(x|\nu,\sigma) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2+\nu^2}{2\sigma^2}\right) I_0\left(\frac{x\nu}{\sigma^2}\right), \quad x > 0 \tag{2.5.26}$$

where $I_0(\cdot)$ is the zero-order modified Bessel function of the first kind. Based on the N_s samples of d_{CPA}, the ML estimates ν^{ML} and σ^{ML} can be obtained using the method

presented in [52].

Decision making

One can define the collision event as

$$\{\text{Collision}\} = \{d_{\text{CPA}} \leq d_{\min}\} \tag{2.5.27}$$

where $d_{\min}$ is the minimum distance between the aircraft for which a collision will not occur, that is, one believe that a collision occurs if the estimated d_{CPA} is less than $d_{\min}$ by taking the aircraft dimensions into account. Therefore, the probability of collision is

$$\begin{aligned} \mathsf{P}_c &= \mathsf{P}(\{\text{Collision}\}) = \mathsf{P}(\{d_{\text{CPA}} \leq d_{\min}\}) \\ &= \int_0^{d_{\min}} p_{d_{\text{CPA}}}(x|\nu^{\text{ML}}, \sigma^{\text{ML}})\, dx \end{aligned} \tag{2.5.28}$$

The integration in (2.5.28) is evaluated using the MATLAB function "cdf". The average computational time in a single run, including the target parameter estimation, sampling, Rician distribution parameter estimation and the integration (2.5.28), is around 0.6 s. This computation is performed in MATLAB 2015a on a Windows machine equipped with a 2.40 GHz Intel Core 2 Quad CPU with 4 GB RAM. Consequently, we feel it is not unreasonable to claim that it would be real-time feasible with a dedicated processor and code in machine language.

If P_c is smaller than, say, 0.0001%, the collision is unlikely and no warning will be issued. Otherwise, a warning will be given.

2.6 Simulation Results

2.6.1 Efficiency of ML Estimator of the Target Parameter

Under the hypothesis $H_{\mathbf{x}}$ that the ML estimator (3.3.6) is unbiased and efficient, that is, the mean of the estimation error is zero and the estimation errors match the covariance given by the CRLB as in (2.2.18), the NEES for the target parameter

$$\epsilon_{\mathbf{x}} = \tilde{\mathbf{x}}' J \tilde{\mathbf{x}} \tag{2.6.1}$$

is chi-square distributed with $n_{\mathbf{x}}$ degrees of freedom. The sample average NEES from N Monte Carlo runs would be

$$\bar{\epsilon}_{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \epsilon_{\mathbf{x}}^{i} \tag{2.6.2}$$

where $\epsilon_{\mathbf{x}}^{i}$ is a sample from i-th Monte Carlo run. The quantity $N\bar{\epsilon}_{\mathbf{x}}$ is chi-square distributed with $Nn_{\mathbf{x}}$ degrees of freedom. Therefore, for a given level of significance α, $H_{\mathbf{x}}$ cannot be rejected if

$$\bar{\epsilon}_{\mathbf{x}} \in \left[\frac{L_{\mathbf{x}}}{N}, \frac{U_{\mathbf{x}}}{N} \right] \tag{2.6.3}$$

where $L_{\mathbf{x}}$ and $U_{\mathbf{x}}$ are the $\dfrac{100\alpha}{2}$ and $\dfrac{100(2-\alpha)}{2}$ percentile points of a chi-square random variable with $Nn_{\mathbf{x}}$ degrees of freedom.

The sample averages of the NEES for the 6-D target parameter ($n_{\mathbf{x}} = 6$) in Scenario 5 from 100 Monte Carlo runs based on the CRLB evaluated at the truth and at the estimate are calculated. The values are 6.2576 and 6.2207, which can be considered practically

identical. Both values fall inside the two-sided 60% probability region [5.70, 6.29], which means that one can accept the null hypothesis $H_{\mathbf{x}}$ at a high significance level of 40%, i.e., we allow a probability of making a type I (reject $H_{\mathbf{x}}$ when it is true) error that is 40% in this case. In addition, the likelihood function (2.2.13) is exponential, which is a necessary, although not sufficient, condition for the MLE to be efficient [56]. This strongly affirms the acceptability of the CRLB as the actual covariance of the 3-D estimator in Scenario 5. The same reasoning was used in [45] to demonstrate the statistical efficiency of composite position measurements from passive sensors for a variety of geometries.

The sample averages of the NEES for the 4-D target parameter ($n_{\mathbf{x}} = 4$) in Scenario 9 from 100 Monte Carlo runs based on the CRLB evaluated at the truth and at the estimate are also calculated. The values are 3.9209 and 3.9199, which can also be considered practically identical. Both values fall inside the two-sided 30% probability region [3.885, 4.103] (i.e., the alternative hypothesis H_1 ("not efficient") is rejected at a rather high significance level of 70%), which confirms the acceptability of the CRLB as the actual covariance of the 2-D estimator in Scenario 9. Therefore, the unbiasedness and efficiency of the ML estimator is verified in both scenarios considered in this dissertation.

Simulation results also show that the collision warning algorithm based on the CRLB covariance provides reliable performance by comparing the CRLB-based error probability of 10^{-5} with 10^5 Monte Carlo runs. The number of missed collision detections in this case was 2 in 10^5 runs. If one considers the following hypothesis test

$$H_0 : P_{\text{FA}} = 10^{-5} \tag{2.6.4}$$

$$H_1 : P_{\text{FA}} > 10^{-5} \tag{2.6.5}$$

then, based on the Poisson approximation with parameter λ of the binomial distribution of

the number of missed detections in 10^5 runs ($H_0 : \lambda = 1$; $H_1 : \lambda > 1$ with 10^5 runs), the probability of getting no more than 2 missed collision detections is 0.9197, i.e., we can accept H_0 at a level of significance of 8%. (The outcome is to the left of the 8% tail.) With 10^5 runs and the threshold set for $P_{\rm FA} = 10^{-4}$ (then $H_0 : \lambda = 10$), we obtained 5 missed collision detections, i.e., in this case H_0 can be accepted at a level of significance of 93% (unusually high).

2.6.2 Efficiency of the CPA Time Estimate

Based on (2.5.2), the CPA time estimate $\hat{t}_{\rm CPA}$ in (2.5.3) is a function of the target parameter estimate $\hat{\mathbf{x}}$, denoted as

$$\hat{t}_{\rm CPA} = g[\hat{\mathbf{x}}] \tag{2.6.6}$$

Unfortunately, the function g has no closed-form expression, therefore, we estimate the variance of $\hat{t}_{\rm CPA}$ using the unscented transformation technique [5] as follows:

Firstly, by the method of moment matching, the Gaussian density $\mathcal{N}(\hat{\mathbf{x}}; \mathbf{x}_{\rm t}, P)$ of the $n_{\mathbf{x}}$-dimensional $\hat{\mathbf{x}}$ (centered at the true value $\mathbf{x}_{\rm t}$; this is in view of the unbiasedness and efficiency discussed in the previous subsection) is replaced by a ($2n_{\mathbf{x}}$+1)-point probability mass function (pmf)

$$p(\hat{\mathbf{x}}) = \sum_{i=-n_{\mathbf{x}}}^{n_{\mathbf{x}}} w^i \delta\left(\hat{\mathbf{x}} - \hat{\mathbf{x}}^i\right) \tag{2.6.7}$$

where $\delta(\cdot)$ is the Dirac delta function. The sigma points of the pmf are

$$\hat{\mathbf{x}}^i = \mathbf{x}_{\mathrm{t}} + \mathrm{sgn}(i)a[P]_{|i|}^{1/2} \quad i = -n_{\mathbf{x}}, \ldots, n_{\mathbf{x}}; \quad a \in \mathbb{R} \tag{2.6.8}$$

where $[P]_i^{1/2}$ is the i-th column of the Cholesky factor of P defined by

$$\sum_{i=1}^{n_{\mathbf{x}}} [P]_i^{1/2} \left([P]_i^{1/2}\right)' = P \tag{2.6.9}$$

and the signum function is defined as

$$\mathrm{sgn}(i) \triangleq \begin{cases} -1 & i < 0 \\ 0 & i = 0 \\ 1 & i > 0 \end{cases} \tag{2.6.10}$$

The point masses are [5]

$$w^i = \begin{cases} \dfrac{1}{2a^2} & |i| = 1, \ldots, n_{\mathbf{x}} \\ \dfrac{a^2 - n_{\mathbf{x}}}{a^2} & i = 0 \end{cases} \tag{2.6.11}$$

which sum up to unity. With the sigma points and point masses specified above, the pmf (2.6.7) has the same mean and covariance matrix as the Gaussian pdf $\mathcal{N}(\hat{\mathbf{x}}; \mathbf{x}_{\mathrm{t}}, P)$ regardless of the value of a. A reasonable choice of a is $\sqrt{n_{\mathbf{x}} + 2}$, so we use that in this dissertation. In a simulated scenario, the true value of the parameter is known. However, in a real scenario where the true value of the parameter is unknown and needs to be estimated, the sigma points of $\hat{\mathbf{x}}$ need to match the moments of $\mathcal{N}(\hat{\mathbf{x}}, P)$ with P evaluated at the estimate.

Secondly, a sigma point of $\hat{t}_{\mathrm{CPA}}$ corresponding to $\hat{\mathbf{x}}^i$ can be obtained as

$$\hat{t}^i = g[\hat{\mathbf{x}}^i] \tag{2.6.12}$$

Lastly, the pdf of $\hat{t}_{\mathrm{CPA}}$ is approximated by the pmf

$$p(\hat{t}_{\mathrm{CPA}}) = \sum_{i=-n_{\mathbf{x}}}^{n_{\mathbf{x}}} w^i \delta\left(\hat{t}_{\mathrm{CPA}} - \hat{t}^i\right) \tag{2.6.13}$$

which has mean

$$\bar{t} = \sum_{i=-n_{\mathbf{x}}}^{n_{\mathbf{x}}} w^i \hat{t}^i \tag{2.6.14}$$

and variance

$$\sigma_t^2 = \sum_{i=-n_{\mathbf{x}}}^{n_{\mathbf{x}}} w^i \left(\hat{t}_i - \bar{t}\right)^2 \tag{2.6.15}$$

If we assume that $\hat{t}_{\mathrm{CPA}}$ is a Gaussian random variable with mean t_{CPA} and variance σ_t^2, then under the hypothesis H_t that the estimator (2.5.3) is unbiased and efficient, the NEES for the CPA time

$$\epsilon_t = \frac{(t_{\mathrm{CPA}} - \hat{t}_{\mathrm{CPA}})^2}{\sigma_t^2} \tag{2.6.16}$$

is chi-square distributed with 1 degree of freedom. The sample average NEES from N Monte Carlo runs would be

$$\bar{\epsilon}_t = \frac{1}{N} \sum_{i=1}^{N} \epsilon_t^i \tag{2.6.17}$$

where ϵ_t^i is a sample from i-th Monte Carlo run. The quantity $N\bar{\epsilon}_t$ is chi-square distributed with N degrees of freedom. Therefore, for a given level of significance α, H_t cannot be rejected if

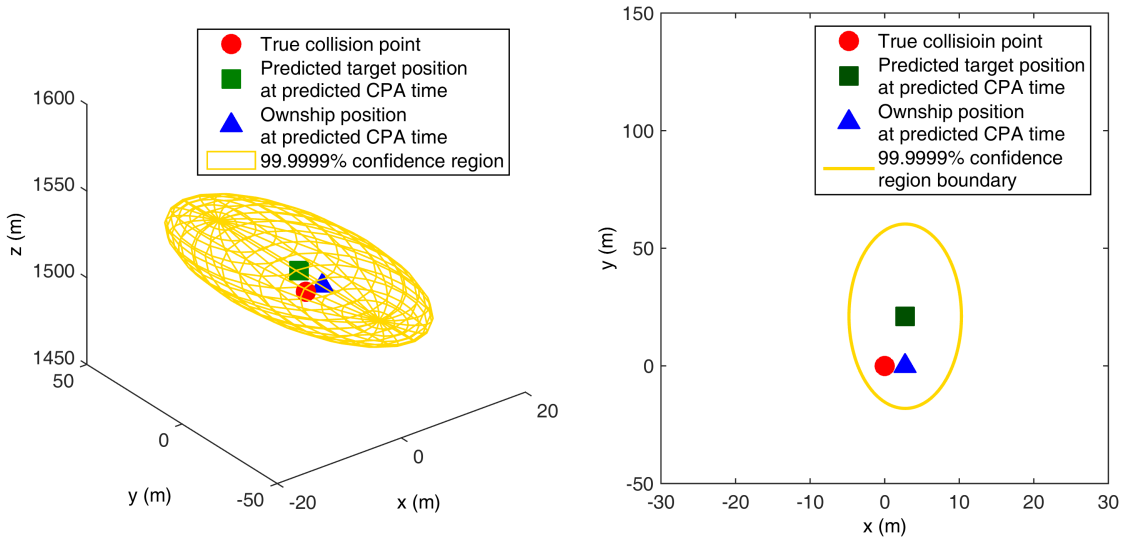
$$\bar{\epsilon}_t \in \left[\frac{L_t}{N}, \frac{U_t}{N}\right] \tag{2.6.18}$$

where L_t and U_t are the $\frac{100\alpha}{2}$ and $\frac{100(2-\alpha)}{2}$ percentile points of a chi-square random variable with N degrees of freedom.

The sample averages of the NEES for the CPA time estimate in Scenario 5 from 100 Monte Carlo runs based on the true value and the estimate of the target parameter are calculated. The values are 1.0914 and 1.0950, which can be considered practically identical. Both values fall inside the two-sided 60% probability region [0.879, 1.117], which confirms the unbiasedness and efficiency of the CPA time estimate in Scenario 5 for the 3-D case. The sample averages of the NEES for the CPA time estimate in Scenario 9 from 100 Monte Carlo runs based on the true value and the estimate of the target parameter are calculated. The values are 0.8753 and 0.8737, which can also be considered practically identical. Both values fall inside the two-sided 60% probability region, which confirms the unbiasedness and efficiency of the CPA time estimate in Scenario 9 for the 2-D case.

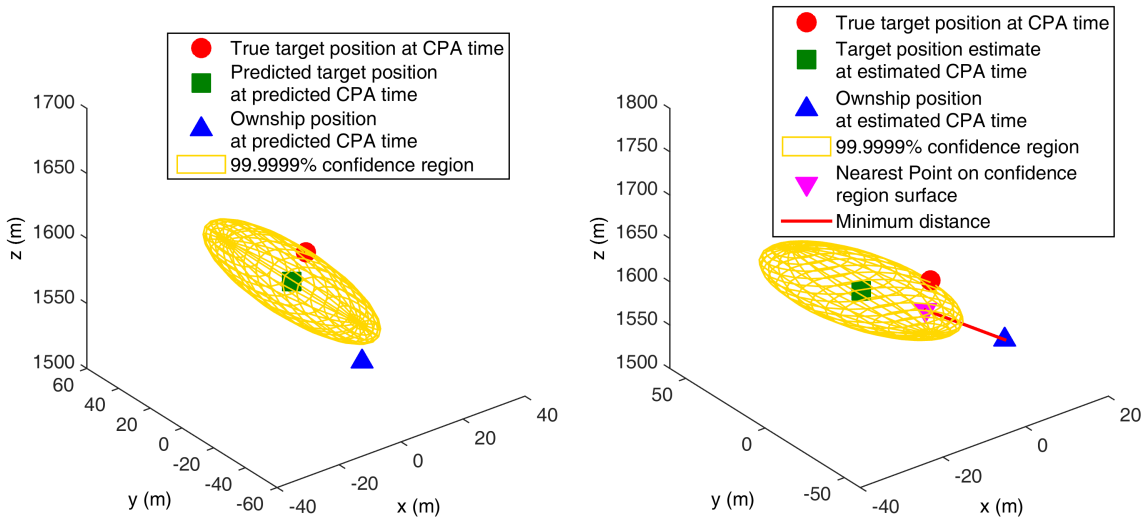
2.6.3 Collision Warning Based on the Generalized Likelihood Function

The collision warning is "on" for all 100 runs in Scenario 5 and 9 with the target and the ownship modeled as points, that is, at the predicted CPA time the ownship is inside the confidence region of the true target around its predicted position as illustrated in Figure



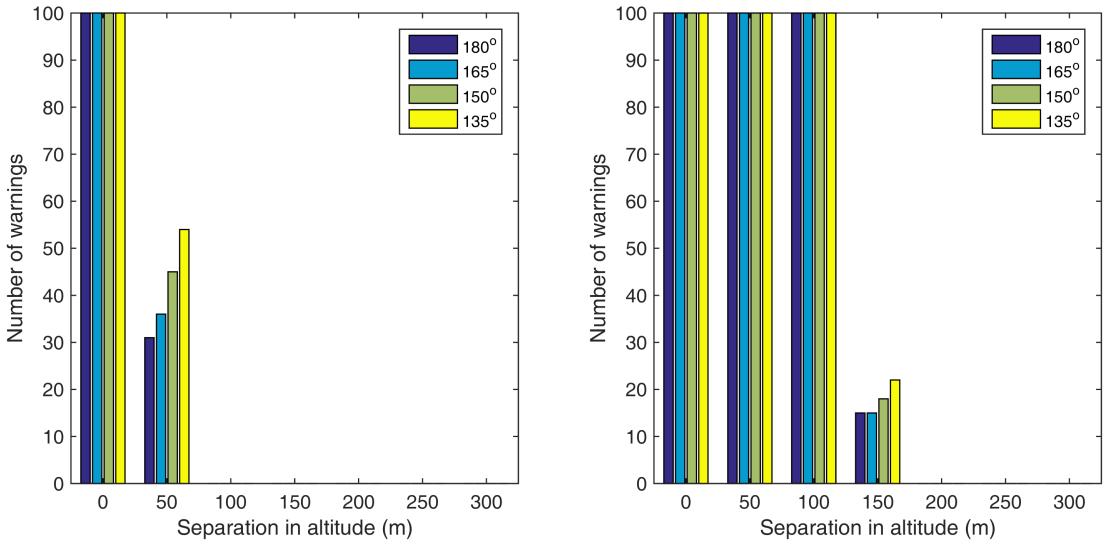
(A) Scenario 5 (B) Scenario 9

FIGURE 2.6.1: Collision warning is "on" in a single run in Scenarios 5 and 9.



(A) Without a safety margin (B) With a safety margin of 100 m

FIGURE 2.6.2: Collision warning decisions in a single run in Scenario 6. Collision warning is "off" without a safety margin but is "on" with a safety margin of 100 m.



(A) Without a safety margin

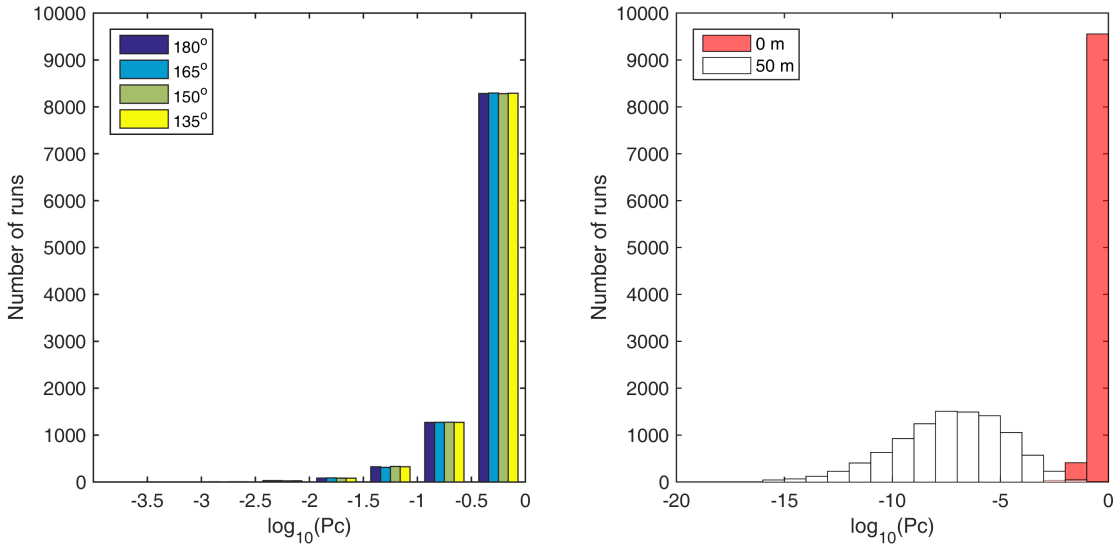
(B) With a safety margin of 100 m

FIGURE 2.6.3: The number of warnings in 100 runs using the 3-D likelihood based collision warning algorithm.

2.6.1(a) and 2.6.1(b). The collision warning is "off" for all 100 runs in Scenario 6, that is, at the predicted CPA time the ownship is outside the confidence region of the true target around its predicted position as illustrated in Figure 2.6.2(a).

Taking the physical dimensions of the aircraft into consideration and using a safety margin of 100 m, the collision warning is "on" for all 100 runs in Scenario 5 and 9. However, the collision warning is "on" for all 100 runs in Scenario 6, that is, false alarms occur. Although at the predicted CPA time the ownship is outside the confidence region of the true target around its predicted position, the minimum distance between the ownship and the ellipsoid is less than the safety margin as illustrated in Figure 2.6.2(b).

The term "CPA angle" is defined as the angle formed by the target velocity vector and the ownship velocity vector at the CPA time when they are projected on a plane at the same altitude. Therefore, the CPA angle is 180° in Scenarios 5, 6 and 9.



(A) Separation in altitude is 0 m

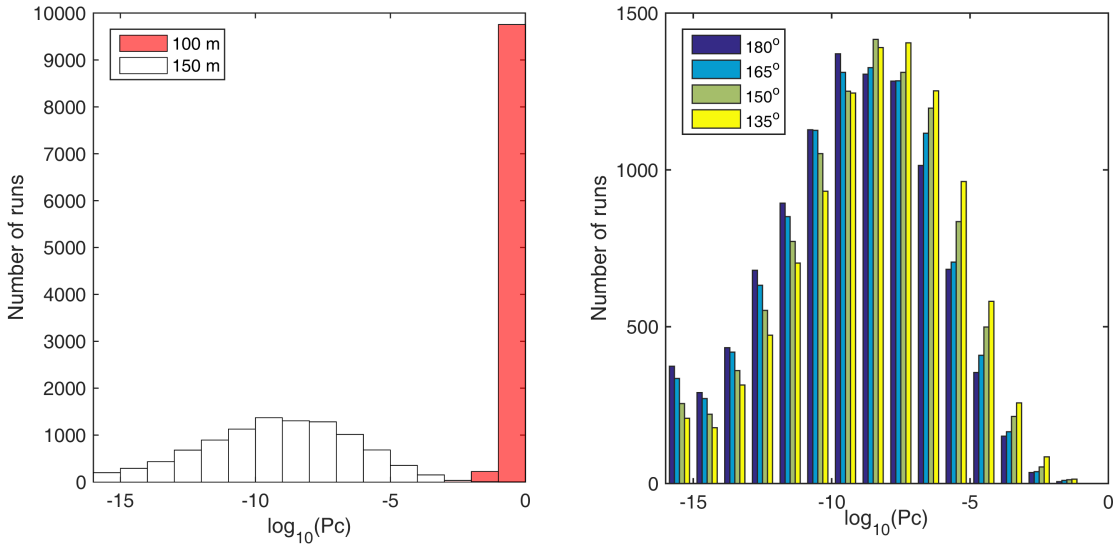
(B) CPA angle is 180^o

FIGURE 2.6.4: The histogram of $\log_{10} P_c$ in 10000 runs using the 3-D likelihood based collision warning algorithm.

The performance of the 3-D likelihood-based collision warning algorithm is further evaluated by varying the target and ownship altitude separation[3] from 0 to 300 m in steps of 50 m and the CPA angle from 180° to 135° in steps of 15° one parameter at a time in Scenario 5. From Figure 2.6.3(a), the 3-D likelihood based collision warning algorithm has no missed detections of a collision in 100 runs. There are some false alarms when the intruder and ownship altitude separation is 50 m and the number of false alarms increases slightly with the CPA angle decreasing. There are no false alarms when the intruder and ownship altitude separation is beyond 100 m.

Figure 2.6.4(a) shows the histogram of the logarithm of the estimated probability of collision in 10, 000 runs from scenarios with different CPA angles when there is a collision (the target and ownship altitude separation is 0 m). The estimated probability of collision

[3] 1000 ft (≈ 300 m) is a global standard for vertical separation



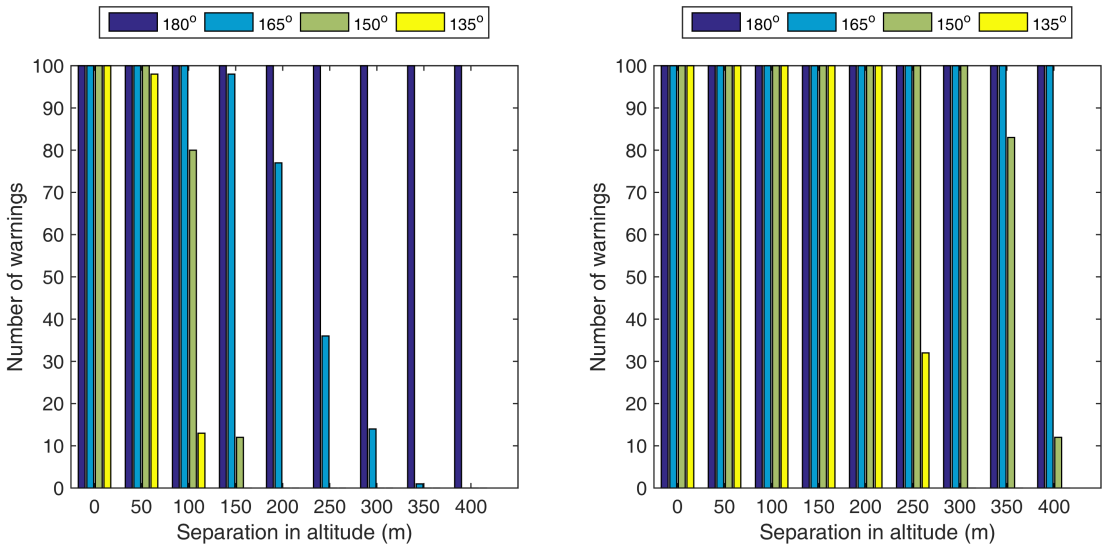
(A) CPA angle is 180^o

(B) Separation in altitude is 150 m

FIGURE 2.6.5: The histogram of $\log_{10} \mathrm{P}_c$ in 10000 runs using the 3-D likelihood based collision warning algorithm with a safety margin of 100 m.

has a similar distribution for different CPA angles, which is also observed at other levels of altitude separation. More than 95% of the time, the probability of collision is estimated to be larger than 10%. Since the probability of collision is always estimated to be larger than 0.0001%, there are no missed detections, which confirms the results shown in Figure 2.6.3(a). As the separation in altitude increases from 0 to 50 m, the estimated probability of collision gets much smaller as shown in Figure 2.6.4(b). Similar phenomena are also observed at other CPA angles. False alarms occur about 30% of the time when the estimated probability of collision is larger than 0.0001%. When the intruder and ownship altitude separation is beyond 100 m, the estimated probability is always less than 10^{-16} and hence the corresponding distributions are not shown in Figure 2.6.4(b).

With a safety margin of 100 m, from Figure 2.6.3(b), the 3-D likelihood based collision warning algorithm has no missed detections of a collision. However, it becomes more



(A) Without a safety margin

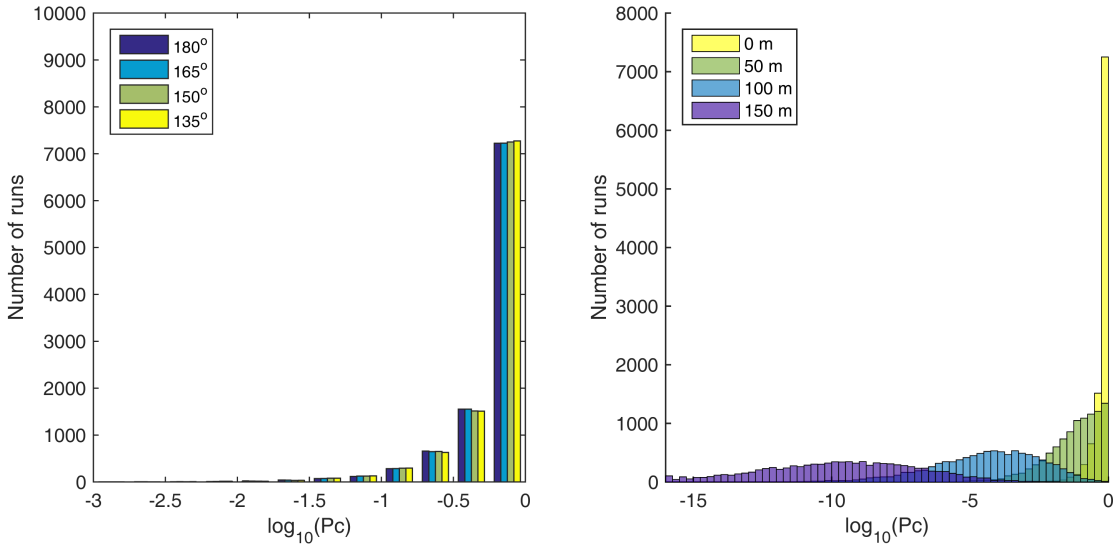
(B) With a safety margin of 100 m

FIGURE 2.6.6: The number of warnings in 100 runs using the 2-D likelihood based collision warning algorithm.

conservative and there are always false alarms when the intruder and ownship altitude separation is below 100 m, which is not surprising because of a safety margin of same distance. The number of false alarms starts to decrease at 150 m altitude separation.

When the altitude separation is 0 or 50 m, it turns out that the estimated probability of collision is always 1 in 10,000 Monte Carlo runs. When the altitude separation is 100 m, the estimated probability of collision is not always unity: see Figure 2.6.5(a) for the distribution of its logarithm. When the separation is 150 m, the estimated probability of collision becomes much smaller. The similar distributions are also observed at other CPA angles as illustrated in Figure 2.6.5(b). It is estimated to be less than 0.0001% for more than 90% of the time. When the intruder and ownship altitude separation is beyond 200 m, the estimated probability is always less than 10^{-16}.

The performance of the 2-D likelihood based collision warning algorithm is evaluated in



(A) Separation in altitude is 0 m

(B) CPA angle is 150^o

FIGURE 2.6.7: The histogram of $\log_{10} P_c$ in 10000 runs using the 2-D likelihood based collision warning algorithm.

the same manner. Figure 2.6.6(a) shows that there are no missed detections of collisions in 100 runs, which is the same as in the 3-D scenarios. The estimated probability of collision is very close to 1 for most of the time in 10000 runs and its distribution is similar at different CPA angles as indicated in Figure 2.6.7(a). The CPA angle has a pronounced effect on false alarms in the 2-D case. Recall that in the 2-D scenarios it is (conservatively) assumed that the intruder is at the same altitude as the ownship, which is not true when the altitude separation is not zero. When the CPA angle is close to 180°, the collision is very likely to occur based on the same altitude assumption, and, consequently, the false alarm rate is therefore very high. At other CPA angles, as the altitude separation increases, the number of false alarms decreases and the estimated probability of collision becomes smaller as indicated in Figure 2.6.7(b).

With a safety margin of 100 m, there are no missed detections of collisions. However,

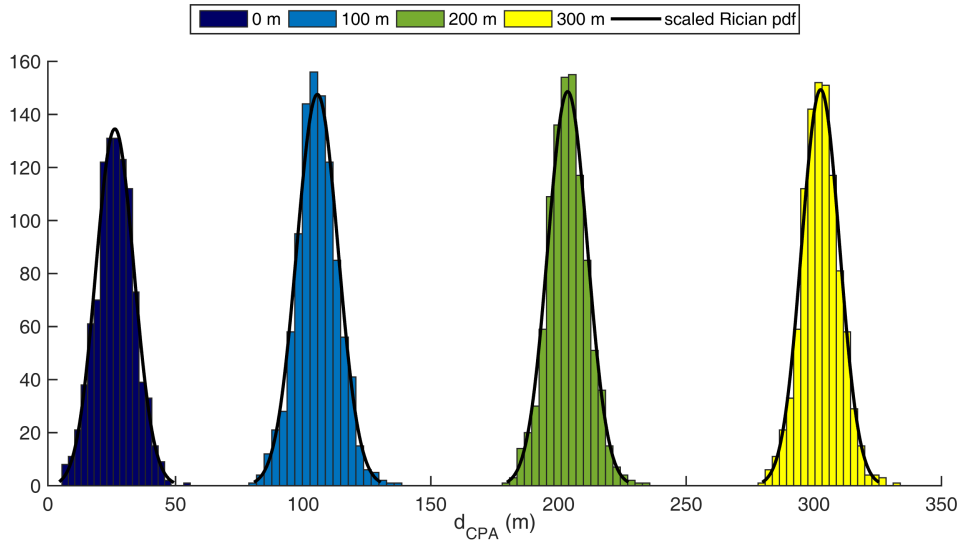


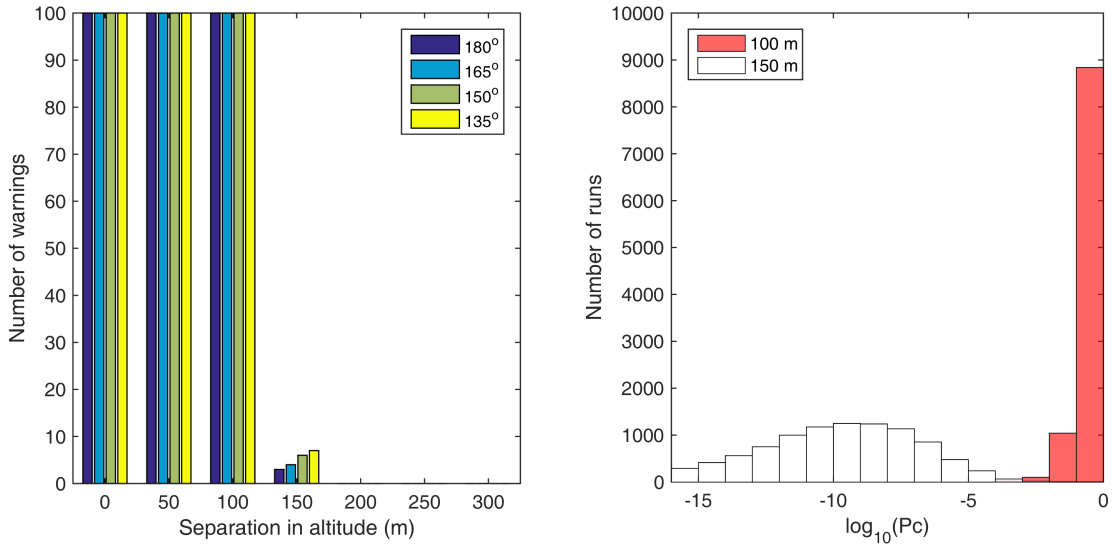
FIGURE 2.6.8: The histograms of d_{CPA} with fitted Rician distributions when the CPA angle is 180^o.

there are more false alarms because of both the same altitude assumption and the safety margin.

Based on the above observations, we submit that 3-D estimation with at least 3 transmitters is the only one reliable configuration for collision warning and that 2-D estimation with 2 transmitters is prone to false alarms when the CPA angle is 180^o even if there is more than 400 m altitude separation.

2.6.4 Collision Warning Based on the Bayesian Approach

The one-sample Kolmogorov-Smirnov test fails to reject the hypothesis that 1000 samples of d_{CPA} comes from a Rician distribution with parameters that are ML estimates based on the same 1000 samples in all 100 Monte Carlo runs in each of the 3-D scenarios that are used to evaluate the likelihood based collision warning algorithm in previous subsection. The fitted Rician distributions with the corresponding histograms of d_{CPA} at different



(A) Histogram of the number of warnings in 100 runs (B) Histogram of $\log_{10} \mathsf{P}_c$ in 10000 runs with CPA angle 180^o

FIGURE 2.6.9: Performance of the 3-D Bayesian collision warning algorithm with $d_{\min} = 100\,\text{m}$.

levels of altitude separations at 180^o CPA angle are shown in Figure 2.6.8. As shown in the previous subsection, 2-D collision warning with 2 transmitters under the same altitude assumption is unreliable because it is prone to false alarm, therefore the Bayesian approach is considered only in the multistatic configuration with 3 transmitters in this dissertation. It turned out that the hypothesis that the pdf of d_{CPA} is Rician is no longer valid in the 2D scenarios when the same altitude assumption does not hold. Nevertheless, it is possible to estimate the probability of collision by fitting a kernel distribution instead of a Rician distribution in those 2-D scenarios.

The performance of 3-D Bayesian collision warning algorithm with $d_{\min} = 100\,\text{m}$ is shown in Figure 2.6.9, which is very similar to that of 3-D likelihood based collision warning algorithm with a safety margin 100 m. There are no missed detections of a collision but there are always false alarms when the intruder and ownship altitude separation is be-

low 100 m. The number of false alarms starts to decrease at 150 m altitude separation and becomes zero when the altitude separation is beyond 200 m.

Comparing Figure 2.6.9(b) and 2.6.5(a), the estimated probability of collision from the 3-D Bayesian algorithm has a similar distribution to that from the 3-D likelihood based algorithm. As the altitude separation increases, the estimated probability of collision is getting smaller. When the altitude separation is 0 or 50 m, it turns out the estimated probability of collision is always 1 in 10,000 Monte Carlo runs. When the intruder and ownship altitude separation is beyond 200 m, the estimated probability is always less than 10^{-16}.

2.7 Conclusions

The ability to sense and avoid non-cooperative targets is essential for UAS to perform routine tasks when they are not alone in the airspace. We investigated several configurations with bistatic range and range rate measurements for collision warning. It turned out that a multistatic configuration is needed to provide good observability of the target, which is useful for collision warning. The minimum number of the transmitters required is three in a 3-D scenario and two in a 2-D scenario. We also implemented an ML estimator in both types of scenarios using the ILS technique and showed that the estimator can be considered as statistically efficient through Monte Carlo simulations for the scenarios considered. Based on the ML estimator, the collision warning was approached in two different ways. The first method is formulating the collision as a hypothesis testing problem using a generalized likelihood function, where the efficiency of the CPA time is also verified. The second method is a Bayesian formulation focusing on the time of CPA modeled as a random variable. Only the multistatic configuration with three transmitters is reliable for

collision warning because the multistatic configuration with two transmitters based on the same target and ownship altitude assumption turns out to be prone to false alarms. When the minimum distance in the Bayesian approach is the same as the safety margin in the likelihood based approach, both algorithms yield very similar collision warning performance.

Chapter 3

Evaluation of Fusion Algorithms for Passive Localization of Multiple Transient Emitters

3.1 Introduction

This chapter considers the problem of multiple transient emitter (target) localization using a group of passive sensors. One particular application is to utilize a network of acoustic gunfire detection systems on a group of soldiers to localize adversaries in a battlefield [31] [47]. It is assumed that the targets are stationary during the time window of interest but the number of the targets are unknown. The sensors can measure line of sight (LOS) angles to the targets by detecting their emitted acoustic signals and record the times of arrival of the detected signals. Missed detections and false alarms are present due to the imperfection of the sensors. Furthermore, the association between the measurements and the targets are unknown, that is, each sensor does not know from which target (or clutter) a

particular measurement originates. Before estimating the position of any target, one has to associate the measurements from all the sensors. Therefore, the quality of data association is critical to the overall localization performance.

The problem of data association has been studied extensively in tracking multiple targets. Methods including multiple hypothesis tracking (MHT) [7], joint probabilistic data association (JPDA) filter [30] and probability hypothesis density (PHD) filter [38] are recursive algorithms that require persistent measurements and provide solutions to a dynamic data association problem. Therefore, they cannot be employed to solve the static data association problem considered in the situation of multiple transient emitter localization.

There are two different philosophies — hard data association and soft data association (see [5], Section 2.4.3) — in solving the static data association problem considered in this chapter. Hard data association either assigns a measurement to one and only one target or condemn it as a false alarm, in other words, the probability of a measurement coming from a target is either 0 or 1 (discrete). In contrast, soft data association assigns the event that a measurement originates from a target to a (continuous) probability, which can be any value between 0 and 1.

The hard data association for S lists of measurements with one list from each sensor[1], assuming a Bernoulli measurement model that the number of measurements from each target received at each sensor is a Bernoulli random variable with parameter equal to the probability of detection, leads to an S-dimensional (S-D) assignment problem, which can be formulated as a discrete constrained optimization problem aiming to find out the set of S-tuples of measurements that minimizes the overall association cost. The number of possible S-tuple set for T targets and S sensors in the absence of missed detections and

[1] In a multisensor localization application, as in this chapter, the number of lists is the same as the number of sensors.

false alarms is $(T!)^{S-1}$, from which it can be seen that S-D assignment problem is non-deterministic polynomial-time (NP) hard with $S \geq 3$. Therefore, it is of great interest and importance to find robust suboptimal algorithms.

The Lagrangian relaxation based approach [18], which is termed as the S-D algorithm in this chapter, provides a measure of how close the final solution is to the (unknown) optimal solution in terms of the association cost. The application of the S-D assignment algorithm on a multiple shooter localization problem using a small number of sensors was presented in [46]. Although it does not explore the entire space of the S-tuple sets, it needs to calculate the cost of candidate S-tuples. The cost calculation involves finding the maximum likelihood estimate of the target locations and can take most of the computational time. The number of candidate S-tuples for T targets and S sensors in the absence of missed detections and false alarms is T^S, which increases exponentially with the number of sensors. Since more sensors generates more accurate estimates in the fusion center, computationally efficient algorithms are required when a large number of sensors are deployed.

The S_0-D+SEQ(2-D) algorithm [61], which performs the S-D assignment algorithm on S_0 lists of measurements before applying the modified Auction algorithm [48] for 2-D assignments on the remaining lists sequentially $S - S_0$ times, is a more efficient algorithm than the S-D assignment. The number of candidate associations increases quadratically (rather than combinatorially/exponentially) with the number of sensors. Because of the ghosting problem [5], the S_0-D step requires, in general, at least 3 lists to achieve reliable association. However, since in the present problem one also has arrival times, one can use $S_0 = 2$.

The problem of multiple shooter localization using a single sensor [32] or using multiple sensors [33] is formulated as a cardinality (number of targets) selection problem which assumes a Poisson measurement model that the number of measurements from each target

received at each sensor is a Poisson random variable with parameter equal to the probability of detection. The measurements at a single sensor from all targets and the clutter are modeled as a Poisson point process (PPP) [17]. For each possible selected cardinality, one solves a sub-problem based on the learning expectation-maximization (EM) algorithm [19] to select the best cardinality based on an information criterion [1] [51]. During every iteration of the EM algorithm, each measurement will be assigned a probability of having originated from a target, which is an example of the soft data association.

In this chapter, we discuss two classes of algorithms, each for a specific measurement model in the multiple passive transient emitter localization problem. For the Bernoulli measurement model, the SEQ[m(2-D)] algorithm [3], the m-best version of the fastest sequential algorithm SEQ(2-D), is shown to be able to yield associations as good as the S-D assignment. The ghosting effect for a pair of sensors is no longer present due to the estimation of the signal emission time, which makes SEQ[m(2-D)] practical. For the Poisson measurement model, we discuss both uniform-Gaussian mixture (UGM) [6] and Poisson point processes (PPP) modeling of the lists of measurements for the cardinality selection formulation. In the previous work on PPP [33], both the range and bearing measurements are assumed available and the initialization in the EM-based algorithm uses a finite set including target locations that are close to the truth. Since the range measurement and prior information for a "good" initialization is not always available in the real world, this chapter considers bearing and time of arrival measurements and presents some measurement-driven initialization approaches for the EM-based algorithms. In the uniform-Gaussian mixture formulation, the probability of detection (assumed not known) and the expected number of false alarms per sensor (which can be known or unknown) are incorporated into the mixture coefficients and the maximization step in the EM algorithm is developed such that the constraint that the resulting probability of detection is not larger than unity is always

satisfied.

The remaining sections of this chapter are organized as follows. Section 4.3 describes the problem of localizing an unknown number of transient emitters. Section 3.3 assumes a Bernoulli measurement model for each target, formulates an S-D assignment problem and presents two assignment algorithms. Sections 3.4 and 3.5 present the uniform-Gaussian mixture and Poisson point process formulations both of which assume a Poisson measurement model for each target. Simulation results are shown and analyzed in Section 4.6 and the conclusions are drawn in Section 3.7. For the convenience of the reader, the list of notations used in this chapter is given in Table 3.1.1.

3.2 Problem Description

Consider a scenario where there are N targets located in $\mathbb{R}^2$. The target locations (fixed) are denoted as

$$\mathbf{T} = (\mathrm{T}_1,\, \mathrm{T}_2,\, \ldots,\, \mathrm{T}_N) = \left(\begin{bmatrix} T_{x_1} \\ T_{y_1} \end{bmatrix},\, \begin{bmatrix} T_{x_2} \\ T_{y_2} \end{bmatrix},\, \ldots,\, \begin{bmatrix} T_{x_N} \\ T_{y_N} \end{bmatrix} \right) \tag{3.2.1}$$

and the emission times are denoted as

$$\mathbf{t}^{\mathrm{e}} = (t_1^{\mathrm{e}},\, t_2^{\mathrm{e}},\, \ldots,\, t_N^{\mathrm{e}}) \tag{3.2.2}$$

TABLE 3.1.1: List of notations

Notation	Definition
S	Dimension of the assignment problem
$\mathbf{T}$	Set of target position vectors
T_i	Position vector of target i
$\mathbf{t}_\mathrm{e}$	Set of emission times
t_i^e	Signal emission time of target i
S_ℓ	Position vector of sensor ℓ
n_ℓ	Number of measurements at sensor ℓ
$\mathbf{z}_{\ell j}$	j-th measurement at sensor ℓ
N	Number of targets
N_s	Number of sensors
N_fa	Expected number of false alarms per sensor
T_0	The clutter
Φ	The range of the sensor field view
$\mathbf{Z}_\ell$	Augmented measurement list at sensor ℓ
$\mathbf{z}_{\ell 0}$	Dummy measurement at sensor ℓ
$Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}}$	An N_s-tuple of measurements, one from each sensor
$c_{j_1 j_2 \ldots j_{N_\mathrm{s}}}$	Cost of associating $Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}}$ with a target
$\rho_{j_1 j_2 \ldots j_{N_\mathrm{s}}}$	Binary variable denoting whether $Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}}$ is an association in the final assignment
$p_{\mathrm{d}\ell}$	Detection probability for sensor ℓ
m	Number of top solutions to be kept in the sequential m-best 2D assignment algorithm
$\mathbf{K}$	Set of all $\mathbf{k}_{\ell j}$
$\mathbf{k}_{\ell j}$	Association variable of $\mathbf{z}_{\ell j}$ in the UGM formulation
π_i	Mixing coefficient of the uniform-Gaussian mixture
$\boldsymbol{\kappa}$	Set of all $\kappa_{\ell j}$
$\kappa_{\ell j}$	Association variable of $\mathbf{z}_{\ell j}$ in the PPP formulation

The number of targets and their locations are unknown quantities of interest, to be estimated. A total number of N_{s} stationary sensors with known locations at

$$\mathbf{S} = (\mathrm{S}_1,\, \mathrm{S}_2,\, \ldots,\, \mathrm{S}_{N_{\mathrm{s}}}) = \left(\begin{bmatrix} S_{x_1} \\ S_{y_1} \end{bmatrix},\, \begin{bmatrix} S_{x_2} \\ S_{y_2} \end{bmatrix},\, \ldots,\, \begin{bmatrix} S_{x_{N_{\mathrm{s}}}} \\ S_{y_{N_{\mathrm{s}}}} \end{bmatrix} \right) \tag{3.2.3}$$

are able to observe transient acoustic events that occurred at target locations at the emission times and measure the bearings to these targets and the time of arrival of the observed acoustic signals. For events and measurements which are separated significantly in time, there is no data association ambiguity, so it is assumed that only measurements falling within a certain time window of interest need to be associated. Let n_ℓ denote the number of such measurements (one measurement is defined as a vector consisting of both a bearing and a time of arrival due to one acoustic signal in this context) obtained by the ℓ-th sensor within the time window.

The j-th measurement (a direction of arrival and time of arrival) received by the ℓ-th sensor, if it corresponds to the event at t_i^{e} from the i-th target, is

$$\mathbf{z}_{\ell j}(\mathrm{T}_i, t_i^{\mathrm{e}}) = \mathbf{h}_\ell\left(\mathrm{T}_i, t_i^{\mathrm{e}}\right) + \mathbf{w}_{\ell j} \quad i = 1, \ldots, N; \quad \ell = 1, \ldots, N_{\mathrm{s}}; \quad j = 1, \ldots, n_\ell \tag{3.2.4}$$

where $\mathbf{w}_{\ell j}$ is a zero mean white Gaussian measurement noise with known covariance matrix R_ℓ and

$$\mathbf{h}_\ell\left(\mathrm{T}_i, t_i^{\mathrm{e}}\right) = \begin{bmatrix} \theta_{\ell i} \\ t_{\ell i} \end{bmatrix} = \begin{bmatrix} \arctan\left[\dfrac{T_{y_i} - S_{y_\ell}}{T_{x_i} - S_{x_\ell}}\right] \\ t_i^{\mathrm{e}} + \dfrac{\sqrt{(T_{x_i} - S_{x_\ell})^2 + (T_{y_i} - S_{y_\ell})^2}}{c} \end{bmatrix} \tag{3.2.5}$$

where t_i^{e} is the unknown emission time of the acoustic signal from T_i and c is the known

speed of sound.

To incorporate false alarms, we denote a clutter target (with index 0) as T_0. A false measurement detected by the ℓ-th sensor consists of a bearing θ_0, which is uniformly distributed in the field of view of the ℓ-th sensor, and its arrival time t_0, which is uniformly distributed in the interval $[0, W]$. The number of false alarms from each sensor is assumed to be a Poisson random variable[2] with mean

$$N_{\mathrm{fa}} = \lambda_{\mathrm{fa}} \Phi W \tag{3.2.6}$$

where Φ is the range of field of view and is assumed to be the same for each sensor and λ_{fa} can be interpreted as the temporal-spatial density.

The pdf of measurement j from sensor ℓ — the likelihood function [4] of the target location and its emission time based on the measurement[3] — is

$$p(\mathbf{z}_{\ell j}|\mathrm{T}_0) = p(\theta_0)p(t_0) = \frac{1}{\Phi W} \triangleq \Lambda(\mathrm{T}_0; \mathbf{z}_{\ell j}) \tag{3.2.7}$$

$$\begin{aligned} p(\mathbf{z}_{\ell j}|\mathrm{T}_i, t_i^{\mathrm{e}}) &= |2\pi R_\ell|^{-1/2} \exp\left\{ -\frac{1}{2} \left[\mathbf{z}_{\ell j} - \mathbf{h}_\ell(\mathrm{T}_i, t_i^{\mathrm{e}})\right]' R_\ell^{-1} \left[\mathbf{z}_{\ell j} - \mathbf{h}_\ell(\mathrm{T}_i, t_i^{\mathrm{e}})\right] \right\} \\ &\triangleq \Lambda(\mathrm{T}_i, t_i^{\mathrm{e}}; \mathbf{z}_{\ell j}) \qquad i = 1, \ldots, N \end{aligned} \tag{3.2.8}$$

where (4.3.9) is the pdf of a measurement from the clutter (a false alarm), and (4.3.10) is the pdf of a measurement from a true target.

The problem is to estimate N and $\mathbf{T} = \{\mathrm{T}_i, i = 1, \ldots, N\}$ given the complete set of observations $\mathbf{Z} = \{\mathbf{z}_{\ell j}, \ell = 1, \ldots, N_{\mathrm{s}}; j = 1, \ldots, n_\ell\}$ in the presence of missed detections and false alarms and without the knowledge of the true data association.

[2]While for targets we consider two measurement models (Bernoulli and Poisson), for clutter only a Poisson model is considered.

[3]If the source is clutter, it has no emission time, only an arrival time.

3.3 The S-D assignment algorithm

3.3.1 Formulation

The S-D assignment problem formulation assumes a Bernoulli measurement model that the number of measurements from a real target received by a sensor is a Bernoulli random variable. Note that the number of false alarms is modeled as a Poisson random variable.

An augmented list of measurements at the ℓ-th sensor is defined as

$$\mathbf{Z}_\ell \triangleq \{\mathbf{z}_{\ell 0}, \ldots, \mathbf{z}_{\ell n_\ell}\} \tag{3.3.1}$$

where $\mathbf{z}_{\ell 0}$ is a dummy measurement[4] representing missed detections. An association of N_s measurements (N_s-tuple) consisting of one measurement from each augmented list will be denoted as

$$Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}} = \left\{\mathbf{z}_{1 j_1}, \mathbf{z}_{2 j_2}, \ldots, \mathbf{z}_{N_\mathrm{s} j_{N_\mathrm{s}}}\right\} \tag{3.3.2}$$

where $j_\ell \in \{0, 1, \ldots, n_\ell\}$ represents the index of the measurement from the augmented list $\mathbf{Z}_\ell$ which is included in the association.[5]

Assuming that the measurements in $Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}}$ originated from the same target at the location T_i and emission time t_i^e, the cost of this association will be given by the (physically dimensionless) negative log-likelihood ratio

$$c_{j_1 j_2 \ldots j_{N_\mathrm{s}}} = -\ln \frac{\Lambda(\mathrm{T}_i, t_i^\mathrm{e}; Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}})}{\Lambda(\mathrm{T}_0; Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}})} \tag{3.3.3}$$

where the numerator is calculated based on (4.3.10) and the denominator (the likelihood

[4] Please see Figure 2 in [20] for the illustration of dummy measurement and N_s-tuple.

[5] Recall that $j_\ell = 0$ represents the dummy measurement, so (3.3.2) need not contain N_s "real" measurements, i.e., missed detections are allowed in the association.

that they are all false) is calculated using (4.3.9).

Assuming the measurements are (conditioned on the true target locations) independent across the sensors, i.e., uncorrelated measurement noises, the likelihood function that the measurements in $Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}}$ originated from the same target at the location T_i and emission time t_i^{e} is

$$\Lambda(\mathrm{T}_i, t_i^{\mathrm{e}}; Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}}) = \prod_{\ell=1}^{N_\mathrm{s}} (1 - p_{\mathrm{d}\ell})^{1-u(j_\ell)} \left(p_{\mathrm{d}\ell}\, p(\mathbf{z}_{\ell j_\ell} | \mathrm{T}_i, t_i^{\mathrm{e}})\right)^{u(j_\ell)} \tag{3.3.4}$$

where $p_{\mathrm{d}\ell}$ is the probability of detection for the ℓ-th sensor (assumed the same for each real target) and the indicator function $u(j_\ell)$ is

$$u(j_\ell) \triangleq \begin{cases} 0 & \text{if } j_\ell = 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.3.5}$$

Since the target location T_i and the emission time t_i^{e} are unknown, we replace them by their maximum likelihood (ML) estimates $\hat{\mathrm{T}}_i$ and $\hat{t}_i^{\mathrm{e}}$ that are obtained by maximizing (3.3.4), that is

$$\hat{\mathrm{T}}_i, \hat{t}_i^{\mathrm{e}} = \arg \max_{\mathrm{T}_i, t_i^{\mathrm{e}}} \Lambda(\mathrm{T}_i, t_i^{\mathrm{e}}; Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}}) \tag{3.3.6}$$

Therefore, (3.3.3) is modified to a *generalized* negative log-likelihood ratio given by

$$c_{j_1 j_2 \ldots j_{N_\mathrm{s}}} = -\ln \frac{\Lambda(\hat{\mathrm{T}}_i, \hat{t}_i^{\mathrm{e}}; Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}})}{\Lambda(\mathrm{T}_0; Z_{j_1 j_2 \ldots j_{N_\mathrm{s}}})} \tag{3.3.7}$$

The likelihood that all the measurements in $Z_{j_1 j_2 \ldots j_{N_s}}$ are false alarms is

$$\Lambda(\mathrm{T}_0; Z_{j_1 j_2 \ldots j_{N_s}}) = \prod_{\ell=1}^{N_s} \left(\frac{1}{\Phi W} \right)^{u(j_\ell)} \tag{3.3.8}$$

The assignment problem is formulated as

$$\min_{\rho_{j_1 j_2 \ldots j_{N_s}}} \sum_{j_1=0}^{n_1} \sum_{j_2=0}^{n_2} \cdots \sum_{j_{N_s}=0}^{n_{N_s}} c_{j_1 j_2 \ldots j_{N_s}} \rho_{j_1 j_2 \ldots j_{N_s}} \tag{3.3.9}$$

subject to

$$\sum_{j_2=0}^{n_2} \sum_{j_3=0}^{n_3} \cdots \sum_{j_{N_s}=0}^{n_{N_s}} \rho_{j_1 j_2 \ldots j_{N_s}} = 1 \qquad j_1 = 1, 2, \ldots, n_1 \tag{3.3.10}$$

$$\sum_{j_1=0}^{n_1} \sum_{j_3=0}^{n_3} \cdots \sum_{j_{N_s}=0}^{n_{N_s}} \rho_{j_1 j_2 \ldots j_{N_s}} = 1 \qquad j_2 = 1, 2, \ldots, n_2 \tag{3.3.11}$$

$$\vdots \qquad\qquad \vdots$$

$$\sum_{j_1=0}^{n_1} \sum_{j_2=0}^{n_2} \cdots \sum_{j_{N_s-1}=0}^{n_{N_s-1}} \rho_{j_1 j_2 \ldots j_{N_s}} = 1 \qquad j_{N_s} = 1, 2, \ldots, n_{N_s} \tag{3.3.12}$$

where $\rho_{j_1 j_2 \ldots j_{N_s}} \in \{0, 1\}$ and $\rho_{j_1 j_2 \ldots j_{N_s}} = 1(0)$ means $Z_{j_1 j_2 \ldots j_{N_s}}$ is (not) an association in the final assignment.

Note that if $c_{j_1 j_2 \ldots j_{N_s}} > 0$, then $Z_{j_1 j_2 \ldots j_{N_s}}$ will not be an association in the final assignment. Since the overall cost will be smaller from the decision that all the real measurements in $Z_{j_1 j_2 \ldots j_{N_s}}$ are false (cost=0) than that they are from the same real target.

The ℓ-th constraint set in (3.3.10) – (3.3.12)

$$\sum_{j_1=0}^{n_1} \cdots \sum_{j_{\ell-1}=0}^{n_{\ell-1}} \sum_{j_{\ell+1}=0}^{n_{\ell+1}} \cdots \sum_{j_{N_s}=0}^{n_{N_s}} \rho_{j_1 j_2 \ldots j_{N_s}} = 1 \qquad j_\ell = 1, 2, \ldots, n_\ell \tag{3.3.13}$$

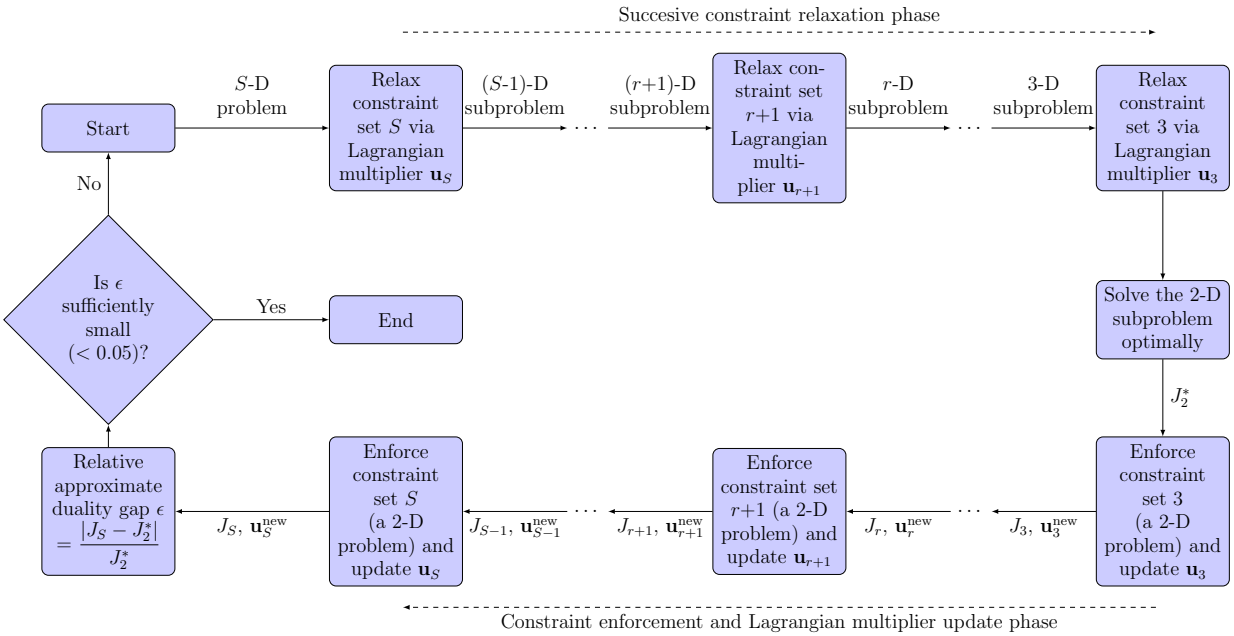


FIGURE 3.3.1: Flow chart of the Lagrangian relaxation based S-D assignment algorithm

enforces that each measurement (except the dummy) is associated with a single measurement from each other list, yielding a "target". Once the minimization problem (3.3.9) is solved, based on the assumption that each target is associated with one and only one measurement in each sensor list (including the dummy measurement), the number of associations will be equal to the number of targets (some will be real and some false). Associations with less than τ real measurements will be considered as from the clutter. The remaining associations will be deemed from real targets. The corresponding locations and emission times will be the ML estimates as obtained in (3.3.6).

3.3.2 The optimization via Lagrangian relaxation

The optimization problem (3.3.9) is NP hard when $N_{\text{s}} \geq 3$. One suboptimal algorithm is the Lagrangian relaxation based S-D assignment algorithm as shown in Figure 3.3.1,

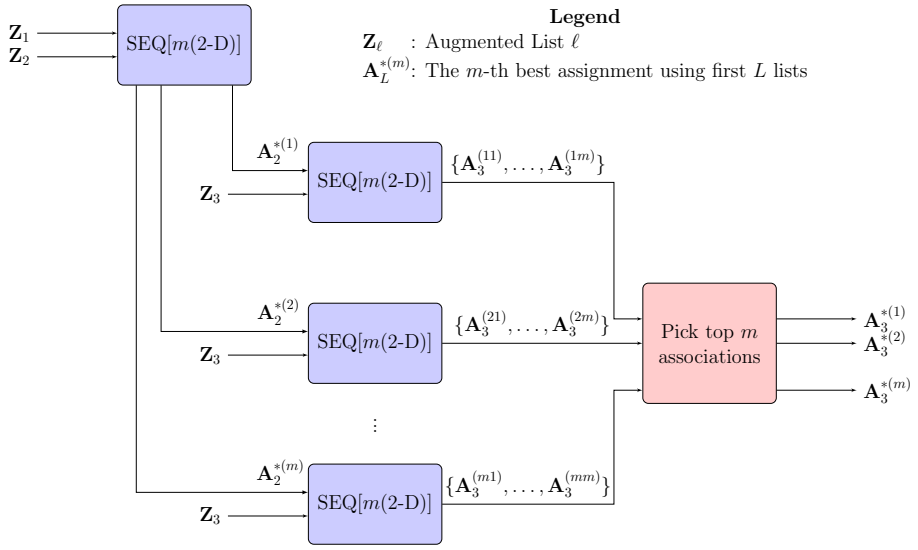


FIGURE 3.3.2: Initial iteration of the sequential m-best 2-D assignment algorithm

which solves the original problem as a series of relaxed 2-D subproblems. The r-th ($r = N_\mathrm{s}, N_\mathrm{s} - 1, \ldots, 3$) constraint set are successively relaxed and appended to the cost with Lagrange multipliers $\mathbf{u}_r$. At stage $r = 3$ one has a 2-D problem, which can be optimally[6] solved using the modified Auction algorithm.

The constraint sets are then reimposed one at a time ($r = 3, 4, \ldots, N_\mathrm{s}$), and the corresponding Lagrange multipliers are updated to $\mathbf{u}_r^{\mathrm{new}}$; at each stage the cost J_r of the resulting feasible solution is computed, until all constraint sets are met. The duality gap — difference between the cost J_2^* from the maximally relaxed problem and J_S from the fully constrained one — is calculated and the iterations continue until this gap is small enough (usually 5% of the cost from the fully constrained one). See [18] [48] for the detailed description.

3.3.3 The sequential m-best 2-D assignment algorithm

When $N_{\rm s} = 2$, (3.3.9) becomes a 2-D assignment problem. By using Murty's ranking algorithm [43], one can find the top m best assignments instead of only the best one. The sequential m-best 2-D assignment algorithm can be described as follows. Initially, one selects two lists of measurements and obtains the top m best 2-D assignments with each assignment being a set of 2-tuples. Next, for each of these one continues to solve an m-best 2-D assignment, which yields a set of 3-tuples, between any one of the previous m association results and a third list of measurement. After this second step, one has m^2 assignments available, out of which the top m solutions in terms of the association cost will be selected for the next step. This procedure (shown in Figure 3.3.2) is repeated until all the $N_{\rm s}$ lists of measurements are processed and the final assignment will be a set of $N_{\rm s}$-tuples.

Note that it is possible to have an association $Z_{j_1 j_2 \cdots j_{N_{\rm s}}}$ with $c_{j_1 j_2 \cdots j_{N_{\rm s}}} > 0$ in the final assignment once the sequential m-best 2-D algorithm terminates. Such associations will be discarded before any association with less than τ real measurements is removed.

3.4 Uniform-Gaussian Mixture (UGM) Formulation

If one assumes a Poisson measurement model that the number of measurements from a real target received by a sensor is a Poisson random variable, then one can model a list of measurements as realizations of a random variable with a uniform-Gaussian mixture density [6] or a Poisson point process. The uniform-Gaussian mixture formulation is presented in this section and the Poisson point process formulation will be presented in next section.

[6] Up to the rounding error, i.e., quasi-optimally

3.4.1 Formulation

Assume (temporarily) the number of targets, N, is given. Since the association between a measurement $\mathbf{z}$ (without subscript, for simplicity) and the targets is unknown, we introduce an $(N+1)$-dimensional random binary-valued association vector

$$\mathbf{k} = [k_0,\, k_1,\, \ldots,\, k_N] \tag{3.4.1}$$

to indicate the target from which the measurement $\mathbf{z}$ originates. In our formulation, the random variable $\mathbf{z}$ is observed. The random variable $\mathbf{k}$ is not observed, thus is called a latent variable[7]. The entries k_i of the vector $\mathbf{k}$ satisfy the following conditions

$$\sum_{i=0}^{N} k_i = 1 \tag{3.4.2}$$

$$k_i \in \{0,\, 1\} \quad \forall\, i \tag{3.4.3}$$

that is, there are $N+1$ possible values for the vector $\mathbf{k}$. Let us use $\mathbf{e}_i$ to denote the $(N+1)$-dimensional vector with one in its i-th entry and zeros elsewhere. The event $\{\mathbf{k} = \mathbf{e}_1\}$, which is the same as the event $\{k_0 = 1\}$, means that $\mathbf{z}$ is a clutter-originated measurement. The event $\{\mathbf{k} = \mathbf{e}_{i+1}\}$ with $i > 0$, which is the same as $\{k_i = 1\}$, means that the measurement $\mathbf{z}$ originated from the i-th target.

The prior probability that $\mathbf{z}$ originated from the i-th target given that the acoustic signal has been detected (assuming that a detected acoustic signal originates equally likely from

[7]Latent variables are random variables whose values we do not observe or measure.

all the targets) is

$$p(\mathbf{k}=\mathbf{e}_{i+1}) = p(k_i = 1) \triangleq \pi_i = \frac{p_{\mathrm{d}}(\mathrm{T}_i)}{\sum_{i=0}^{N} p_{\mathrm{d}}(\mathrm{T}_i)} \quad i = 0, 1, \ldots, N \tag{3.4.4}$$

where $p_{\mathrm{d}}(\mathrm{T}_i)$ is the probability of detection for the real target i ($i \neq 0$) and is assumed to be the same at each sensor and

$$p_{\mathrm{d}}\left(\mathrm{T}_0\right) = N_{\mathrm{fa}} \tag{3.4.5}$$

With abuse of notation, (4.3.13) is the expected number of false alarms at each sensor. The probabilities π_i, therefore, satisfy the following two conditions

$$0 \leq \pi_i \leq 1 \tag{3.4.6}$$

$$\sum_{i=0}^{N} \pi_i = 1 \tag{3.4.7}$$

Because of (3.4.2) and (3.4.3), the prior probability in (3.4.4) can be equivalently expressed, in the form of a probability mass function, as

$$p(\mathbf{k}=\mathbf{e}_{i+1}) = p(k_i = 1) = \pi_i = \prod_{i=0}^{N} \pi_i^{k_i} \tag{3.4.8}$$

where the last equality holds because only the exponent k_i is equal to 1 while all other exponents are equal to 0, and thus do not affect the product.

From (4.3.9) and (4.3.10), the conditional pdf of a measurement $\mathbf{z}$ (without subscript,

for simplicity) obtained by the ℓ-th sensor given that it is associated with the i-th target is

$$p_\ell(\mathbf{z}|k_0 = 1, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) = \frac{1}{\Phi W} \tag{3.4.9}$$

$$p_\ell(\mathbf{z}|k_i = 1, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) = \mathcal{N}\left(\mathbf{z}; \mathbf{h}_\ell(\mathrm{T}_i, t_i^{\mathrm{e}}), R_\ell\right) \qquad i = 1, \ldots, N \tag{3.4.10}$$

For notational simplicity, let us denote[8]

$$g_{\ell i}(\mathbf{z}) = p_\ell(\mathbf{z}|k_i = 1, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) \qquad i = 0, 1, \ldots, N \tag{3.4.11}$$

In a similar way as we derived (3.4.8), using (3.4.2) and (3.4.3) we have

$$p_\ell(\mathbf{z}|\mathbf{k} = \mathbf{e}_{i+1}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) = p_\ell(\mathbf{z}|k_i = 1, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) = g_{\ell i}(\mathbf{z}) = \prod_{i=0}^{N} \left(g_{\ell i}(\mathbf{z})\right)^{k_i} \tag{3.4.12}$$

The joint density of a measurement $\mathbf{z}$ from sensor ℓ and its association vector $\mathbf{k}$ is therefore

$$p_\ell(\mathbf{z}, \mathbf{k} = \mathbf{e}_{i+1}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}) = p_\ell(\mathbf{z}|\mathbf{k} = \mathbf{e}_{i+1}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) p(\mathbf{k} = \mathbf{e}_{i+1}) = \prod_{i=0}^{N} \left(\pi_i \, g_{\ell i}(\mathbf{z})\right)^{k_i} = \pi_i \, g_{\ell i}(\mathbf{z}) \tag{3.4.13}$$

where the first equality holds because of the conditional probability definition; the second equality holds as a result of direct substitutions of $p(\mathbf{k} = \mathbf{e}_{i+1})$ from (3.4.8) and $p_\ell(\mathbf{z}|\mathbf{k} = \mathbf{e}_{i+1}, \mathbf{T}, \mathbf{t}^{\mathrm{e}})$ from (3.4.12); and the last equality holds because only the exponent k_i is equal to 1 and other exponents are zero. The marginal density of $\mathbf{z}$ is then obtained by summing

[8] $g_{\ell i}(\mathbf{z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}})$ will be used when the conditioning needs to be explicitly indicated.

the joint density over all the $N+1$ values of $\mathbf{k}$ as

$$p_\ell(\mathbf{z}|\mathbf{T},\mathbf{t}^{\mathrm{e}}) = \sum_{i=0}^{N} p_\ell(\mathbf{z},\mathbf{k}=\mathbf{e}_{i+1}|\mathbf{T},\mathbf{t}^{\mathrm{e}}) = \sum_{i=0}^{N} \pi_i\, g_{\ell i}(\mathbf{z}) \tag{3.4.14}$$

where the first equality holds because of the total probability theorem and the second equality holds as a result of substitution of $p_\ell(\mathbf{z},\mathbf{k}=\mathbf{e}_{i+1}|\mathbf{T},\mathbf{t}^{\mathrm{e}})$ from (3.4.13). Therefore, the marginal density of one measurement is a mixture (termed as "uniform-Gaussian" mixture in this chapter) of one uniform density and N Gaussian densities with the parameters π_i being the mixing coefficients. The conditional density of $\mathbf{k}$ given $\mathbf{z}$ is obtained using Bayes' theorem as

$$p_\ell(\mathbf{k}|\mathbf{z},\mathbf{T},\mathbf{t}^{\mathrm{e}}) = \frac{p_\ell(\mathbf{z},\mathbf{k}|\mathbf{T},\mathbf{t}^{\mathrm{e}})}{p_\ell(\mathbf{z}|\mathbf{T},\mathbf{t}^{\mathrm{e}})} = \frac{\prod_{i=0}^{N}\left(\pi_i\, g_{\ell i}(\mathbf{z})\right)^{k_i}}{\sum_{i=0}^{N}\pi_i\, g_{\ell i}(\mathbf{z})} \tag{3.4.15}$$

which is equivalent to

$$P(k_i=1|\mathbf{z},\mathbf{T},\mathbf{t}^{\mathrm{e}}) = \frac{\pi_i\, g_{\ell i}(\mathbf{z})}{\sum_{i=0}^{N}\pi_i\, g_{\ell i}(\mathbf{z})} \tag{3.4.16}$$

Let (with $\mathbf{k}$ indexed as in (4.3.6))

$$\mathbf{K} = \{\mathbf{k}_{\ell j}, \ell = 1,2,\ldots,N_{\mathrm{s}}; j = 1,2,\ldots,n_\ell\} \tag{3.4.17}$$

be the corresponding set of association vectors (or latent variables) for $\mathbf{Z}$ and

$$\boldsymbol{\pi} = \{\pi_i, i = 0,\ 1,\ \ldots,\ N\} \tag{3.4.18}$$

From (3.4.13), the conditional independence of measurements across all the sensors yields

the joint density of $\mathbf{Z}$ and $\mathbf{K}$

$$p(\mathbf{Z}, \mathbf{K}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}) = \prod_{\ell=1}^{N_{\mathrm{s}}} \prod_{j=1}^{n_\ell} \prod_{i=0}^{N} \left(\pi_i \, g_{\ell i}(\mathbf{z}_{\ell j})\right)^{\left[k_{\ell j}\right]_i} \tag{3.4.19}$$

where $\left[k_{\ell j}\right]_i$ is the i-th component of the association vector $\mathbf{k}_{\ell j}$. The marginal density of $\mathbf{Z}$ is obtained by summing the joint density (3.4.19) over all possible values of $\mathbf{K}$ as

$$p\left(\mathbf{Z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}\right) = \prod_{\ell=1}^{N_{\mathrm{s}}} \prod_{j=1}^{n_\ell} \left(\sum_{i=0}^{N} \pi_i \, g_{\ell i}(\mathbf{z}_{\ell j}) \right) \tag{3.4.20}$$

and the posterior density (actually pmf since $\mathbf{K}$ is discrete) of $\mathbf{K}$ conditioned on $\mathbf{Z}$ is

$$p(\mathbf{K}|\mathbf{Z}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) = \prod_{\ell=1}^{N_{\mathrm{s}}} \prod_{j=1}^{n_\ell} \frac{\prod_{i=0}^{N} \left(\pi_i g_{\ell i}(\mathbf{z}_{\ell j})\right)^{\left[k_{\ell j}\right]_i}}{\sum_{i=0}^{N} \pi_i \, g_{\ell i}(\mathbf{z}_{\ell j})} \tag{3.4.21}$$

3.4.2 The Expectation-Maximization Algorithm

We are interested in finding the ML estimates of $\mathbf{T}$ and $\mathbf{t}^{\mathrm{e}}$ that maximize $p\left(\mathbf{Z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}\right)$ or $\ln p\left(\mathbf{Z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}\right)$. However, it is difficult to obtain these estimates since the data association between $\mathbf{Z}$ and $\mathbf{T}$ is unknown, that is, $\mathbf{K}$ is not observed. Mathematically, setting the derivatives of $\ln p\left(\mathbf{Z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}\right)$ with respect to $\mathbf{T}$ and $\mathbf{t}^{\mathrm{e}}$ does not lead to a closed-form solution, which suggests an iterative approach.

The expectation-maximization (EM) algorithm [19] is a two-step iterative optimization technique to find the maximum likelihood estimate from incomplete data. In this context, $\{\mathbf{Z}, \mathbf{K}\}$ are the complete data set and the observed data $\mathbf{Z}$ is the incomplete data available since the association variables in $\mathbf{K}$ are unknown.

Each iteration of the EM algorithm has an expectation step (E step) and a maximization step (M step). In the E step, we use temporary estimates of $\mathbf{T}$ and $\mathbf{t}^{\mathrm{e}}$ to find the posterior distribution in (3.4.21) to "learn" about $\mathbf{K}$. We then use this posterior distribution of $\mathbf{K}$ to find the expectation of the joint density of $\mathbf{Z}$ and $\mathbf{K}$ in (3.4.19). In the M step, we maximize the expectation obtained in the E step to obtain updated estimates of $\mathbf{T}$ and $\mathbf{t}^{\mathrm{e}}$ [6].

3.4.3 Optimization

Initialization

The EM algorithm is an iterative method. The first step is to initialize the parameters $\mathbf{T}$, $\mathbf{t}^{\mathrm{e}}$ and $\boldsymbol{\pi}$. Here we assume that the mixing coefficients π_i are scalar quantities that need to be estimated along with T_i and t_i^{e}.

The EM algorithm guarantees that $p\left(\mathbf{Z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}\right)$ increases at each iteration. However, a poor initialization can cause convergence to a local maximum as opposed to the global one. As shown later, because of the relationship between π_i and $p_{\mathrm{d}}(\mathrm{T}_i)$ in (3.4.4), the iterative procedures depends on whether N_{fa} is known. In either case, $p_{\mathrm{d}}(\mathrm{T}_i)$ is initialized to be 1 and the initial values of π_i will be calculated according to (3.4.4). In this chapter, three initialization approaches for $\mathbf{T}$ and $\mathbf{t}^{\mathrm{e}}$ are considered and will be discussed in the Section 3.6.4.

E step

Let $\mathbf{T}^{(n-1)}$, $\mathbf{t}^{\mathrm{e},(n-1)}$ and $\boldsymbol{\pi}^{(n-1)}$ denote the estimates from the previous step. In the expectation step, we compute $p(\mathbf{K}|\mathbf{Z}, \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ and evaluate the expected value of

$\ln p(\mathbf{Z}, \mathbf{K}|\mathbf{T}, \mathbf{t}^{\mathrm{e}})$ conditioned on $p(\mathbf{K}|\mathbf{Z}, \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$, which is given by

$$\begin{aligned} Q(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) &= \mathbb{E}\left[\ln\left(p(\mathbf{Z}, \mathbf{K} \left| \mathbf{T}, \mathbf{t}^{\mathrm{e}} \right.) \right) \middle| p(\mathbf{K} \left| \mathbf{Z}, \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)} \right.) \right] \\ &= Q_T(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) + Q_\pi(\boldsymbol{\pi} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) \end{aligned} \tag{3.4.22}$$

where

$$Q_T(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) = \sum_{\ell=1}^{N_{\mathrm{s}}} \sum_{j=1}^{n_\ell} \sum_{i=0}^{N} \ln\left(g_{\ell i}(\mathbf{z}_{\ell j}) \right) w_{\ell j i}^{(n-1)} \tag{3.4.23}$$

$$Q_\pi(\boldsymbol{\pi} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) = \sum_{\ell=1}^{N_{\mathrm{s}}} \sum_{j=1}^{n_\ell} \sum_{i=0}^{N} \ln\left(\pi_i \right) w_{\ell j i}^{(n-1)} \tag{3.4.24}$$

where

$$w_{\ell j i}^{(n-1)} = \frac{\pi_i^{(n-1)} g_{\ell i}^{(n-1)}(\mathbf{z}_{\ell j})}{\sum_{i=0}^{N} \pi_i^{(n-1)} \, g_{\ell i}^{(n-1)}(\mathbf{z}_{\ell j})} \tag{3.4.25}$$

$$g_{\ell i}^{(n-1)}(\mathbf{z}_{\ell j}) = p_\ell(\mathbf{z}_{\ell j} | k_i = 1, \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) \tag{3.4.26}$$

$w_{\ell j i}^{(n-1)}$ is the posterior probability that the measurement $\mathbf{z}_{\ell j}$ originates from the i-th target, given that the target locations are $\mathbf{T}^{(n-1)}$ and the emission times are $\mathbf{t}^{\mathrm{e},(n-1)}$.

M Step

In the maximization step, we maximize $Q(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ over all feasible $\mathbf{T}$, $\mathbf{t}^{\mathrm{e}}$ and $\boldsymbol{\pi}$. Inspection of (3.4.22) reveals that $Q_T(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ depends only on the locations $\mathbf{T}$ and $Q_\pi(\boldsymbol{\pi} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ depends only on detection probabilities through mixing coefficients. Therefore, maximization of $Q(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ can be done by maximizing $Q_T(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ and $Q_\pi(\boldsymbol{\pi} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ separately.

We define $Q_T(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ as

$$Q_T(\mathbf{T}, \mathbf{t}^{\mathrm{e}} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) = \sum_{i=0}^{N} Q_{\mathrm{T}_i}(\mathrm{T}_i \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) \tag{3.4.27}$$

where

$$Q_{\mathrm{T}_i}(\mathrm{T}_i \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) = \sum_{\ell=1}^{N_{\mathrm{s}}} \sum_{j=1}^{n_\ell} \ln\left(g_{\ell i}(\mathbf{z}_{\ell j})\right) w_{\ell j i}^{(n-1)} \tag{3.4.28}$$

Note that $Q_{\mathrm{T}_0}(\mathrm{T}_0 \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ is a constant, and there is no functional relation between T_{i_1} and T_{i_2} for $i_1 \neq i_2$ Therefore, each target location T_i can be obtained separately by maximizing $Q_{\mathrm{T}_i}(\mathrm{T}_i \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$.

Next we maximize $Q_\pi(\boldsymbol{\pi} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ with respect to π_i, while accounting for the constraint that the mixing coefficients sum to one. This can be achieved using a Lagrange multiplier λ and maximizing the following quantity

$$Q_\pi^L(\boldsymbol{\pi} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) = \sum_{\ell=1}^{N_{\mathrm{s}}} \sum_{j=1}^{n_\ell} \sum_{i=0}^{N} \ln\left(\pi_i\right) w_{\ell j i}^{(n-1)} + \lambda \left(\sum_{i=0}^{N} \pi_i - 1 \right) \tag{3.4.29}$$

which gives

$$\pi_i{}^{(n)} = \frac{\sum_{\ell=1}^{N_\mathrm{s}} \sum_{j=1}^{n_\ell} w_{\ell ji}^{(n-1)}}{\sum_{\ell=1}^{N_\mathrm{s}} \sum_{j=1}^{n_\ell} \sum_{i=0}^{N} w_{\ell ji}^{(n-1)}} \tag{3.4.30}$$

When N_fa is unknown, one can set $p_\mathrm{d}(\mathrm{T}_i)$ and N_fa based on (3.4.30) as follows

$$j = \arg\max_{i, i\neq 0} \pi_i \tag{3.4.31}$$

$$p_\mathrm{d}(\mathrm{T}_i) = \frac{\pi_i}{\pi_j}, i \neq j \tag{3.4.32}$$

$$N_\mathrm{fa} = \frac{\pi_0}{\pi_j} \tag{3.4.33}$$

which guarantees the constraints

$$p_\mathrm{d}(\mathrm{T}_i) \leq 1, i > 0 \tag{3.4.34}$$

However, when N_fa is known, it is not always possible to find $p_\mathrm{d}(\mathrm{T}_i) \leq 1$ such that (3.4.30) holds. For instance, the following may not hold for π_0 from (3.4.30) and a given N_fa

$$\pi_0 = \frac{N_\mathrm{fa}}{\sum_{i=1}^{N} p_\mathrm{d}(\mathrm{T}_i) + N_\mathrm{fa}} \geq \frac{N_\mathrm{fa}}{N + N_\mathrm{fa}} \tag{3.4.35}$$

We need to maximize $Q_\pi(\boldsymbol{\pi} \big| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ with respect to $p_\mathrm{d}(\mathrm{T}_i)$ subject to (3.4.34).

The Karush-Kuhn-Tucker (KKT) conditions [36] give rise to the following proposition (see Appendix for proof):

Proposition 3.4.1. *Let*

$$\mathcal{S} = \left\{ i \left| \sum_{\ell=1}^{N_\mathrm{s}} \sum_{j=1}^{n_\ell} w_{\ell ji}^{(n-1)} N_\mathrm{fa} > \sum_{\ell=1}^{N_\mathrm{s}} \sum_{j=1}^{n_\ell} w_{\ell j0}^{(n-1)} \right. \right\} \tag{3.4.36}$$

which can be an empty set, and its cardinality is denoted by $|\mathcal{S}|$*. The optimal values of* $p_\mathrm{d}(\mathrm{T}_i)$ *are given by*

$$p_\mathrm{d}^{(n)}(\mathrm{T}_i) = \begin{cases} 1 & \text{if } i \in \mathcal{S} \\ \dfrac{\sum\limits_{\ell=1}^{N_\mathrm{s}} \sum\limits_{j=1}^{n_\ell} w_{\ell ji}^{(n-1)} (|\mathcal{S}| + N_\mathrm{fa})}{\sum\limits_{k>0, k\in\mathcal{S}} \sum\limits_{\ell=1}^{N_\mathrm{s}} \sum\limits_{j=1}^{n_\ell} w_{\ell jk}^{(n-1)} + \sum\limits_{\ell=1}^{N_\mathrm{s}} \sum\limits_{j=1}^{n_\ell} w_{\ell j0}^{(n-1)}} & \text{if } i \notin \mathcal{S} \end{cases} \tag{3.4.37}$$

Therefore, by (3.4.4)

$$\pi_i{}^{(n)} = \frac{p_\mathrm{d}^{(n)}(\mathrm{T}_i)}{\sum\limits_{i=0}^{N} p_\mathrm{d}^{(n)}(\mathrm{T}_i)} \tag{3.4.38}$$

which can be verified to be identical to (3.4.30) when the set $\mathcal{S}$ is empty.

The EM algorithm is terminated when the likelihood function (3.4.20) converges, that is

$$\left| \ln p\left(\mathbf{Z} | \mathbf{T}^{(n)}, \mathbf{t}^{\mathrm{e}(n)}\right) - \ln p\left(\mathbf{Z} \left| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right.\right) \right| \leq \epsilon \tag{3.4.39}$$

where ϵ is a small number (e.g., 10^{-3}).

One can use a fixed $\boldsymbol{\pi}$ throughout the EM iterations and skip the update process in (3.4.30) under the assumption that both the detection probabilities and the false alarm density are known.

3.4.4 Use of the Information Criterion for Cardinality Selection

So far we have assumed that the number of targets is given. Since the number of targets is unknown, we can use the above described procedure to estimate the parameters $\boldsymbol{\pi}$ and $\mathbf{T}$ given a specific cardinality, i.e., the number of targets N. Now we are faced with a cardinality selection problem, or a model selection problem where the dimensionality of a model is the number of targets. One of the most widely used criteria for model selection problems is the Bayesian information criterion (BIC) [51].

Let $M^k = \{\hat{\boldsymbol{\pi}}^k, \hat{\mathbf{T}}^k, \hat{N}^k\}$ denote the set of estimated parameters based on the k-th cardinality. According to BIC, we choose the model for which the following is largest

$$\ln p\left(\mathbf{Z}\left|M^k\right.\right) - \frac{1}{2} d^k \ln(N_{\mathbf{z}}) \tag{3.4.40}$$

where from (3.4.20)

$$p\left(\mathbf{Z}\left|M^k\right.\right) = \prod_{\ell=1}^{N_{\mathrm{s}}} \prod_{j=1}^{n_\ell} \left(\sum_{i=0}^{\hat{N}^k} \hat{\pi}_i^k \, g_{\ell i}(\mathbf{z}_{\ell j}|\hat{\mathbf{T}}^k) \right) \tag{3.4.41}$$

and $N_{\mathbf{z}}$ is the total number of measurements across all the sensors; d^k is the total number of parameters to be estimated based on the k-th cardinality. In our case, d^k is $4\hat{N}^k$+1 ($2\hat{N}^k$ position coordinates for a problem in 2D, $\hat{N}^k$ emission times, $\hat{N}^k$+1 uniform-Gaussian mixture coefficients including the expected number of false alarms) if it is assumed the detection probabilities and the false alarm density are unknown. If the detection probabilities and the false alarm density are assumed to be known, then d^k is $3\hat{N}^k$.

3.5 Poisson Point Process (PPP) Model

3.5.1 Formulation

Assume the number of targets, N, is given. Let $\mathbf{w} = \{w_i, i = 1, \ldots, N\}$ where

$$w_i = p_{\mathrm{d}}(\mathrm{T}_i) \tag{3.5.1}$$

are the detection probabilities. The number of measurements n_ℓ and $\{\mathbf{z}_{\ell j}, j = 1, 2, \ldots, n_\ell\}$ obtained at the ℓ-th sensor is jointly modeled as a realization of a PPP. The measurement set at ℓ-th the sensor is denoted as

$$\boldsymbol{\psi}_\ell = \{n_\ell,\, \mathbf{z}_{\ell 1},\, \mathbf{z}_{\ell 2},\, \ldots,\, \mathbf{z}_{\ell n_\ell}\} \tag{3.5.2}$$

In this case, the points $\mathbf{z}_{\ell j}$ occur in the space $\mathbb{S} = \{(\theta, t) : \theta \in [-\pi, \pi), t \in [0, W]\}$ and their order is irrelevant. The PPP is fully parameterized by its spatial intensity function

$$\mu_\ell(\mathbf{z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}) = \sum_{i=0}^{N} p_{\mathrm{d}}(\mathrm{T}_i)\, g_{\ell i}(\mathbf{z}) \tag{3.5.3}$$

where, similarly to (4.3.13),

$$p_{\mathrm{d}}\left(\mathrm{T}_0\right) = N_{\mathrm{fa}} \tag{3.5.4}$$

The number of points in the PPP is a Poisson random variable with rate $\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}$, that is, the probability mass function of n_ℓ is

$$p(n_\ell) = \frac{(\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z})^{n_\ell}}{n_\ell\,!} \exp\left\{ -\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z} \right\} \tag{3.5.5}$$

The n_ℓ points are defined as independent and identically distributed (i.i.d.) samples of a random variable with probability density function

$$p(\mathbf{z}) = \frac{\mu_\ell(\mathbf{z})}{\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}} = \frac{\sum_{i=0}^{N} p_{\mathrm{d}}(\mathrm{T}_i)\, g_{\ell i}(\mathbf{z})}{\sum_{i=0}^{N} p_{\mathrm{d}}(\mathrm{T}_i)} \tag{3.5.6}$$

The joint pmf-pdf of $\boldsymbol{\psi}^\ell$ is

$$p\left(\boldsymbol{\psi}^\ell\right) = \exp\left(-\int_{\mathbb{S}} \mu_\ell(\mathbf{z} \left| \mathbf{T}, \mathbf{t}^{\mathrm{e}} \right) d\mathbf{z} \right) \prod_{j=1}^{n_\ell} \mu_\ell(\mathbf{z}_{\ell j} \left| \mathbf{T}, \mathbf{t}^{\mathrm{e}} \right) \tag{3.5.7}$$

The factorial term $n_\ell\,!$ in (4.3.17) is canceled out because there are $n_\ell\,!$ permutations of an ordered list of measurements.

Let $\boldsymbol{\Psi}$ denote the set of all measurement sets (from the N_{s} sensors), i.e.,

$$\boldsymbol{\Psi} = \left\{ \boldsymbol{\psi}_1,\, \boldsymbol{\psi}_2,\, \ldots,\, \boldsymbol{\psi}_{N_{\mathrm{s}}} \right\} \tag{3.5.8}$$

The independence of the N_{s} measurement sets yields

$$p\left(\boldsymbol{\Psi} \left| \mathbf{T}, \mathbf{t}^{\mathrm{e}} \right.\right) = \prod_{\ell=1}^{N_{\mathrm{s}}} p\left(\boldsymbol{\psi}_\ell \left| \mathbf{T}, \mathbf{t}^{\mathrm{e}} \right.\right), \tag{3.5.9}$$

Since the intensity function is a mixture of uniform or Gaussian pdf and the association

is unknown, we model the latent association variables as conditionally independent random variables

$$\kappa_{\ell j} \in \{0, 1, 2, \ldots, N\} \tag{3.5.10}$$

that identify which component spawned the j-th measurement in the ℓ-th sensor. Here $\kappa_{\ell j} = 0$ indicates that the measurement is generated by the clutter. The set of latent variables for the ℓ-th sensor is denoted as

$$\kappa_\ell = \{\kappa_{\ell 1}, \ldots, \kappa_{\ell n_\ell}\} \tag{3.5.11}$$

such that the full set is

$$\boldsymbol{\kappa} = \{\kappa_1, \ldots, \kappa_{N_\mathrm{s}}\} \tag{3.5.12}$$

The latent association variables may be regarded as "marks" associated with each of the points in the PPP. Define a mark space

$$M \triangleq \{0, 1, 2, \ldots, N\} \tag{3.5.13}$$

Now

$$\boldsymbol{\psi}_\ell^{\mathrm{M}} = \{n_\ell, (\mathbf{z}_{\ell 1}, \kappa_{\ell 1}), \ldots, (\mathbf{z}_{\ell n_\ell}, \kappa_{\ell n_\ell})\} \tag{3.5.14}$$

denotes a realization of the marked PPP for the ℓ-th sensor, where "M" indicates that the associations are known ("marked"). Based on the Marking Theorem [41], the intensity

function of $\boldsymbol{\psi}_\ell^{\mathrm{M}}$ is

$$\mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa|\mathbf{T}, \mathbf{t}^{\mathrm{e}}) = \mu_\ell(\mathbf{z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}) p_\ell\left(\kappa|\mathbf{z}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}\right), \tag{3.5.15}$$

where $p_\ell\left(\kappa|\mathbf{z}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}\right)$ denotes the conditional probability of κ given $\mathbf{z}$ (without subscript here, for simplicity) with $\kappa = i$ indicating the probability of $\mathbf{z}$ originating from target i. Using the same reasoning as for the derivation of (3.4.4), we have the prior probability of κ

$$p_\ell(\kappa|\mathbf{T}, \mathbf{t}^{\mathrm{e}}) = \frac{w_\kappa}{\sum_{i=0}^{N} w_i} \tag{3.5.16}$$

Given that a point $\mathbf{z}$ in the PPP is associated with the κ-th mixture component, the conditional intensity becomes

$$\mu_\ell(\mathbf{z}|\kappa) = w_\kappa \, g_{\ell\kappa}\left(\mathbf{z}\right) \tag{3.5.17}$$

and the conditional density of $\mathbf{z}$ given κ is

$$p_\ell(\mathbf{z}|\kappa, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) = g_{\ell\kappa}(\mathbf{z}) \tag{3.5.18}$$

Using Bayes' theorem

$$p_\ell\left(\kappa|\mathbf{z}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}\right) = \frac{p_\ell(\mathbf{z}|\kappa, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) p_\ell(\kappa|\mathbf{T}, \mathbf{t}^{\mathrm{e}})}{p_\ell(\mathbf{z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}})} = \frac{w_\kappa g_{\ell\kappa}(\mathbf{z})}{\mu_\ell(\mathbf{z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}})} \tag{3.5.19}$$

Substituting (3.5.19) into (3.5.15) yields

$$\mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa|\mathbf{T}, \mathbf{t}^{\mathrm{e}}) = w_\kappa g_{\ell\kappa}(\mathbf{z}). \tag{3.5.20}$$

The joint probability density function of $\boldsymbol{\psi}_\ell^{\mathrm{M}}$ is, similarly to (4.3.19), given by

$$p\left(\boldsymbol{\psi}_\ell^{\mathrm{M}}\right) = \exp\left(\sum_{\kappa=0}^{N} - \int_{\mathbb{S}} \mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa \big| \mathbf{T}, \mathbf{t}^{\mathrm{e}})\, d\mathbf{z}\right) \prod_{j=1}^{n_\ell} \mu_\ell^{\mathrm{M}}(\mathbf{z}_{\ell j}\, \kappa_{\ell j} \big| \mathbf{T}, \mathbf{t}^{\mathrm{e}}) \tag{3.5.21}$$

Now let the complete data from (4.3.26) be

$$\boldsymbol{\Psi}^{\mathrm{M}} = \left\{\boldsymbol{\psi}_1^{\mathrm{M}},\, \boldsymbol{\psi}_2^{\mathrm{M}},\, \ldots,\, \boldsymbol{\psi}_{N_{\mathrm{s}}}^{\mathrm{M}}\right\}, \tag{3.5.22}$$

The conditional independence of the N_{s} measurement sets yields the pmf-pdf for the complete data

$$p\left(\boldsymbol{\Psi}^{\mathrm{M}} | \mathbf{T}, \mathbf{t}^{\mathrm{e}}\right) = \exp\left(-N_{\mathrm{s}} \sum_{i=0}^{N} w_i\right) \prod_{\ell=1}^{N_{\mathrm{s}}} \prod_{j=1}^{n_\ell} w_{\kappa_{\ell j}} g_{\ell \kappa_{\ell j}}\left(\mathbf{z}_{\ell j} | \mathbf{T}, \mathbf{t}^{\mathrm{e}}\right). \tag{3.5.23}$$

where we have used the fact

$$\sum_{\kappa=0}^{N} \left(\int w_\kappa g_{\ell\kappa}(\mathbf{z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}})\, d\mathbf{z}\right) = \sum_{i=0}^{N} w_i. \tag{3.5.24}$$

Dividing (4.3.30) by (4.3.21) leads to the density of the marks conditioned on the observed measurements and the unknown parameters

$$p(\boldsymbol{\kappa}|\mathbf{Z}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}) = \prod_{\ell=1}^{N_{\mathrm{s}}} \prod_{j=1}^{n_\ell} p_\ell\left(\kappa_{\ell j} \,|\, \mathbf{z}_{\ell j}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}\right), \tag{3.5.25}$$

where

$$p_\ell\left(\kappa_{\ell j} \,|\, \mathbf{z}_{\ell j}, \mathbf{T}, \mathbf{t}^{\mathrm{e}}\right) = \frac{w_{\kappa_{\ell j}} g_{\ell \kappa_{\ell j}}\left(\mathbf{z}_{\ell j} | \mathbf{T}, \mathbf{t}^{\mathrm{e}}\right)}{\mu_\ell(\mathbf{z}_{\ell j} | \mathbf{T}, \mathbf{t}^{\mathrm{e}})}. \tag{3.5.26}$$

In this PPP formulation, the goal is find the maximum likelihood estimate of $\mathbf{T}$ by

maximizing (4.3.21), which can also be solved using the EM algorithm.

3.5.2 Optimization

Initialization

In this chapter, three initialization approaches are considered and will be discussed in the Section 3.6.4.

E step

Let $\mathbf{w}^{(n-1)}$, $\mathbf{T}^{(n-1)}$ and $\mathbf{t}^{\mathrm{e},(n-1)}$ denote the estimates from the previous step. In the expectation step, we use them to find $p(\boldsymbol{\kappa}|\mathbf{Z}, \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$ and compute the expected value of $p\left(\boldsymbol{\Psi}^{\mathrm{M}}|\mathbf{T}, \mathbf{t}^{\mathrm{e}}\right)$ conditioned on $p(\boldsymbol{\kappa}|\mathbf{Z}, \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})$, that is, we evaluate

$$\begin{aligned} Q(\mathbf{T}, \mathbf{t}^{\mathrm{e}}|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) &= \mathbb{E}\left[\ln\left(p(\boldsymbol{\Psi}^{\mathrm{M}}\big|\mathbf{T}, \mathbf{t}^{\mathrm{e}})\right)\Big| p(\boldsymbol{\kappa}\,\big|\mathbf{Z},\ \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)})\right] \\ &= Q_w(\mathbf{w}|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) + Q_T(\mathbf{T}, \mathbf{t}^{\mathrm{e}}|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) \end{aligned} \tag{3.5.27}$$

where

$$Q_w(\mathbf{w}|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) = -N_{\mathrm{s}} \sum_{i=0}^{N} w_i + \sum_{\ell=1}^{N_{\mathrm{s}}} \sum_{j=1}^{n_\ell} \sum_{i=0}^{N} \ln(w_i)\, \alpha_{\ell ji}^{(n-1)} \tag{3.5.28}$$

$$Q_T(\mathbf{T}, \mathbf{t}^{\mathrm{e}}|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}) = \sum_{\ell=1}^{N_{\mathrm{s}}} \sum_{j=1}^{n_\ell} \sum_{i=0}^{N} \ln\left(g_{\ell i}(\mathbf{z}_{\ell j})\right) \alpha_{\ell ji}^{(n-1)} \tag{3.5.29}$$

and

$$\alpha_{\ell ji}^{(n-1)} = p_\ell\left(\kappa_{\ell j} = i \,|\, \mathbf{z}_{\ell j}, \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right) = \frac{w_i^{(n-1)} g_{\ell i}^{(n-1)}(\mathbf{z}_{\ell j})}{\sum_{i=0}^{N} w_i^{(n-1)} g_{\ell i}^{(n-1)}(\mathbf{z}_{\ell j})} \tag{3.5.30}$$

The weight $\alpha_{\ell ji}^{(n-1)}$ is the probability that the point $\mathbf{z}_{\ell j}$ is generated by the i-th target given $\mathbf{T}^{(n-1)}$ and $\mathbf{t}^{\mathrm{e},(n-1)}$.

M Step

The M step maximizes $Q\left(\mathbf{T}, \mathbf{t}^{\mathrm{e}} | \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right)$ over all feasible values for $\mathbf{T}$ and $\mathbf{t}^{\mathrm{e}}$. Inspection of (4.4.4) reveals that $Q_w\left(\mathbf{w} \,|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right)$ depends only on the values of $p_{\mathrm{d}}(T_i)$ because $w_i = p_{\mathrm{d}}(T_i)$ for $i = 1, \ldots, N$. Likewise, $Q_T\left(\mathbf{T}, \mathbf{t}^{\mathrm{e}} | \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right)$ depends only on the target locations and emission times through $g_{\ell i}(\mathbf{z}_{\ell j})$. Therefore, maximization of $Q\left(\mathbf{T}, \mathbf{t}^{\mathrm{e}} | \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right)$ is accomplished by maximizing $Q_w\left(\mathbf{w} \,|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right)$ and $Q_T\left(\mathbf{T}, \mathbf{t}^{\mathrm{e}} | \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right)$ separately.

The value of $Q_T\left(\mathbf{T}, \mathbf{t}^{\mathrm{e}} | \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right)$ in (4.4.5) decomposes as

$$Q_T\left(\mathbf{T}, \mathbf{t}^{\mathrm{e}} | \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right) = \sum_{i=0}^{N} Q_{T_i}\left(T_i \,|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right), \tag{3.5.31}$$

where

$$Q_{T_i}\left(T_i \,|\mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)}\right) = \sum_{\ell=1}^{N_{\mathrm{s}}} \sum_{j=1}^{n_\ell} \ln\left(g_{\ell i}\left(\mathbf{z}_{\ell j}\right)\right) \alpha_{\ell ji}^{(n-1)}. \tag{3.5.32}$$

For $i = 0$, $Q_{T_i}\left(T_i \,|\mathbf{T}^{(n-1)}\right)$ is constant with respect to T_i since the density is assumed known for the clutter. When $i \neq 0$, $g_{\ell i}\left(\mathbf{z}_{\ell j}\right)$ depends only on T_i through $\mathbf{h}_\ell(T_i, t_i^{\mathrm{e}})$; thus $T_i^{(n)}$ is determined by maximizing (3.5.32) separately for each value of i.

The values of $w_i^{(n)}$ are determined by maximizing (4.4.4) given the fact that $p_{\mathrm{d}}(T_i) =$

w_i for $k = 1, \ldots, N$ and the assumption that they are scalar quantities. The detection probabilities are also constrained to less than or equal to one. By setting up the Lagrange multipliers, it is easy to see that the KKT conditions are satisfied when

$$w_i{}^{(n)} = \min \left\{ 1, \frac{1}{N_{\mathrm{s}}} \sum_{\ell=1}^{N_{\mathrm{s}}} \sum_{j=1}^{n_\ell} \alpha_{\ell ji}^{(n-1)} \right\}. \tag{3.5.33}$$

The EM algorithm is terminated when the likelihood function (4.3.21) converges, that is

$$\left| \ln p \left(\mathbf{\Psi} \left| \mathbf{T}^{(n)}, \mathbf{t}^{\mathrm{e}(n)} \right. \right) - \ln p \left(\mathbf{\Psi} \left| \mathbf{T}^{(n-1)}, \mathbf{t}^{\mathrm{e},(n-1)} \right. \right) \right| \leq \epsilon \tag{3.5.34}$$

One can use a fixed $\mathbf{w}$ throughout the EM iterations and skip the update process in (3.5.33) under the assumption that both the detection probabilities and the false alarm density are known.

3.5.3 Use of the Information Criterion for Cardinality Selection

The EM algorithm will eventually converge to $\hat{\mathbf{w}}^k$ and $\hat{\mathbf{T}}^k$ given the number of targets $\hat{N}^k$. Let the set $M_p^k = \{\hat{\mathbf{w}}^k, \hat{\mathbf{T}}^k, \hat{N}^k\}$ denote the estimation result based on the k-th cardinality. The BIC selects the set M_p^k that minimizes

$$-2 \ln p \left(\mathbf{\Psi} \left| M_p^k \right. \right) + d \ln N_{\mathbf{z}} \tag{3.5.35}$$

where from (4.3.21)

$$p\left(\mathbf{\Psi}\,\middle|\,M_p^k\right) = \exp\left(-N_{\mathrm{s}}\sum_{i=0}^{\hat{N}^k} w_i^k\right)\prod_{\ell=1}^{N_{\mathrm{s}}}\prod_{j=1}^{n_\ell}\sum_{i=0}^{\hat{N}^k} w_i^k\, g_{\ell i}(\mathbf{z}_{\ell j}\,|\hat{\mathbf{T}}^k) \tag{3.5.36}$$

and d is the total number of parameters to be estimated.

3.6 Simulation Results

3.6.1 Scenario

Assume there are four targets ($N = 4$). The emission times of the acoustic events at the target locations are 0.2 s, 0.25 s, 0.3 s and 0.35 s, respectively. The speed of the acoustic signal is assumed to be 342 m/s. The measurement noise covariance matrix is

$$R_\ell = \begin{bmatrix} 7.6\times 10^{-5} & 0 \\ 0 & 2.5\times 10^{-5} \end{bmatrix} \tag{3.6.1}$$

i.e., the bearing standard deviation amounts to $\sqrt{76}\,\mathrm{mrad} = 0.5^{\mathrm{o}}$ and the time of arrival measurement standard deviation amounts to $5\,\mathrm{ms}$, assumed to be the same for all targets. The time window W is chosen to be $1\,\mathrm{s}$ and the density of the false alarms is set to be $0.32\,\mathrm{s}^{-1}\mathrm{radian}^{-1}$ such that the expected number of false alarms at each sensor is 1. The field view of each sensor is from 0 to π. Figure 3.6.1 shows one example using 10 sensors to localize these 4 targets. In the simulation, the targets and the sensors are arranged in the way such that the angle between two LOS from two neighboring targets to any sensor is 2^{o}, which is 4 times the standard deviation of LOS measurement noise.

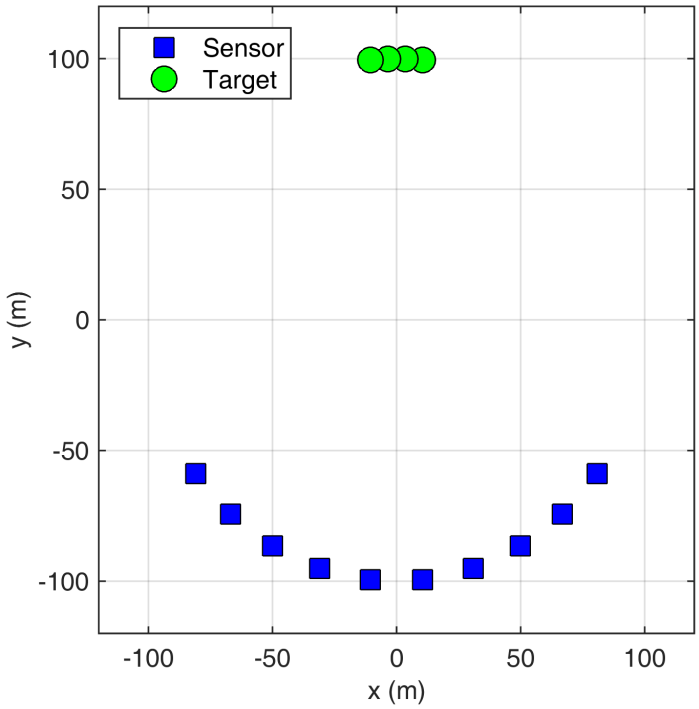


FIGURE 3.6.1: Overhead view of a 10-sensor 4-target scenario

3.6.2 Performance metrics

The performance metrics of interest for N_{MC} Monte Carlo runs include

1. φ_{over}: fraction of N_{MC} runs for which N (the number of targets) has been overestimated;
2. $\bar{M}_{\mathrm{over}}$: average magnitude of estimation error for N from $N_{\mathrm{MC}}\varphi_{\mathrm{over}}$ runs;
3. $T_{\mathrm{over}}^{\mathrm{RMSE}}$: RMSE of the target location estimate from $N_{\mathrm{MC}}\varphi_{\mathrm{over}}$ runs;
4. $\theta_{\mathrm{over}}^{\mathrm{RMSE}}$: RMSE of the bearing estimate from $N_{\mathrm{MC}}\varphi_{\mathrm{over}}$ runs;
5. φ_{under}: fraction of N_{MC} runs for which N has been underestimated;
6. $\bar{M}_{\mathrm{under}}$: average magnitude of estimation error for N from $N_{\mathrm{MC}}\varphi_{\mathrm{under}}$ runs;
7. $T_{\mathrm{under}}^{\mathrm{RMSE}}$: RMSE of the target location estimate from $N_{\mathrm{MC}}\varphi_{\mathrm{under}}$ runs;
8. $\theta_{\mathrm{under}}^{\mathrm{RMSE}}$: RMSE of the bearing estimate from $N_{\mathrm{MC}}\varphi_{\mathrm{under}}$ runs;
9. φ_{exact}: fraction of N_{MC} runs for which N has been correctly estimated;
10. $T_{\mathrm{exact}}^{\mathrm{RMSE}}$: RMSE of the target location estimate from $N_{\mathrm{MC}}\varphi_{\mathrm{exact}}$ runs;
11. $\theta_{\mathrm{exact}}^{\mathrm{RMSE}}$: RMSE of the bearing estimate from $N_{\mathrm{MC}}\varphi_{\mathrm{exact}}$ runs;
12. $T_{\mathrm{all}}^{\mathrm{RMSE}}$: RMSE of the target location estimate from all N_{MC} runs;
13. $\theta_{\mathrm{all}}^{\mathrm{RMSE}}$: RMSE of the bearing estimate from all N_{MC} runs.
14. t: average processing time in a single run.

For all the simulations in this chapter, $N_{\mathrm{MC}} = 1000$ unless otherwise specified. In the overestimation cases, RMSE is calculated by mapping the best (yielding the minimum RMSE) subset of estimated targets to true targets. In the underestimation cases, RMSE is calculated by mapping the estimated targets to the best subset of true targets. The bearing estimate is examined here, because in some applications (for instance, shooter localization), bearing accuracy is more critical than location accuracy.

3.6.3 Assignment algorithms

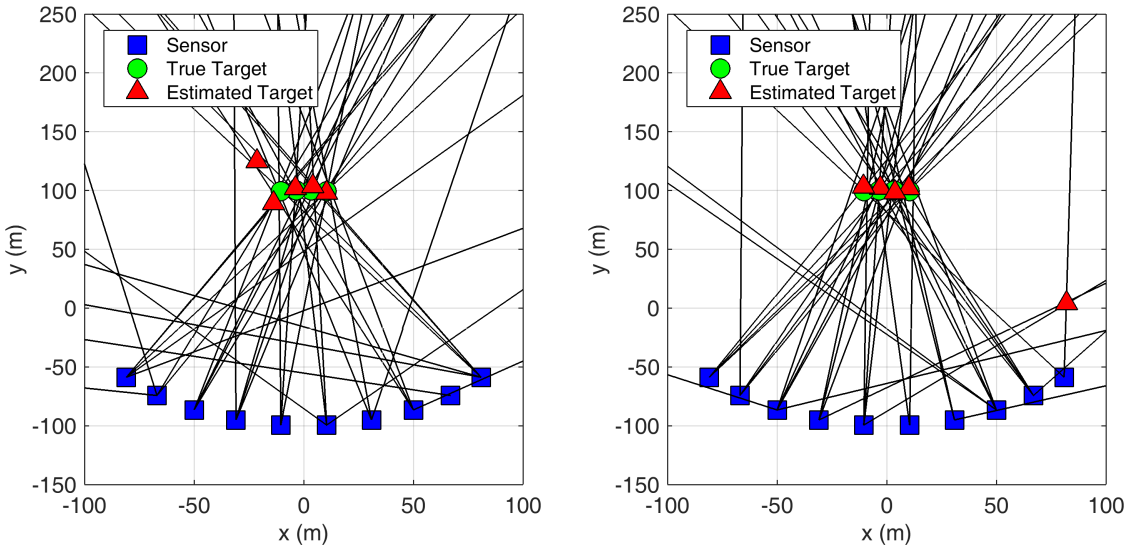
The multidimensional assignment problem (3.3.9), which is solved by the two assignment algorithms — the S-D assignment algorithm and the sequential m-best 2-D algorithm, assumes a Bernoulli measurement model that the number of measurements from a real target received by a sensor is a Bernoulli random variable whose parameter equal to the probability of detection p_{d}. For the evaluation of these two assignment algorithms in this subsection, the target measurements are generated according to this Bernoulli measurement model, specifically, one measurement from each target is generated for each sensor with a probability p_{d} or nothing with a probability $1-p_{\mathrm{d}}$. The false alarms are generated for each sensor according to the Poisson model (4.3.8) and (4.3.9).

Note that the values of the probability of detection, p_{d}, and the expected number of false alarms, N_{fa}, are required to generate the target measurements. The assignment algorithms do not need to know the value of N_{fa} but need to know the value of p_{d}. However, the assignment algorithms are shown to be robust to incorrect p_{d} (see Table 3.6.3).

Table 3.6.1 shows the effects of the algorithm parameter m, the number of best assignments to be kept at each step, and p_{d}, probability of detection, on the performance of the sequential m-best 2-D assignment algorithm in a scenario using 10 sensors to locate 4

TABLE 3.6.1: Sequential m-best 2-D assignment performance using different m for known p_{d}

p_{d}	0.7			0.8			0.9		
m	1	2	4	1	2	4	1	2	4
φ_{over}	1.6%	1.2%	0.9%	0.9%	0.8%	0.7%	0.6%	0.6%	0.4%
$\bar{M}_{\mathrm{over}}$	1	1	1	1	1	1	1	1	1
$T_{\mathrm{over}}^{\mathrm{RMSE}}$ (m)	2.86	5.13	3.01	1.97	1.63	1.72	2.19	2.19	1.98
$\theta_{\mathrm{over}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.289	0.460	0.298	0.255	0.251	0.262	0.272	0.272	0.274
φ_{under}	3.2%	2.3%	2.1%	0.2%	0	0	0	0	0
$\bar{M}_{\mathrm{under}}$	1	1	1	1.5	N.A.	N.A.	N.A.	N.A.	N.A.
$T_{\mathrm{under}}^{\mathrm{RMSE}}$ (m)	2.58	2.69	2.69	4.50	N.A.	N.A.	N.A.	N.A.	N.A.
$\theta_{\mathrm{under}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.296	0.299	0.298	0.411	N.A.	N.A.	N.A.	N.A.	N.A.
φ_{exact}	95.2%	96.5%	97.0%	98.9%	99.2%	99.3%	99.4%	99.4%	99.6%
$T_{\mathrm{exact}}^{\mathrm{RMSE}}$ (m)	2.92	2.81	2.76	3.23	2.59	2.59	1.99	1.96	1.96
$\theta_{\mathrm{exact}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.329	0.320	0.313	0.329	0.282	0.282	0.239	0.236	0.236
$T_{\mathrm{all}}^{\mathrm{RMSE}}$ (m)	2.91	2.85	2.76	3.23	2.58	2.58	1.99	1.96	1.96
$\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.328	0.322	0.313	0.328	0.282	0.282	0.239	0.236	0.236
t (s)	0.068	0.142	0.310	0.077	0.166	0.353	0.084	0.186	0.394



(A) Target Splitting (B) False Association

FIGURE 3.6.2: Two cardinality overestimation situations.

TABLE 3.6.2: Sequential m-best 2-D assignment performance using different N_{s} for known p_{d} with $m = 4$

p_{d}	0.7			0.8			0.9		
N_{s}	6	8	10	6	8	10	6	8	10
φ_{over}	0.1%	0.4%	0.9%	0.2%	0.6%	0.7%	0.4%	0.5%	0.4%
$\bar{M}_{\mathrm{over}}$	1	1	1	1	1	1	1	1	1
$T_{\mathrm{over}}^{\mathrm{RMSE}}$ (m)	4.37	4.50	3.01	3.90	2.87	1.72	4.21	2.98	1.98
$\theta_{\mathrm{over}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.453	0.386	0.298	0.361	0.294	0.262	0.302	0.289	0.274
φ_{under}	28.2%	7.8%	2.1%	7.8%	1%	0	0.4%	0	0
$\bar{M}_{\mathrm{under}}$	1.12	1.01	1	1.04	1	N.A.	1	N.A.	N.A.
$T_{\mathrm{under}}^{\mathrm{RMSE}}$ (m)	5.95	3.89	2.69	4.93	4.11	N.A.	4.64	N.A.	N.A.
$\theta_{\mathrm{under}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.402	0.344	0.298	0.347	0.328	N.A.	0.330	N.A.	N.A.
φ_{exact}	71.7%	91.8%	97.0%	92%	98.4%	99.3%	99.2%	99.5%	99.6%
$T_{\mathrm{exact}}^{\mathrm{RMSE}}$ (m)	7.26	4.37	2.76	5.66	3.41	2.59	4.42	2.76	1.96
$\theta_{\mathrm{exact}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	1.03	0.442	0.313	0.369	0.310	0.282	0.314	0.266	0.236
$T_{\mathrm{all}}^{\mathrm{RMSE}}$ (m)	6.99	4.35	2.76	5.62	3.42	2.58	4.42	2.76	1.96
$\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.931	0.437	0.313	0.368	0.310	0.282	0.314	0.266	0.236
t (s)	0.132	0.212	0.310	0.158	0.274	0.353	0.183	0.277	0.394

targets. The true probability of detection is assumed to be known. Once the assignment algorithm is finished, any association with less than three ($\tau = 3$) real measurements is discarded. For each p_{d}, keeping more (larger m) top assignments at each 2-D step makes it more likely to find the best assignment, therefore, a larger m gives better estimates for the number of targets, the locations of the targets and the directions to the targets (for counterfire). In fact, if the association between the measurements and the targets is known, the values of $\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ would be 2.50, 2.15 and 1.91 m in the three scenarios with p_{d} varying from 0.7 to 0.9. The sequential algorithm with m=4 yields very good association accuracy since the values of $\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ are very close to these lower bounds. The performance gain is at the expense of a higher computational cost, which, however, is acceptable for real-time applications. For a fixed value of m, the algorithm performs better when the probability of detection increases. Lower p_{d} makes it more likely for the final association to have

fewer real measurements, which could fail in the real threshold test, therefore, there are more cases when the number of targets is underestimated at $p_{\mathrm{d}} = 0.7$. There are two situations when the number of targets is overestimated. The first one is "target splitting", that is, one real target is perceived as two (or more) targets which are close to each other as shown in Figure 3.6.2(a) where the target at location (-10, 99) are split into the targets at locations (-14, 89) and (-21, 125). The second one is when three (or more) false measurements are perceived to be from a real target as shown in Figure 3.6.2(b) at location (82, 4). Both situations are more likely to occur at a lower p_{d}, that is why there are slightly fewer overestimation cases as p_{d} increases, as shown in the Table 3.6.1. When p_{d} is higher, there could be more real measurements available, which, if correctly associated, can lead to more accurate location and direction estimates. This is also at the expense of a slightly longer processing time.

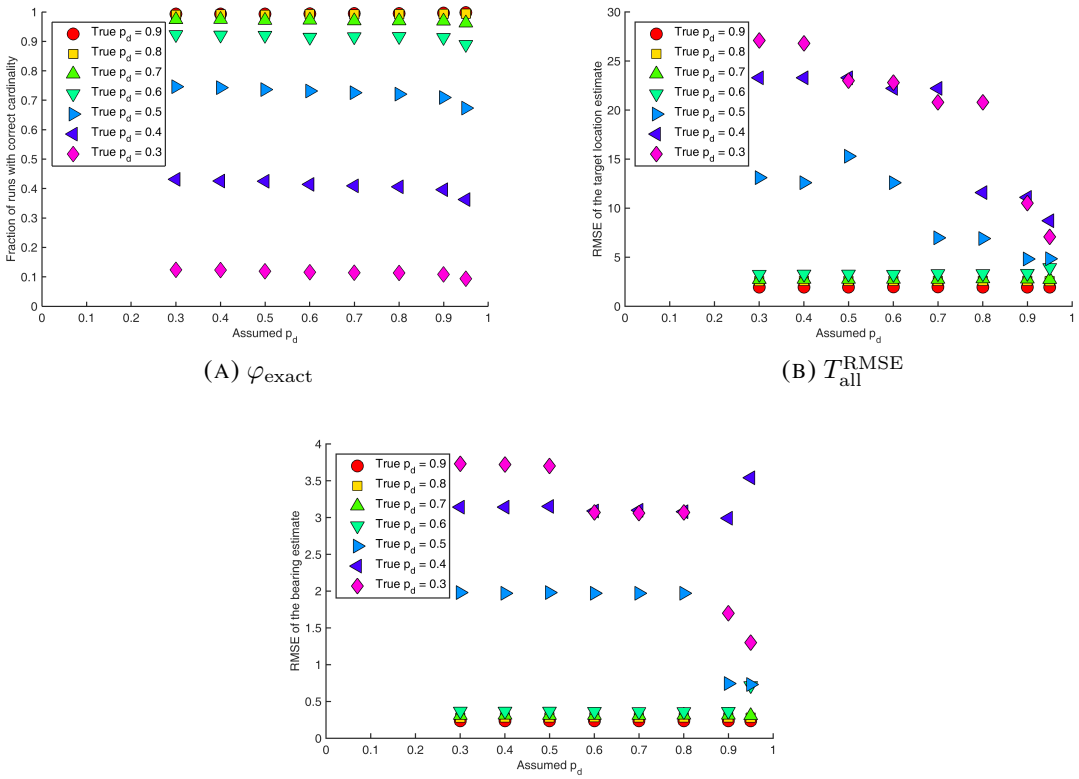
Table 3.6.2 shows how the performance of the sequential m-best 2-D assignment algorithm with $m = 4$ varies with the number of sensors for three levels of known p_{d} in the scenario with 4 real targets. For the threshold test, any associations with less than 3 real measurement are discarded. For each p_{d}, it is observed that using a smaller number of sensors leads to more cardinality underestimation cases and fewer overestimation cases. This is because the chance that only one sensor detects a real target is higher when a smaller number of sensors are used. On the other hand, a larger number of sensors in the presence of false alarms makes it more likely to associate false measurements into a ghost like the situation in Figure 3.6.2(b). However, with a larger number of sensors, the decrease in the occurrence of overestimation cases is more significant than the increase in the underestimation cases. In addition, deploying more sensors could give rise to more measurements for a target, which in return generate more accurate location and direction estimates. In general, it is more beneficial to use a larger number of sensors.

TABLE 3.6.3: Sequential m-best 2-D assignment performance using different assumed values for unknown p_{d}

true p_{d}	0.7			0.8			0.9		
assumed p_{d}	0.6	0.7	0.8	0.7	0.8	0.9	0.8	0.9	0.95
φ_{over}	1.0%	0.9%	0.8%	0.8%	0.7%	0.5%	0.5%	0.4%	0.2%
$\bar{M}_{\mathrm{over}}$	1	1	1	1	1	1	1	1	1
$T_{\mathrm{over}}^{\mathrm{RMSE}}$ (m)	3.12	3.01	2.14	2.40	1.72	1.71	1.97	1.98	2.12
$\theta_{\mathrm{over}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.314	0.298	0.252	0.315	0.262	0.249	0.263	0.274	0.270
φ_{under}	1.7%	2.1%	2.2%	0	0	0.1%	0	0	0
$\bar{M}_{\mathrm{under}}$	1	1	1	N.A.	N.A.	1	N.A.	N.A.	N.A.
$T_{\mathrm{under}}^{\mathrm{RMSE}}$ (m)	2.80	2.69	2.79	N.A.	N.A.	2.91	N.A.	N.A.	N.A.
$\theta_{\mathrm{under}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.311	0.298	0.317	N.A.	N.A.	0.325	N.A.	N.A.	N.A.
φ_{exact}	97.3%	97.0%	97.0%	99.2%	99.3%	99.4%	99.5%	99.6%	99.8%
$T_{\mathrm{exact}}^{\mathrm{RMSE}}$ (m)	2.76	2.76	2.80	2.60	2.59	2.60	1.96	1.96	1.96
$\theta_{\mathrm{exact}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.313	0.313	0.318	0.284	0.282	0.282	0.236	0.236	0.236
$T_{\mathrm{all}}^{\mathrm{RMSE}}$ (m)	2.76	2.76	2.80	2.60	2.58	2.59	1.96	1.96	1.96
$\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.313	0.313	0.317	0.284	0.282	0.282	0.236	0.236	0.236
t (s)	0.307	0.310	0.316	0.353	0.353	0.350	0.394	0.394	0.392

The calculation of the cost (3.3.3) requires the knowledge of p_{d}. In practical scenarios, the actual value of p_{d} may not be available, in which case one has to use an estimated value of p_{d}. Table 3.6.3 shows that the performance of the sequential m-best 2-D assignment algorithm is almost insensitive to the mismatch between the assumed value and the true value of p_{d} when they are close (up to 0.1 difference) in the scenario where 10 sensors are used to locate 4 targets.

Figure 3.6.3 shows the performance of the sequential m-best 2-D assignment algorithm for a wider range of true p_{d}. The quality of cardinality, location and bearing estimates is almost independent of the assumed p_{d} value when the true p_{d} is from 0.6 to 0.9. When the true p_{d} is as low as 0.5, the quality of those estimates will vary with the assumed p_{d} value. A good location or bearing estimate requires at least 3 real measurements to be associated correctly, however, the probability that at least 3 out of 10 sensors have detected the same



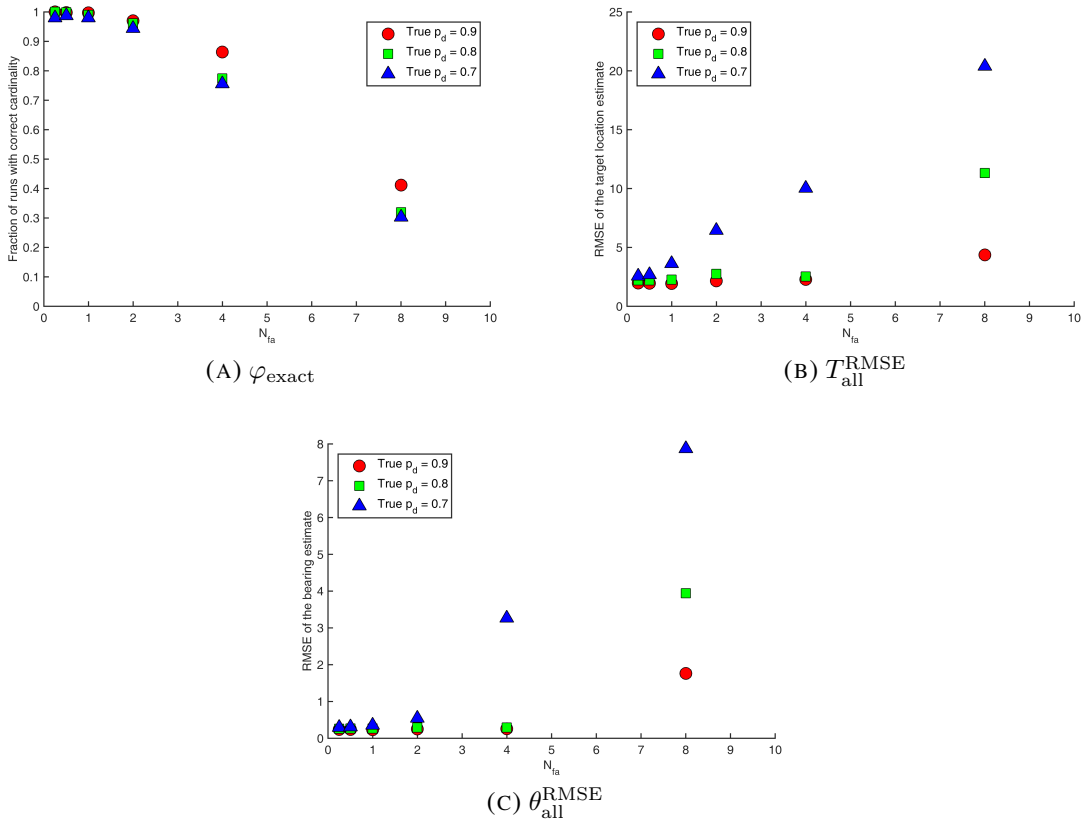
(A) φ_{exact} (B) $T_{\text{all}}^{\text{RMSE}}$

(C) $\theta_{\text{all}}^{\text{RMSE}}$

FIGURE 3.6.3: The performance (in terms of φ_{exact}, $T_{\text{all}}^{\text{RMSE}}$ and $\theta_{\text{all}}^{\text{RMSE}}$) of the sequential m-best 2-D assignment algorithm using different assumed values of unknown p_{d} for true p_{d} values ranging from 0.3 to 0.9.

target is only around 0.80 at true $p_{\text{d}} = 0.5$.. In other words, at true $p_{\text{d}} = 0.5$. in about 200 simulation runs either there is at least one target missing or there is at least a false association, which could cause very different performances at different assumed p_{d} values. It gets worse at lower p_{d} values such as 0.4 and 0.3. Therefore, given a fixed number of sensors, there is a lower bound on the true p_{d}, below which it is very difficult to achieve good location and bearing estimates.

Figure 3.6.4 shows the dependence of the performance of the sequential m-best 2-D assignment algorithm on the expected number of false alarms per sensor when 10 sensors are

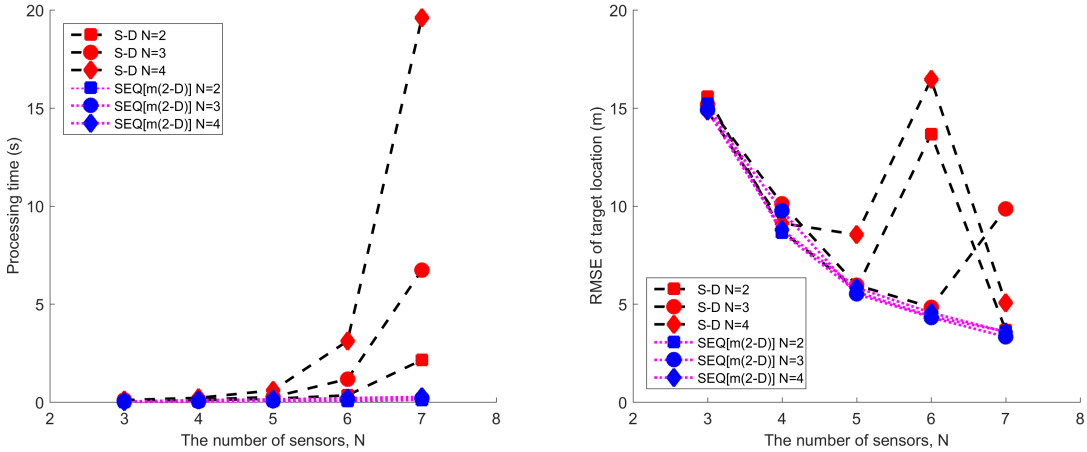


(A) φ_{exact} (B) $T_{\text{all}}^{\text{RMSE}}$

(C) $\theta_{\text{all}}^{\text{RMSE}}$

FIGURE 3.6.4: The performance (in terms of φ_{exact}, $T_{\text{all}}^{\text{RMSE}}$ and $\theta_{\text{all}}^{\text{RMSE}}$) of the sequential m-best 2-D assignment algorithm in scenarios with different known expected number of false alarms (0.25, 0.5, 1, 2, 4 and 8) for known p_{d} values at 0.7, 0.8 and 0.9.

used to localize 4 targets with known p_{d} values at 0.7, 0.8 and 0.9. In the final assignment, the least number of real measurements required to be associated with a real target is 3. As expected, the performance is getting worse when the false alarm rate is higher. When the true p_{d} is 0.8 or 0.9, the localization results are reliable even for N_{fa} = 4 (the total expected number of false alarms is larger than the total expected number of real measurements). When the expected number of false alarms per sensor is very large (N_{fa} = 8), there are more false targets in the final assignment, leading to a worse performance. Setting a higher number for the required minimum number of real measurements reduces the number of



(A) Processing time (B) RMSE of target location

FIGURE 3.6.5: Performance comparison between the S-D assignment algorithm and the sequential m-best 2-D (SEQ[m(2-D)]) assignment algorithm for $p_{\rm d} = 0.9$ and $N_{\rm fa} = 1$.

false targets but will also miss real targets in a scenario, especially for a low $p_{\rm d}$ value.

Figure 3.6.5(a) shows the processing time (averaged over 100 Monte Carlo runs) of the S-D assignment algorithm and the sequential m-best ($m = 4$) 2-D assignment algorithm in scenarios with the probability of detection $p_{\rm d} = 0.9$ and the expected number of false alarms (per sensor) $N_{\rm fa} = 1$. When the number of targets, N, is 4, the processing time using the S-D assignment algorithm with 7 sensors is around 20 seconds, which is too long as far as real-time applications are concerned. This is due to the fact that the processing time of the S-D assignment algorithm scales exponentially with the number of sensors while the sequential processing time scales quadratically with the number of sensors in the worst case. In terms of the localization performance, the S-D assignment algorithm is also shown to be inferior to the sequential m-best assignment algorithm.

We must note that both algorithms are very different suboptimal solutions to the problem (3.3.9). The sequential m-best assignment algorithm makes efficient use of the modified auction algorithm in a sequential manner. As a greedy algorithm, it solves a locally

optimal solution based on two lists of measurements followed by considering one more list of measurements at a time until a final solution is obtained for all lists of measurements. It uses a suitably large value of the parameter m with the hope that the optimal solution is kept in the search space at all times and the final solution is close to optimal.

In contrast, the S-D assignment algorithm is a much more sophisticated, iterative technique as shown in Figure 3.3.1. However, it was shown in [18] that the S-D assignment algorithm could be underperforming in challenging localization scenarios where the association graph is strongly connected and the number of candidate associations is huge. In a scenario with $p_{\mathrm{d}} = 0.9$ where 5 targets and 7 sensors are placed such that the worst angular intertarget separations as seen by the 7 sensors are 2, 3.5, 5, 6.5, 5, 3.5, and 2 standard deviations of the bearing measurement noise, the association accuracy of the S-D assignment algorithm is shown to be 74%. In our simulation scenario, the worst angular intertarget separation is 4 standard deviations for every sensor, which poses a similar challenging situation in terms of the density of the association graph (or the number of candidate associations). Therefore, the somewhat inferior localization performance of the S-D assignment algorithm at $p_{\mathrm{d}} = 0.9$ is not unexpected.

As suggested in [18], the algorithmic parameters are selected such that the algorithm is terminated if the relative approximate duality gap is less than 5% or the number of iterations exceeds 100. Although the chance is very small, it is still possible that the algorithm already terminates at the 100-th iteration and the relative approximate duality gap has not been reduced to 5%. Even if the relative approximate duality gap is less than 5% long before the 100-th iteration, it is still possible that the algorithm stops at a local minimum of the objective function (3.3.9) and, although it has a similar association cost to the optimal solution, it yields different target locations. In addition, the more false alarms the sensors detect, the more likely that the algorithm terminates at a local minimum.

TABLE 3.6.4: Comparison between S-D assignment and Sequential m-best (SmB) 2-D assignment performance for $p_{\rm d} = 0.9$ (assumed unknown)

	S-D			SmB(m=4)		
assumed $p_{\rm d}$	0.8	0.9	0.99	0.8	0.9	0.99
$\varphi_{\rm over}$	1.8%	2.0%	3.3%	0.4%	0.4%	0.2%
$\bar{M}_{\rm over}$	1	1	1	1	1	1
$T^{\rm RMSE}_{\rm over}$ (m)	14.88	14.24	6.93	3.68	3.55	3.64
$\theta^{\rm RMSE}_{\rm over}$ ($^{\rm o}$)	0.714	0.640	0.459	0.309	0.306	0.241
$\varphi_{\rm under}$	9.7%	1.9%	0.8%	0.4%	0.3%	0.3%
$\bar{M}_{\rm under}$	1.06	1	1	1	1	1
$T^{\rm RMSE}_{\rm under}$ (m)	30.67	18.14	8.72	3.08	2.91	2.91
$\theta^{\rm RMSE}_{\rm under}$ ($^{\rm o}$)	1.17	0.792	0.518	0.272	0.286	0.286
$\varphi_{\rm exact}$	88.5%	96.1%	95.9%	99.2%	99.3%	99.5%
$T^{\rm RMSE}_{\rm exact}$ (m)	10.47	9.44	4.87	4.91	4.42	4.43
$\theta^{\rm RMSE}_{\rm exact}$ ($^{\rm o}$)	1.14	0.832	0.348	0.325	0.318	0.319
$T^{\rm RMSE}_{\rm all}$ (m)	13.14	9.74	4.98	4.90	4.42	4.42
$\theta^{\rm RMSE}_{\rm all}$ ($^{\rm o}$)	1.14	0.828	0.354	0.325	0.318	0.319
t (s)	0.549	0.568	0.541	0.166	0.162	0.160

As the number of sensors increases, the association graph becomes more dense and the number of candidate associations explodes combinatorially and it becomes more difficult for the S-D assignment algorithm to solve the association problem. Therefore, it is not practical to apply the S-D assignment algorithm directly[9] when the number of sensors is large. We suggest the use of the sequential m-best assignment algorithm in applications with a large number of sensors.

For additional comparison, the performance of both the S-D assignment and the sequential m-best assignment algorithms for different $p_{\rm d}$ in a scenario with 4 targets and 6 sensors is listed in Tables 3.6.4–3.6.6.

[9]One possible practice is to use the S-D assignment algorithm on a subset of sensors followed by sequential processing as in [61].

TABLE 3.6.5: Comparison between S-D assignment and Sequential m-best (SmB) 2-D assignment performance for $p_{\rm d} = 0.8$ (assumed unknown)

	S-D			SmB(m=4)		
assumed $p_{\rm d}$	0.7	0.8	0.9	0.7	0.8	0.9
$\varphi_{\rm over}$	0.7%	0.6%	0.9%	0.4%	0.4%	0.4%
$\bar{M}_{\rm over}$	1	1	1	1	1	1
$T_{\rm over}^{\rm RMSE}$ (m)	33.13	4.21	15.96	6.74	6.74	6.74
$\theta_{\rm over}^{\rm RMSE}$ ($^{\rm o}$)	0.882	0.302	0.829	0.421	0.421	0.421
$\varphi_{\rm under}$	31.1%	25.4%	11.8%	7.6%	7.5%	7.5%
$\bar{M}_{\rm under}$	1.24	1.23	1.03	1.03	1.03	1.03
$T_{\rm under}^{\rm RMSE}$ (m)	31.40	30.12	20.52	5.46	5.49	5.88
$\theta_{\rm under}^{\rm RMSE}$ ($^{\rm o}$)	1.53	1.58	0.989	0.366	0.367	0.380
$\varphi_{\rm exact}$	68.2%	74.0%	87.3%	92%	92.1%	92.1%
$T_{\rm exact}^{\rm RMSE}$ (m)	12.69	12.86	18.51	5.11	5.11	5.11
$\theta_{\rm exact}^{\rm RMSE}$ ($^{\rm o}$)	1.10	1.06	1.10	0.359	0.359	0.359
$T_{\rm all}^{\rm RMSE}$ (m)	19.08	17.49	18.67	5.14	5.14	5.16
$\theta_{\rm all}^{\rm RMSE}$ ($^{\rm o}$)	1.22	1.18	1.09	0.360	0.360	0.361
t (s)	0.415	0.420	0.413	0.142	0.141	0.150

3.6.4 EM-based algorithms

In Sections 3.4 and 3.5, we have considered the Poisson measurement model for each target, which leads to either the UGM or the PPP formulation. Both are solved using the EM algorithm. For the evaluation of these two EM-based algorithms in this subsection, the target measurements are generated according to this Poisson measurement model, specifically, if a sensor has a certain $p_{\rm d}$, the number of measurements originated from a target is a Poisson random variable with parameter $p_{\rm d}$. The clutter follows a Poisson model with parameter $N_{\rm fa}$.

Note that the values of the probability of detection, $p_{\rm d}$, and the expected number of false alarms, $N_{\rm fa}$, are required to generate the target measurements. However, these EM-based algorithms do not need to know the values of $N_{\rm fa}$ and $p_{\rm d}$. They adapt to these values by

TABLE 3.6.6: Comparison between S-D assignment and Sequential m-best (SmB) 2-D assignment performance for $p_{\rm d} = 0.7$ (assumed unknown)

	S-D			SmB(m=4)		
assumed $p_{\rm d}$	0.6	0.7	0.8	0.6	0.7	0.8
$\varphi_{\rm over}$	0.5%	0.3%	0.2%	0	0	0
$\bar{M}_{\rm over}$	1	1	1	N.A.	N.A.	N.A.
$T_{\rm over}^{\rm RMSE}$ (m)	5.93	6.30	6.80	N.A.	N.A.	N.A.
$\theta_{\rm over}^{\rm RMSE}$ ($^{\rm o}$)	0.360	0.382	0.395	N.A.	N.A.	N.A.
$\varphi_{\rm under}$	62.6%	57.4%	52.2%	27.5%	27.5%	27.5%
$\bar{M}_{\rm under}$	1.45	1.46	1.39	1.14	1.14	1.14
$T_{\rm under}^{\rm RMSE}$ (m)	35.39	35.24	34.48	6.93	6.97	6.59
$\theta_{\rm under}^{\rm RMSE}$ ($^{\rm o}$)	2.14	2.19	2.20	0.398	0.400	0.402
$\varphi_{\rm exact}$	36.9%	42.3%	47.6%	72.5%	72.5%	72.5%
$T_{\rm exact}^{\rm RMSE}$ (m)	14.78	14.90	14.92	6.96	6.96	6.96
$\theta_{\rm exact}^{\rm RMSE}$ ($^{\rm o}$)	0.941	0.911	0.910	0.860	0.860	0.860
$T_{\rm all}^{\rm RMSE}$ (m)	27.41	26.29	24.99	6.95	6.96	6.88
$\theta_{\rm all}^{\rm RMSE}$ ($^{\rm o}$)	1.67	1.63	1.58	0.784	0.785	0.785
t (s)	0.318	0.318	0.322	0.131	0.127	0.131

"learning them".

The EM-based algorithm starts with an initialization, which determines whether the objective function can converge to the global maximum or a local maximum. Three initialization approaches are considered in this chapter.

The first approach is to initialize the target locations and the emission times using their true values. This initialization approach works well as shown later, however, it critically depends on the truth, which is not available in the real world. Nevertheless, it provides a benchmark on how well the EM-based algorithms can perform. Since the number of the targets N is unknown, one needs to evaluate a range of values for N and the algorithm selects the best N based on BIC. Such possible values for N can be selected based on the number of measurements obtained at each sensor, five values (2 to 6) are chosen for the 4-target scenario considered here. When the evaluated number of targets is less than the true

value, a subset of the true targets are used for initialization. When the evaluated number of targets is more than the true value, auxiliary targets in additional to the true ones are used for initialization.

The second approach is based on the k-means clustering algorithm. Any two bearing (or LOS) measurements from two different sensors can lead to a potential target. In the absence of measurement noise, the LOS measurements coming from the same target intersect at a single point. In the presence of measurement noise, the LOS measurements originating from the same target should intersect with each other in a close neighborhood. Therefore, the points of intersection from the LOS measurements of any two sensors are clustered and the centroids of each cluster are used to initialize the target locations. The emission times are initialized in the same way. As in the first approach, five values are evaluated for N.

The third approach is based on the sequential m-best 2-D assignment algorithm. The associations with more than 2 real measurements correspond to potential targets. Let $N_{\max}$ denote the number of such associations. These associations are ranked in terms of the association cost. A range of values from 1 to $N_{\max}$ will be evaluated for N, and the top N associations will be used to initialized the EM-based algorithm.

Tables 3.6.7–3.6.9 present the performance of the EM-based algorithm with both UGM and PPP formulations (UGM/EM and PPP/EM) using different initialization approaches at three levels[10] of p_{d} with a known false alarm rate ($N_{\mathrm{fa}} = 1$) in a scenario where 10 sensors are used to locate 4 targets.

Initialization at the truth enables the EM-based algorithm to estimate the number of targets, target locations and directions accurately and the estimation becomes more accurate as p_{d} increases. In this case, the global maximum is attained.

[10]The probability of detection is set to be the same for each target in the simulation studies only for simplicity, the EM-based algorithms can deal with the case that the probabilities of detection for different targets are distinct.

TABLE 3.6.7: UGM/EM and PPP/EM performance using different initialization ("I:") approaches for unknown $p_{\mathrm{d}} = 0.7$

	I: truth		I: clustering		I: SmB(m=1)		I: SmB(m=2)	
	UGM/EM	PPP/EM	UGM/EM	PPP/EM	UGM/EM	PPP/EM	UGM/EM	PPP/EM
φ_{over}	0.1%	0	49.7%	50.5%	19.4%	19.3%	19.8%	19.7%
$\bar{M}_{\mathrm{over}}$	1	N.A.	1.57	1.51	1.14	1.14	1.15	1.15
$T_{\mathrm{over}}^{\mathrm{RMSE}}$ (m)	4.96	N.A.	36.69	28.51	3.46	3.47	3.34	3.35
$\theta_{\mathrm{over}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.535	N.A.	2.23	2.37	0.493	0.494	0.449	0.451
φ_{under}	0.6%	0.7%	29.7%	30.5%	23.4%	23.4%	23%	22.9%
$\bar{M}_{\mathrm{under}}$	1	1	1.52	1.48	1.10	1.10	1.09	1.08
$T_{\mathrm{under}}^{\mathrm{RMSE}}$ (m)	3.06	2.57	34.21	31.8	8.28	8.30	2.88	3.04
$\theta_{\mathrm{under}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.403	0.352	3.66	3.30	0.442	0.449	0.347	0.357
φ_{exact}	99.3%	99.3%	20.6%	19%	57.2%	57.3%	57.2%	57.4%
$T_{\mathrm{exact}}^{\mathrm{RMSE}}$ (m)	3.05	3.01	70.95	52.96	5.56	5.54	5.53	5.55
$\theta_{\mathrm{exact}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.351	0.347	5.38	5.27	0.615	0.615	0.615	0.620
$T_{\mathrm{all}}^{\mathrm{RMSE}}$ (m)	3.05	3.01	46.37	35.69	5.81	5.81	4.74	4.78
$\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.351	0.347	3.49	3.38	0.563	0.564	0.543	0.548
t (s)	2.52	2.40	4.72	4.32	1.23	1.21	1.31	1.31

The clustering based initialization is very prone to ghosting and therefore results in very large errors in terms of the number of targets, target locations and directions. It also takes a longer processing time with such a poor initialization. In this case, the algorithm terminates at a local maximum.

The assignment based initialization overcomes the ghosting problem. With m=2 in the sequential m-best 2-D assignment, the target direction errors are less than the standard deviation of the bearing measurement noise and the target location errors are close to those obtained using initialization at the truth. At a higher p_{d}, the number of overestimation cases increases. This is due to double counting of the same target by the assignment algorithm when two acoustic events occur at the same location. The assignment algorithm does not differentiate the acoustic events that occurred at the same location. Although it indicates that there are more targets than the truth, all real targets have actually been identified.

TABLE 3.6.8: UGM/EM and PPP/EM performance using different initialization ("I:") approaches for unknown $p_{\mathrm{d}} = 0.8$

	I: truth		I: clustering		I: SmB(m=1)		I: SmB(m=2)	
	UGM/EM	PPP/EM	UGM/EM	PPP/EM	UGM/EM	PPP/EM	UGM/EM	PPP/EM
φ_{over}	0	0	51.5%	54.8%	26.9%	27.1%	27.5%	27.8%
$\bar{M}_{\mathrm{over}}$	N.A.	N.A.	1.53	1.52	1.21	1.21	1.23	1.23
$T_{\mathrm{over}}^{\mathrm{RMSE}}$ (m)	N.A.	N.A.	30.97	40.07	2.84	2.87	2.85	2.88
$\theta_{\mathrm{over}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	N.A.	N.A.	2.14	2.17	0.391	0.405	0.394	0.409
φ_{under}	0.1%	0.1%	29.6%	27.1%	14.2%	14.0%	14.1%	13.8%
$\bar{M}_{\mathrm{under}}$	1	1	1.50	1.46	1.09	1.09	1.10	1.09
$T_{\mathrm{under}}^{\mathrm{RMSE}}$ (m)	0.69	0.74	33.60	29.94	2.58	2.57	2.57	2.57
$\theta_{\mathrm{under}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.0886	0.0903	3.28	3.09	0.317	0.316	0.317	0.316
φ_{exact}	99.9%	99.9%	18.9%	18.1%	58.9%	58.9%	58.4%	58.4%
$T_{\mathrm{exact}}^{\mathrm{RMSE}}$ (m)	2.55	2.52	46.88	47.86	3.06	3.06	2.97	2.98
$\theta_{\mathrm{exact}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.306	0.303	4.60	4.64	0.431	0.430	0.429	0.428
$T_{\mathrm{all}}^{\mathrm{RMSE}}$ (m)	2.55	2.52	35.44	40.07	2.95	2.96	2.89	2.91
$\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.306	0.303	3.06	3.00	0.409	0.412	0.408	0.412
t (s)	2.17	2.16	4.51	4.62	1.69	1.71	1.85	1.84

Figure 3.6.6 compares the performance of the EM-based algorithm with both UGM and PPP formulations (UGM/EM and PPP/EM) with initialization at the truth for different known expected numbers of false alarms (or false alarm rate) in a scenario where 10 sensors are used to locate 4 targets. The quality of the cardinality, location and bearing estimates using both formulations is almost identical for the same p_{d} value, which demonstrates the effectiveness of the UGM formulation to incorporate the false alarm rate when it is known.

3.6.5 Assignment algorithms and EM-based algorithms

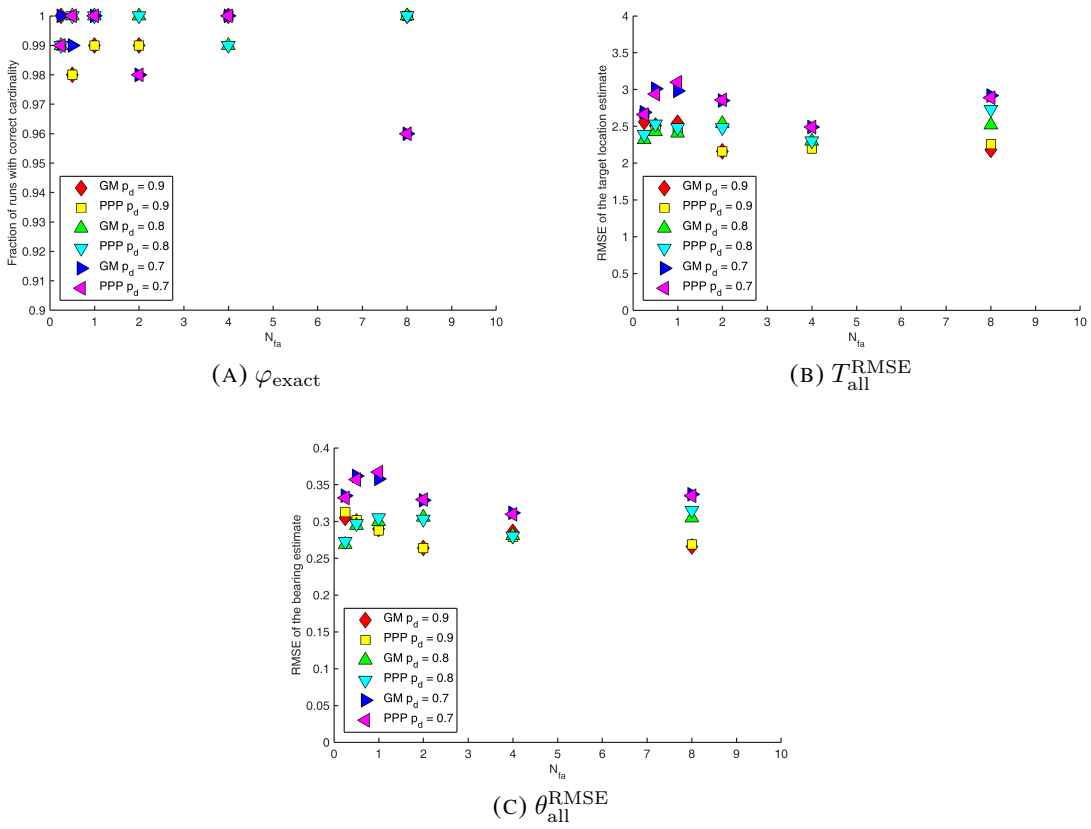
In the previous two subsections, the two types of algorithms — the assignment algorithms and the EM-based algorithms — were evaluated separately according to their assumed target-originated measurement models (Bernoulli and Poisson, respectively). Since

TABLE 3.6.9: UGM/EM and PPP/EM performance using different initialization ("I:") approaches for unknown $p_{\mathrm{d}} = 0.9$

	I: truth		I: clustering		I: SmB(m=1)		I: SmB(m=2)	
	UGM/EM	PPP/EM	UGM/EM	PPP/EM	UGM/EM	PPP/EM	UGM/EM	PPP/EM
φ_{over}	0	0	52.1%	53.8%	33.6%	33.6%	34.7%	34.7%
$\bar{M}_{\mathrm{over}}$	N.A.	N.A.	1.60	1.59	1.26	1.27	1.26	1.27
$T_{\mathrm{over}}^{\mathrm{RMSE}}$ (m)	N.A.	N.A.	37.92	29.36	2.44	2.46	2.49	2.50
$\theta_{\mathrm{over}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	N.A.	N.A.	1.99	1.89	0.321	0.321	0.331	0.331
φ_{under}	0.3%	0.3%	28%	27.2%	9.9%	9.9%	8.9%	8.9%
$\bar{M}_{\mathrm{under}}$	1	1	1.48	1.48	1.03	1.03	1.03	1.03
$T_{\mathrm{under}}^{\mathrm{RMSE}}$ (m)	1.41	1.40	27.82	25.97	2.55	2.58	2.29	2.36
$\theta_{\mathrm{under}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.245	0.244	2.68	2.52	0.281	0.284	0.263	0.269
φ_{exact}	99.7%	99.7%	19.9%	19%	56.5%	56.5%	56.4%	56.4%
$T_{\mathrm{exact}}^{\mathrm{RMSE}}$ (m)	2.23	2.22	43.77	52.87	2.95	2.96	2.99	3.00
$\theta_{\mathrm{exact}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.275	0.273	4.39	4.35	0.377	0.380	0.388	0.391
$T_{\mathrm{all}}^{\mathrm{RMSE}}$ (m)	2.23	2.21	37.59	35.14	2.76	2.77	2.77	2.79
$\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.275	0.273	2.83	2.71	0.352	0.354	0.361	0.364
t (s)	2.22	2.22	4.88	4.92	1.97	1.95	2.21	2.19

the Bernoulli measurement model is the more realistic one, the target measurements are generated in the next evaluation according to this Bernoulli measurement model for comparing all the algorithms. Therefore, there is no measurement model mismatch for the assignment algorithms but there is a measurement model mismatch for the EM-based algorithms.

Tables 3.6.10–3.6.12 compare the assignment algorithms and EM-based algorithms with assignment based initialization. In this case, one may consider the EM-based algorithms as post-processing procedures. Such a processing increases the entire processing time and only leads to an insignificant improvement of the estimation accuracies. However, it reflects the capability of the EM-based algorithm to solve the data association problem. Associating the measurements to a good degree of accuracy requires a good initialization, such as the assignment approach. In this case, although there is a mismatch in the mea-



(A) φ_{exact} (B) $T_{\text{all}}^{\text{RMSE}}$

(C) $\theta_{\text{all}}^{\text{RMSE}}$

FIGURE 3.6.6: The performance (in terms of φ_{exact}, $T_{\text{all}}^{\text{RMSE}}$ and $\theta_{\text{all}}^{\text{RMSE}}$) of UGM/EM and PPP/EM in scenarios with different known expected number of false alarms (0.25, 0.5, 1, 2, 4 and 8) for true p_{d} values at 0.7, 0.8 and 0.9.

surement model, the EM-based algorithms estimate the number of targets, target locations and target directions quite accurately due to the fact that the initialization by the sequential m-best 2-D assignment algorithm is close to the truth.

3.7 Conclusions

This chapter considers the problem of multiple transient emitter localization using a network of passive sensors. It is assumed that the number of targets as well as the associa-

TABLE 3.6.10: Performance comparison among S-D assignment, sequential m-best 2-D (SmB) assignment and EM-based algorithms (with different initializations "I:") for $p_{\rm d} = 0.9$ ($N_{\rm s}$=6)

	Assignment			I: SmB(m=2)		I: SmB(m=4)	
	S-D	SmB(m=2)	SmB(m=4)	UGM/EM	PPP/EM	UGM/EM	PPP/EM
$\varphi_{\rm over}$	2.0%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%
$\bar{M}_{\rm over}$	1	1	1	1	1	1	1
$T_{\rm over}^{\rm RMSE}$ (m)	14.24	3.55	3.55	3.54	3.54	3.54	3.54
$\theta_{\rm over}^{\rm RMSE}$ ($^{\rm o}$)	0.640	0.306	0.306	0.307	0.307	0.307	0.307
$\varphi_{\rm under}$	1.9%	0.4%	0.3%	0.4%	0.4%	0.3%	0.3%
$\bar{M}_{\rm under}$	1	1	1	1	1	1	1
$T_{\rm under}^{\rm RMSE}$ (m)	18.14	2.86	2.91	2.86	2.86	2.91	2.91
$\theta_{\rm under}^{\rm RMSE}$ ($^{\rm o}$)	0.792	0.279	0.286	0.279	0.279	0.286	0.286
$\varphi_{\rm exact}$	96.1%	99.2%	99.3%	99.2%	99.2%	99.3%	99.3%
$T_{\rm exact}^{\rm RMSE}$ (m)	9.44	4.42	4.42	4.41	4.41	4.40	4.40
$\theta_{\rm exact}^{\rm RMSE}$ ($^{\rm o}$)	0.832	0.318	0.318	0.318	0.318	0.318	0.318
$T_{\rm all}^{\rm RMSE}$ (m)	9.74	4.41	4.42	4.40	4.40	4.40	4.40
$\theta_{\rm all}^{\rm RMSE}$ ($^{\rm o}$)	0.828	0.318	0.318	0.318	0.318	0.318	0.318
t (s)	0.568	0.096	0.162	0.181	0.190	0.249	0.247

tion between measurements and targets are unknown in the presence of missed detections and false alarms. Two different measurement models — the Bernoulli measurement model and the Poisson measurement model — are considered for each target and two types of algorithms — assignment and EM based are presented, one for each measurement model. Simulation studies show that the sequential m-best 2-D assignment algorithm has very promising performance and can be employed in real time applications. While the EM-based algorithms have the capability of solving the data association problem, simulation results suggest that they require the right initial estimates to provide reliable localization results and the processing time could be longer than required. The fusion algorithms discussed in this chapter assume that there is a fusion center to which each sensor can communicate. In the next chapter, fusion algorithms, which assume that no such fusion center exists and allow distributed processing and only single-hop communication, will be developed.

TABLE 3.6.11: Performance comparison among S-D assignment, sequential m-best 2-D (SmB) assignment and EM-based algorithms (with different initializations "I:") for $p_{\mathrm{d}} = 0.8$ (N_{s}=6)

	Assignment			I: SmB(m=2)		I: SmB(m=4)	
	S-D	SmB(m=2)	SmB(m=4)	UGM/EM	PPP/EM	UGM/EM	PPP/EM
φ_{over}	0.6%	0.5%	0.4%	0.4%	0.3%	0.3%	0.3%
$\bar{M}_{\mathrm{over}}$	1	1	1	1	1	1	1
$T_{\mathrm{over}}^{\mathrm{RMSE}}$ (m)	4.21	6.75	6.74	5.69	6.27	6.26	6.26
$\theta_{\mathrm{over}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	0.302	0.423	0.421	0.367	0.395	0.396	0.395
φ_{under}	25.4%	7.2%	7.5%	7.2%	7.2%	7.5%	7.5%
$\bar{M}_{\mathrm{under}}$	1.23	1.03	1.03	1.03	1.03	1.03	1.03
$T_{\mathrm{under}}^{\mathrm{RMSE}}$ (m)	30.12	5.55	5.49	5.19	5.19	5.14	5.14
$\theta_{\mathrm{under}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	1.58	0.369	0.367	0.359	0.359	0.358	0.358
φ_{exact}	74.0%	92.3%	92.1%	92.4%	92.5%	92.2%	92.2%
$T_{\mathrm{exact}}^{\mathrm{RMSE}}$ (m)	12.86	5.15	5.11	5.08	5.08	5.05	5.05
$\theta_{\mathrm{exact}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	1.06	0.362	0.359	0.358	0.358	0.356	0.356
$T_{\mathrm{all}}^{\mathrm{RMSE}}$ (m)	17.49	5.18	5.14	5.09	5.09	5.06	5.06
$\theta_{\mathrm{all}}^{\mathrm{RMSE}}$ ($^{\mathrm{o}}$)	1.18	0.362	0.360	0.358	0.359	0.356	0.356
t (s)	0.420	0.070	0.141	0.151	0.148	0.231	0.237

TABLE 3.6.12: Performance comparison among S-D assignment, sequential m-best 2-D (SmB) assignment and EM-based algorithms (with different initializations "I:") for $p_{\rm d} = 0.7$ ($N_{\rm s}$=6)

	Assignment			I: SmB(m=2)		I: SmB(m=4)	
	S-D	SmB(m=2)	SmB(m=4)	UGM/EM	PPP/EM	UGM/EM	PPP/EM
$\varphi_{\rm over}$	0.3%	0	0	0	0	0	0
$\bar{M}_{\rm over}$	1	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
$T_{\rm over}^{\rm RMSE}$ (m)	6.30	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
$\theta_{\rm over}^{\rm RMSE}$ ($^{\rm o}$)	0.382	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
$\varphi_{\rm under}$	57.4%	27.6%	27.5%	27.6%	27.6%	27.5%	27.5%
$\bar{M}_{\rm under}$	1.46	1.14	1.14	1.14	1.14	1.14	1.14
$T_{\rm under}^{\rm RMSE}$ (m)	35.24	7.10	6.96	5.69	5.68	5.65	5.65
$\theta_{\rm under}^{\rm RMSE}$ ($^{\rm o}$)	2.19	0.402	0.400	0.383	0.383	0.381	0.382
$\varphi_{\rm exact}$	42.3%	72.4%	72.5%	72.4%	72.4%	72.5%	72.5%
$T_{\rm exact}^{\rm RMSE}$ (m)	14.90	7.44	6.96	7.27	7.27	6.88	6.86
$\theta_{\rm exact}^{\rm RMSE}$ ($^{\rm o}$)	0.911	0.872	0.860	0.864	0.864	0.851	0.851
$T_{\rm all}^{\rm RMSE}$ (m)	26.29	7.37	6.96	6.96	6.96	6.63	6.62
$\theta_{\rm all}^{\rm RMSE}$ ($^{\rm o}$)	1.63	0.795	0.785	0.786	0.786	0.775	0.775
t (s)	0.318	0.058	0.127	0.126	0.133	0.191	0.196

Chapter 4

Distributed Fusion Algorithm for Passive Localization of Multiple Transient Emitters

4.1 Introduction

4.1.1 Background

This chapter considers the problem of multiple transient emitter (target) localization using a wireless sensor network (WSN). One particular application is to utilize a network of acoustic gunfire detection systems mounted on a group of soldiers to localize adversaries in a battlefield [31] [32]. It is assumed that the targets are stationary during the time window of interest but the number of the targets is unknown. The sensors can measure the line of sight (LOS) angles to the targets by detecting their emitted acoustic signals and record the times of arrival (TOAs) of the detected signals. This implies incomplete target location

observability for any single sensor. Missed detections and false alarms are present due to the imperfection of the sensors. Furthermore, the associations between the measurements and the targets are unknown, that is, each sensor does not know from which target (or clutter) a particular measurement originates. Before estimating the position of any target, one has to associate the measurements from all the sensors. Therefore, the quality of data association is critical to the overall localization performance.

Two different fusion algorithms developed in our previous work [25] solved this problem using a centralized approach, i.e., we assumed that there is a fusion center collecting all the information from individual sensors either directly or by multi-hop relay, typically by wireless communication. Centralized access to all information can be difficult. For example, it requires a high transmission power to deliver the information from a single sensor directly to a fusion center in applications covering a large area. Moreover, the fusion center based approach is not robust, i.e., if the fusion center fails, the whole system fails. This has motivated a lot of work on distributed fusion or distributed optimization algorithms including the one presented in this chapter.

One straightforward distributed solution is flooding, i.e., broadcasting the actual sensor measurements through the links in the network. In [13], a communication strategy of broadcasting new measurements was presented to allow distributed measurement fusion, which produces the optimal estimate at each node given all the measurements received up to any time for a linear dynamic system. For the localization problem considered in this chapter, one has a nonlinear static system. The flooding approach still applies, by careful bookkeeping and a number of iterations of information exchange, each sensor would have all the information and can act as a fusion center to find the same global solution as a centralized approach. This method requires a large amount of data communication, storage memory, and bookkeeping overhead. For instance, it requires about S (the number of

TABLE 4.1.1: Classification of the various versions of the shooter localization problem.

	Single target	Multiple targets
No missed detections or false alarms	P1	P5
Only missed detections exist	P2	P6
Only false alarms exist	P3	P7
Missed detections and false alarms exist	P4	P8

sensors) times the memory storage of the average consensus (AC) based approach.

When it is used for the localization problem, the flooding approach is distributed in the sense that the information (all the measurements) is communicated in a distributed manner but it is centralized in the sense that the estimation algorithm including all computations is applied on all the information collected at every node, i.e., the flooding approach is a multiple replica of the centralized approach. In this chapter, we present a consensus based algorithm that is different from the flooding approach and is distributed in the sense that both communication and estimation are performed in a distributed manner.

One of our approaches in [25] formulated the localization problem as an optimization problem and solved it using the expectation maximization (EM) algorithm. We observe that two types of subproblem are solved in the EM algorithm. One is to compute the average of variables with one variable from one sensor and the other is to solve a nonlinear least squares problem. Both subproblems can be formulated to optimize a global objective function, which can be written as a sum of local objective functions. Such problems can be solved using distributed optimization approaches whose goal is to recover the optimal global solution without any global coordination or interactions (like using a fusion center). Their solutions often contain a step where the sum or average of some quantity needs to be calculated and this can be achieved by an average consensus (AC) based approach.

The average consensus based approach with communication only between the one-hop neighbors scales well in that the communication overhead per sensor can be kept at an af-

fordable level as the size of the network increases. Unlike the full flooding approach, which requires the local variables labeled with their origins, the average consensus approach does not need such labels and therefore uses less storage. If new nodes join the network, our consensus based distributed algorithm does not need to restart the whole process because the local variables can be updated following a (mini) flooding of only the new information.

In this chapter, we assume that centralized access of all the information is not possible and we are interested in solving the problem of multiple transient emitter localization using an alternative algorithm that is different from the flooding approach and that is distributed in the sense that both communication and estimation are performed in a distributed manner. Since the goal is to have each sensor obtain a global estimate (which is a vector consisting of the number of targets and the position estimates of all targets) as good (or almost as good) as can be obtained by a fusion center using a centralized algorithm, information diffusion either in the form of raw measurements or in the form of some intermediate estimates (a function of raw measurements) within the network is necessary. Instead of using the raw measurement diffusion approach as in the flooding approach, we diffuse the intermediate estimates using the average consensus approach, i.e., the estimation is also performed in a distributed manner.

Without a fully connected network (each node can reach each other node via one or multiple "hops"), sending raw measurements to all nodes in order to achieve global optimal solution is a difficult task which requires "subnetwork" coordination, which is beyond the scope of this chapter (multiple layers would be necessary). Therefore, we assume that the network is fully connected, i.e., there is a (not necessarily direct) path between every two sensors. If the network is not connected and has more than one connected subnetwork due to node or link failures, each subnetwork can be processed by our distributed algorithm independently. In such case, the consensus is achieved within each connected subnetwork.

Table 4.1.1 presents a classification of the various versions of the shooter localization problem. In view of the above discussion, it is necessary to develop a distributed algorithm to solve the problem P8 in Table 4.1.1 relying solely on local communications between one-hop neighboring sensors. Problems P3, P4, P6 and P7 are special cases of P8, therefore can be solved by the same distributed algorithm. Problem P1 is addressed in Section 4.2.7. Problems P2 are P5 are special extensions of P1 and will not be covered here.

4.1.2 Related Work

Distributed data fusion strategies, such as methods in [12] [14] [15] [16] [28] [40] among others, are available for joint state estimation and data association in multi-sensor multi-target tracking scenarios. Since they are recursive algorithms that require sequential measurements and provide solutions to dynamic data association problems, they cannot be employed to solve joint parameter estimation and data association in a multi-sensor multi-target localization scenario (with incomplete observability at each sensor) considered in this chapter. While most of the distributed estimation work in the literature assumes linear measurement models, this chapter deals with nonlinear and incomplete target location measurements (direction of arrival and delayed arrival time). Although, one could imagine linearizing the localization problem and sharing messages between the nodes, we suspect that the linearization will probably cause more errors than the distributed ADMM and will investigate this in our future work. Related work from robotics can be found in [35]. A recent comparison of optimal distributed estimation and consensus filtering for dynamic systems was done in [13].

A multi-dimensional assignment formulation assuming a Bernoulli measurement generation model that the number of measurements from each target received at each sensor

is a Bernoulli random variable with parameter equal to the probability of detection as well as a cardinality selection formulation assuming a Poisson measurement generation model that the number of measurements from each target received at each sensor is a Poisson random variable with parameter equal to the probability of detection were considered in the centralized fusion algorithms [25] to solve the same problem of multiple transient emitter localization. This chapter only considers developing a distributed algorithm to solve the cardinality selection problem assuming a Poisson measurement generation model[1] and leaves distributed multi-dimensional assignment algorithms for future work. While a list of measurements at each sensor was modeled as either realizations of a random variable with a mixture density or a Poisson point process (PPP) in [25], only PPP modeling is considered in this chapter due to its simpler mathematical solution expression. Since the centralized algorithm solving the cardinality selection problem, which combines expectation maximization (EM) algorithm to estimating target parameters given a fixed number of targets and information criterion for selection of the best possible number of targets, is not amendable to a distributed implementation, it is necessary to develop a distributed EM algorithm.

Distributed EM algorithms have attracted a lot of attentions in sensor network applications for density estimation, data clustering and target tracking. For a fixed number of target, the localization problem can be considered as a density estimation problem. An incremental distributed EM algorithm presented in [44] is the first known scheme for density estimation and clustering in distributed sensor network. A distributed EM algorithm based on the averaging consensus filtering was developed in [34] for particle filter based target

[1]The Bernoulli measurement generation model is more realistic than the Poisson measurement generation model. Therefore, the Bernoulli model is used to generate the synthetic data for the evaluation of the developed algorithm, whereas the Poisson model is assumed in the derivation of the developed algorithm. Using the Bernoulli model in the algorithm would make it excessively complicated because of the need to use multidimensional assignment.

tracking. A distributed EM algorithm based on alternating direction method of multipliers (ADMM) was proposed in [29] for distributed data clustering. However, all these works assumes a *linear* generative model for their respective applications, which does not apply to a *nonlinear* generative model (see the measurement model in (4.3.6)) considered here due to the incomplete position measurement based on bearings and TOAs in the emitter localization scenario considered in this chapter. Moreover, the parameters in these distributed EM algorithms are initialized to be either fixed values (zeros) or random values. This initialization approach was shown to be useless for our centralized EM algorithm, which requires an initialization based on the sequential m-best 2-D assignment algorithm applied on the lists of measurements from all sensors for the convergence to the global maximum.

4.1.3 Contributions

In this chapter, we develop a distributed EM algorithm to solve the same problem as considered in [25] but in a distributed manner. The distributed processing introduces a number of challenges.

Firstly, the convergence of an EM algorithm (whether being centralized or distributed) depends highly on the initialization step. Previous studies on developing distributed EM algorithms assumed a linear measurement model and thus the initialization with fixed values (such as zeros) or random values, which is commonly used, works fine. This initialization does not work in the problem considered in this chapter where the measurements (incomplete position observations) are nonlinear functions of target locations. Our earlier work shows that the assignment based initialization leads to global convergence. However, due to limited connections in a distributed setting, each sensor can only obtain a different EM initialization, which is a set of vectors, using the sequential m-best 2-D assignment algorithm

on the measurement lists of its own and its neighbors (a subset of all the lists of measurements). For the global convergence of the EM algorithm, we developed a distributed set consensus algorithm ensuring that every node has the same initialization (the same number of targets and the same target locations).

Secondly, the maximization step in the standard EM algorithm has to be evaluated in a distributed manner. Although the probability of detection can be estimated by a distributed averaging consensus subroutine and the locations of the targets can be estimated by a distributed ADMM subroutine, this would result in a nested iterative algorithm with two subroutines being iterative algorithms themselves. Even more challenging, these two subroutines are needed for a number of iterations and at each iteration both of them requires local communications between sensors for a number of times, which would result in a very high communication cost. Instead, we manage to formulate a constrained optimization problem with equality conditions that force all local variables to be identical and developed a new distributed ADMM algorithm enabling a lower communication cost at the expense of additional local computation. The EM and AC based distributed ADMM algorithm is a generalization of previous distributed algorithms allowing the handling of the nonlinear and incomplete measurement models such as bearings in the passive sensing applications as here.

Last but not least, since we feel that a Bernoulli measurement generation model is a more realistic assumption and it reflects best the physical process of measurement generation, we used a likelihood function based thresholding approach to determine the number of targets.

4.1.4 Chapter Organization

The remaining sections of this chapter are organized as follows. Section 4.2 presents some preliminaries required for the development of the desired distributed algorithm. These include (i) graph modeling, (ii) a distributed AC algorithm for both single parameter estimation and multiple parameter estimation, (iii) data association test for two estimates as well as two sets of estimates, (iv) an algorithm of alternating direction method of multipliers and (v) a distributed nonlinear least squares algorithm, which can solve problem P1 in Table 4.1.1. Section 4.3 formulates the problem by modeling each measurement set as a realization of a Poisson point process. Section 4.4 reviews a recently developed centralized algorithm that uses an EM algorithm to estimate the location and emission time parameters for a fixed number of targets. The distributed algorithm for problem P8 is presented in Section 4.5. The initialization issues of this algorithm — how to reach the consensus on the number of targets and how to reach the consensus on the target-estimate association — are discussed in Sections 4.5.1 and 4.5.2, respectively. An EM and AC based distributed ADMM algorithm is developed in Section 4.5.3. Section 4.5.4 describes a thresholding approach to distinguish real target estimates from false target estimates using the estimated probability of detection values. Section 4.6 presents and analyzes simulation results and Section 4.7 concludes the chapter.

4.2 Preliminaries

4.2.1 Graph Model

A wireless sensor network with S nodes (sensors) is deployed to collect data and perform data association and parameter estimation task. Every node is only able to communicate with its neighbors. Mathematically, this network can be modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the set of nodes

$$\mathcal{V} = \{\upsilon_1, \upsilon_2, \ldots, \upsilon_S\} \tag{4.2.1}$$

and the set of edges $\mathcal{E}$, where an edge $(\upsilon_i, \upsilon_j) \in \mathcal{E}$ is an unordered pair of distinct nodes, representing a two-way communication link between υ_i and υ_j. The graph $\mathcal{G}$ is assumed connected, meaning that there is a path between any two nodes. The set of neighbors of node υ_i is defined as

$$\mathcal{N}_i = \{\upsilon_j \in \mathcal{V} : (\upsilon_i, \upsilon_j) \in \mathcal{E}\} \tag{4.2.2}$$

The degree of node υ_i is defined as

$$d_i = |\mathcal{N}_i| \tag{4.2.3}$$

where $|\cdot|$ denotes the set cardinality. The maximum degree of the graph $\mathcal{G}$ is defined as

$$d_{\max} = \max_i d_i \tag{4.2.4}$$

The Laplacian matrix L of the graph $\mathcal{G}$ is defined as

$$L_{ij} = \begin{cases} -1 & \text{if } \upsilon_j \in \mathcal{N}_i \\ d_i & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{4.2.5}$$

4.2.2 Distributed Averaging Consensus Algorithm

Suppose a wireless sensor network with S nodes is deployed to estimate an unknown constant parameter $x \in \mathbf{R}^n$. Each node υ_i makes a measurement

$$z_i = x + w_i \tag{4.2.6}$$

where w_i are independent, identically distributed, normal, zero mean, and with a known identity covariance matrix I. The maximum likelihood estimate of x is $\frac{1}{S}\sum_{i=1}^{S} z_i$, which is the mean vector of all measurements z_i. This estimate can be obtained by the following distributed averaging consensus algorithm.

Let us denote an initial value (z_i for the estimate problem) at node υ_i by $u_i(0) \in \mathbf{R}^n$ at time $t = 0$. The matrix formed by the column vectors at all nodes is denoted as

$$\mathbf{U}(0) = \begin{bmatrix} u_1(0) & u_2(0) & \dots & u_S(0) \end{bmatrix}^T \in \mathbf{R}^{S \times n} \tag{4.2.7}$$

The goal of distributed averaging consensus is to make every node obtain the mean vector $\frac{1}{S}\sum_{i=1}^{S} u_i(0)$ eventually after gradually updating its value with a linear combination of its previously stored value and the values of its neighbors. One iteration of the process can be

represented with a weight matrix W as

$$u_i(t+1) = W_{ii}u_i(t) + \sum_{j \in \mathcal{N}_i} W_{ij}u_j(t) \qquad i = 1, \ldots, S \tag{4.2.8}$$

where t = 0,1,... is the discrete time index, and W_{ij} is the weight on u_j at node υ_i. Setting $W_{ij} = 0$ for $j \notin \mathcal{N}_i$, this iteration can be written in matrix form as

$$\mathbf{U}(t+1) = W\mathbf{U}(t) \tag{4.2.9}$$

and W is selected such that

$$\lim_{t \to \infty} \mathbf{U}(t) = \frac{1}{S}\mathbf{1}\mathbf{1}'\mathbf{U}(0) \tag{4.2.10}$$

The best constant edge weight matrix is given by [59]

$$W = I - \beta L \tag{4.2.11}$$

with

$$\beta = \frac{2}{\eta_1(L) + \eta_{S-1}(L)} \tag{4.2.12}$$

where $\eta_1(L)$ and $\eta_{S-1}(L)$ are the largest and second smallest eigenvalues of L, respectively.

In some cases, each node only has the knowledge of its neighbors rather than the connectivity of the entire network. It is more suitable to use the Metropolis weight matrix,

which is defined as [60]

$$W_{ij} = \begin{cases} \dfrac{1}{1+\max\{d_i, d_j\}} & \text{if } \upsilon_j \in \mathcal{N}_i \\ 1 - \sum_{\upsilon_k \in \mathcal{N}_i} W_{ik} & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{4.2.13}$$

4.2.3 Distributed Averaging Consensus Algorithm for Multiple Parameter Estimation with Unknown Data Association

Suppose a WSN with S nodes is used to estimate a set of N unknown constant parameters

$$X = \{x_1, x_2, \ldots, x_N\} \tag{4.2.14}$$

with each $x_j \in \mathbf{R}^n$. Each node υ_i has a set of N measurements

$$Z_i = \{z_{i1}, z_{i2}, \ldots, z_{iN}\} \tag{4.2.15}$$

with one for each x_j. Let Π_N denote all permutations of the set $\{1, 2, \ldots, N\}$, then the jth measurement of node υ_i is

$$z_{ij} = x_{\pi_i(j)} + w_i \tag{4.2.16}$$

where $\pi_i \in \Pi_N$ is a permutation[2] at node υ_i, and w_i are independent, identically distributed, normal, zero mean measurement noises with a known identity covariance matrix I.

[2]It is a one-to-one mapping function from an ordered set $\{1, 2, \ldots, N\}$ to a particular permutation of this set.

Since the second index j of z_{ij} in the set Z_i contains no labeling information, one needs to perform data association and weighted averaging update (4.2.8) simultaneously for multiple parameter estimation. Let us denote the stacked vector at node υ_i at time t as

$$\mathbf{u}_i(t) = \left[u_{i1}^T(t), u_{i2}^T(t), \ldots, u_{iN}^T(t)\right]^T \tag{4.2.17}$$

and $u_{ij}(0)$ is initialized as z_{ij}. At time t, node υ_i calculates an optimal permutation[3] π_{ji} for each of its neighbor nodes υ_j as

$$\pi_{ji} = \arg\min_{\pi \in \Pi_N} \sum_{k=1}^{N} \|u_{ik}(t) - u_{j\pi(k)}(t)\|^2 \tag{4.2.18}$$

Then node υ_i updates each segment of its stacked vector (4.2.17) as

$$u_{ik}(t+1) = W_{ii}u_{ik}(t) + \sum_{j \in \mathcal{N}_i} W_{ij}u_{j\pi_{ji}(k)}(t) \tag{4.2.19}$$

where the index $\pi_{ji}(k)$ refers to the segment of the stacked vector at node υ_j that associates with the k-th segment of the stacked vector at node υ_i according to the permutation π_{ji} (4.2.18), and the weight matrix is given by (4.2.13).

4.2.4 Association Test for Two Estimates

Suppose that sensor υ_i has an unbiased estimate $\hat{x}_i$ of the n-dimensional (unknown) parameter x_i with a covariance matrix P_i and sensor υ_j has an unbiased estimate $\hat{x}_j$ of the n-dimensional (unknown) parameter x_j with a covariance matrix P_j. We are interested in

[3]The second index i of π_{ji} indicates that the optimal permutation is obtained with respect to $\mathbf{u}_i(t)$.

testing whether $x_i = x_j$. Let us denote the difference of the two estimates as

$$\hat{\Delta}_{ij} = \hat{x}_i - \hat{x}_j \tag{4.2.20}$$

which is the estimate of the difference of the parameters

$$\Delta_{ij} = x_i - x_j \tag{4.2.21}$$

Since the estimation errors

$$\tilde{x}_i = x_i - \hat{x}_i \tag{4.2.22}$$

$$\tilde{x}_j = x_j - \hat{x}_j \tag{4.2.23}$$

are zero-mean, the estimation error of the difference of the parameters

$$\tilde{\Delta}_{ij} = \Delta_{ij} - \hat{\Delta}_{ij} = \tilde{x}_i - \tilde{x}_j \tag{4.2.24}$$

is also zero-mean and it has the covariance matrix

$$\begin{aligned} T_{ij} &= E\{\tilde{\Delta}_{ij}\tilde{\Delta}_{ij}^T\} = E\{(\tilde{x}_i - \tilde{x}_j)(\tilde{x}_i - \tilde{x}_j)^T\} \\ &= P_i + P_j - E\{\tilde{x}_i\tilde{x}_j^T\} - E\{\tilde{x}_j\tilde{x}_i^T\} \end{aligned} \tag{4.2.25}$$

If $\tilde{x}_i$ and $\tilde{x}_j$ are independent, then we have

$$T_{ij} = P_i + P_j \tag{4.2.26}$$

Assuming that $\tilde{x}_i$ and $\tilde{x}_j$ are Gaussian, the normalized estimation error squared (NEES) [4] for Δ

$$\epsilon_{ij} \triangleq \tilde{\Delta}_{ij}^T T_{ij}^{-1} \tilde{\Delta}_{ij} \tag{4.2.27}$$

is chi-square distributed with n degrees of freedom.

The null hypothesis that the two parameters are the same and the alternative hypothesis are

$$H_0 : \Delta = 0 \tag{4.2.28}$$

$$H_1 : \Delta \neq 0 \tag{4.2.29}$$

Under H_0 ($\Delta = 0$), we have the following

$$\tilde{\Delta}_{ij} = -\hat{\Delta}_{ij} \tag{4.2.30}$$

$$\epsilon_{ij} = \hat{\Delta}_{ij}^T T_{ij}^{-1} \hat{\Delta}_{ij} \tag{4.2.31}$$

Therefore, the test of H_0 vs. H_1 is as follows. If

$$\hat{\Delta}_{ij}^T T_{ij}^{-1} \hat{\Delta}_{ij} \leq F_{\chi_n^2}^{-1}(1-\alpha) \tag{4.2.32}$$

where $F_{\chi_n^2}^{-1}(\cdot)$ is the inverse of the cumulative distribution function (cdf) of a chi-square random variable with n degrees of freedom, we will not reject H_0 at a significance level of α. Then it is likely that $\hat{x}_i$ and $\hat{x}_j$ are estimates of the same parameter.

4.2.5 Association Test for Two Sets of Estimates

Suppose that there are N unknown n-dimensional constant parameters

$$X = \{x_1, x_2, \ldots, x_N\} \tag{4.2.33}$$

Sensor υ_i has a set of N_i estimates with corresponding covariance matrices

$$\hat{X}_i = \{\hat{x}_{i1}, \hat{x}_{i2}, \ldots, \hat{x}_{iN_i}\} \tag{4.2.34}$$

$$\mathcal{P}_i = \{P_{i1}, P_{i2}, \ldots, P_{iN_i}\} \tag{4.2.35}$$

Similarly, sensor υ_j has N_j estimates with corresponding covariance matrices

$$\hat{X}_j = \{\hat{x}_{j1}, \hat{x}_{j2}, \ldots, \hat{x}_{jN_j}\} \tag{4.2.36}$$

$$\mathcal{P}_j = \{P_{j1}, P_{j2}, \ldots, P_{jN_j}\} \tag{4.2.37}$$

We assume that each sensor has at most one estimate for a particular parameter and the estimation errors are independent.

If $\hat{x}_{ik}$ and $\hat{x}_{j\ell}$ are estimates of the same parameter, then the NEES

$$d_{k\ell} = (\hat{x}_{ik} - \hat{x}_{j\ell})^T (P_{ik} + P_{j\ell})^{-1} (\hat{x}_{ik} - \hat{x}_{j\ell}) \tag{4.2.38}$$

can be regarded as a distance measure between $\hat{x}_{ik}$ and $\hat{x}_{j\ell}$. A small value of $d_{k\ell}$ indicates a high probability of both being the estimates of the same parameter.

To deal with incomplete associations caused by missed detections, we add dummy estimates $\hat{x}_{i0}$ and $\hat{x}_{j0}$ to the sets $\hat{X}_i$ and $\hat{X}_j$, respectively [48]. The distance involving a dummy

estimate is defined as

$$d_{k0} = d_{0\ell} = F^{-1}_{\chi^2_n}(1-\alpha) \tag{4.2.39}$$

for a small value (say, 0.01) of α.

To associate the estimates in set $\hat{X}_i$ with those in set $\hat{X}_j$, we solve a generalized 2-D assignment problem

$$\min_{\rho_{k\ell}} \sum_{k=0}^{N_i} \sum_{\ell=0}^{N_j} \rho_{k\ell} d_{k\ell} \tag{4.2.40}$$

subject to

$$\sum_{\ell=0}^{N_j} \rho_{k\ell} = 1 \quad \forall\, k = 1, 2, \ldots, N_i \tag{4.2.41}$$

$$\sum_{k=0}^{N_i} \rho_{k\ell} = 1 \quad \forall\, \ell = 1, 2, \ldots, N_j \tag{4.2.42}$$

$$\rho_{k\ell} \in \{0, 1\} \qquad k = 0, 1, \ldots, N_i;\ \ell = 0, 1, \ldots, N_j \tag{4.2.43}$$

The modified auction algorithm [48] can be applied to the above problem.

The association results of $\hat{x}_{ik}$ are determined as follows.

If

$$\rho_{k0} = 1 \tag{4.2.44}$$

then $\hat{x}_{ik}$ is assigned to the dummy estimate $\hat{x}_{j0}$, that is, the probability that no estimate in $\hat{X}_j$ comes from the same parameter as $\hat{x}_{ik}$ is 0.99 for $\alpha = 0.01$. In this case, $\hat{x}_{ik}$ is not

associated.

If

$$\rho_{k\ell} = 1 \tag{4.2.45}$$

then $\hat{x}_{ik}$ is associated with $\hat{x}_{j\ell}$.

The association results of $\hat{x}_{j\ell}$ are determined in a similar way.

4.2.6 The Alternating Direction Method of Multipliers (ADMM) Algorithm

Consider the following equality-constrained optimization problem

$$\min_{z,\,y} \{f(z) + g(y)\} \tag{4.2.46}$$

subject to

$$Az + By = c \tag{4.2.47}$$

with variables $z \in \mathbf{R}^p$ and $y \in \mathbf{R}^q$, where $A \in \mathbf{R}^{m\times p}$, $B \in \mathbf{R}^{m\times q}$ and $c \in \mathbf{R}^m$ are given.

The augmented Lagrangian of (4.2.46) is defined as

$$\begin{aligned} L_\rho(z, y, \lambda) =& f(z) + g(y) + \lambda^T (Az + By - c) \\ &+ \frac{\rho}{2} \|Az + By - c\|_2^2 \end{aligned} \tag{4.2.48}$$

where λ is the dual variable or Lagrange multiplier and $\rho > 0$ is the penalty parameter.

The ADMM algorithm [9] solves (4.2.46) by iterating the following 3 steps

$$z^{k+1} \triangleq \arg\min_{z} L_\rho\left(z, y^k, \lambda^k\right) \qquad z\text{-minimization} \tag{4.2.49}$$

$$y^{k+1} \triangleq \arg\min_{y} L_\rho\left(z^{k+1}, y, \lambda^k\right) \qquad y\text{-minimization} \tag{4.2.50}$$

$$\lambda^{k+1} \triangleq \lambda^k + \rho\left(Az^{k+1} + By^{k+1} - c\right) \qquad \text{dual update} \tag{4.2.51}$$

where ρ is used as the step size for the dual update and the superscript is the iteration counter.

In the ADMM, the variables z and y are updated in an alternating or sequential fashion instead of being minimized jointly, which accounts for the term *alternating direction*. Separating the minimization over z and y into two steps is precisely what allows for decomposition when f (or g) is separable with respect to a partition of the variable z (or y) into subvectors.

4.2.7 Distributed Nonlinear Least Squares Algorithm

This subsection presents a distributed solution to the problem P1 in Table 4.1.1. We are interested in localizing a single target using the network $\mathcal{G}$ without missed detections or false alarms. Suppose each node v_i has a scalar measurement a_i from the target, we need to solve the unconstrained optimization problem

$$\min_{x} \quad \sum_{i=1}^{S} \left(h(x) - a_i\right)^2 \tag{4.2.52}$$

where $x \in \mathbf{R}^2$ is the parameter to be estimated (or the variable for the minimization), $h(\cdot)$ is a nonlinear function of x (for instance, $h(x)$ is an $\arctan$ function in a bearing-only

localization problem) and S is the number of sensors.

Consider the constrained optimization problem, which is equivalent to (4.2.52)

$$\min_{x_1, x_2, ..., x_S} \quad \sum_{i=1}^{S} \left(h(x_i) - a_i\right)^2 \tag{4.2.53}$$

subject to

$$x_1 = x_2 = \ldots = x_S = w \tag{4.2.54}$$

We can put (4.2.54) in the form of (4.2.47) by setting

$$z = [x_1^T \; x_2^T \; \ldots x_S^T]^T \tag{4.2.55}$$

$$y = w \tag{4.2.56}$$

$$f(z) = \sum_{i=1}^{S} \left(h(x_i) - a_i\right)^2 \tag{4.2.57}$$

$$g(y) = 0 \tag{4.2.58}$$

$$A = I_{2S} \tag{4.2.59}$$

$$B = \begin{bmatrix} -I_2 \; -I_2 \cdots -I_2 \end{bmatrix}^T \in \mathbf{R}^{2S \times 2} \tag{4.2.60}$$

$$c = 0 \tag{4.2.61}$$

Therefore, the augmented Lagrangian is

$$\begin{aligned} &L_\rho\left(x_1, x_2, \ldots, x_S, w, \lambda\right) = \\ &\sum_{i=1}^{S} \left[\left(h(x_i) - a_i\right)^2 + \lambda_i^T (x_i - w) + \frac{\rho}{2} \left\| x_i - w \right\|_2^2 \right] \end{aligned} \tag{4.2.62}$$

TABLE 4.2.1: Averaging consensus based distributed ADMM algorithm.

Node v_i initializes x_i^1 and $\lambda_i^1 = 0$
Compute $\overline{x}^1 = \frac{1}{S}\sum_{i=1}^{S} x_i^1$ via a distributed averaging consensus algorithm
for $k = 1, 2, \ldots$ **do** until convergence
    for all v_i **do**
        Compute x_i^{k+1} via (4.2.71)
        Compute $\overline{x}^{k+1} = \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1}$ via a distributed averaging consensus algorithm
        Compute λ_i^{k+1} via (4.2.72)
    end for
end for

where

$$\lambda = [\lambda_1^T \ \lambda_2^T \ \ldots \lambda_S^T]^T \tag{4.2.63}$$

The z-minimization step (4.2.49) is

$$\begin{aligned} &\left(x_1^{k+1},\, x_2^{k+1},\, \ldots,\, x_S^{k+1}\right) = \\ &\arg \min_{x_1, x_2, \ldots, x_S} L_\rho\left(x_1, x_2, \ldots, x_S, w^k, \lambda^k\right) \end{aligned} \tag{4.2.64}$$

which can be carried out in a distributed fashion as

$$\begin{aligned} x_i^{k+1} = \arg\min_{x_i} &\left(h(x_i) - a_i\right)^2 + \lambda_i^{kT}(x_i - w^k) \\ &+ \frac{\rho}{2}\left\|x_i - w^k\right\|_2^2 \quad i = 1, 2, \ldots, S \end{aligned} \tag{4.2.65}$$

The y-minimization step (4.2.50) is

$$\begin{aligned} w^{k+1} &= \arg\min_{w} L_{\rho}\left(x_1^{k+1}, x_2^{k+1}, \ldots, x_S^{k+1}, w, \lambda^k\right) \\ &= \arg\min_{w} \sum_{i=1}^{S} \left[\lambda_i^{kT}(x_i^{k+1} - w) + \frac{\rho}{2}\left\|x_i^{k+1} - w\right\|_2^2\right] \\ &= \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1} + \frac{1}{S\rho}\sum_{i=1}^{S}\lambda_i^k \end{aligned} \tag{4.2.66}$$

The dual update step (4.2.51) is

$$\lambda_i^{k+1} = \lambda_i^k + \rho\left(x_i^{k+1} - w^{k+1}\right) \qquad i = 1, 2, \ldots, S \tag{4.2.67}$$

If we carry out the summation of (4.2.67) over i and substitute w^{k+1} from (4.2.66), then

$$\sum_{i=1}^{S}\lambda_i^{k+1} = \sum_{i=1}^{S}\lambda_i^k + \rho\sum_{i=1}^{S}x_i^{k+1} - S\rho w^{k+1} = 0 \qquad k \neq 0 \tag{4.2.68}$$

which means that the dual variables have average value zero after the first iteration. If the dual variables are initialized such that

$$\sum_{i=1}^{S}\lambda_i^1 = 0 \tag{4.2.69}$$

then, the y-minimization step simplifies to

$$w^{k+1} = \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1} \triangleq \overline{x}^{k+1} \tag{4.2.70}$$

The simplified ADMM steps, in a distributed form, become

$$x_i^{k+1} \triangleq \arg\min_{x_i} [h(x_i) - a_i]^2 + \lambda_i^{kT}(x_i - \overline{x}^k) + \frac{\rho}{2} \left\| x_i - \overline{x}^k \right\|_2^2 \quad i = 1, 2, \ldots, S \tag{4.2.71}$$

$$\lambda_i^{k+1} \triangleq \lambda_i^k + \rho \left(x_i^{k+1} - \overline{x}^{k+1} \right) \quad i = 1, 2, \ldots, S \tag{4.2.72}$$

Based on the above ADMM steps, we obtain an averaging consensus based distributed algorithm as shown in Table 4.2.1. Each node v_i stores and updates two vectors x_i and λ_i. At iteration $k = 1$, each node initializes a local parameter estimate x_i^1 and obtains $\overline{x}^1$ via a distributed averaging consensus algorithm as discussed in Section 4.2.2. The dual variables $\lambda_i^1 = 0$ are also initialized. During the k-th iteration, each node updates its local parameter estimate x_i^{k+1} using (4.2.71). Next, each node reaches the consensus on $\overline{x}^{k+1}$, and subsequently, updates its local dual variable λ_i^{k+1} using (4.2.72), which concludes the k-th iteration.

Reformulations of (4.2.52) other than (4.2.53) include [50] and [42], which result in a bridge-sensor based distributed ADMM and a coloring-scheme based distributed ADMM, respectively. However, either prior assignment of bridge sensors [50] or colors [42] is required for the respective algorithm to function properly. However, in these versions it is difficult to make a new assignment in case of node or link failures. Whereas, the averaging consensus based distributed ADMM algorithm does not require any feature assignment to individual nodes since it relies solely on information diffusion across the network.

4.3 Problem Statement and Formulation

4.3.1 Problem Statement

Consider a scenario where there are N stationary targets located in $\mathbb{R}^2$. The target locations are denoted as

$$\mathbf{T} = (\mathrm{T}_1,\, \mathrm{T}_2,\, \ldots,\, \mathrm{T}_N) = \left(\begin{bmatrix} T_{x_1} \\ T_{y_1} \end{bmatrix},\, \begin{bmatrix} T_{x_2} \\ T_{y_2} \end{bmatrix},\, \ldots,\, \begin{bmatrix} T_{x_N} \\ T_{y_N} \end{bmatrix} \right) \tag{4.3.1}$$

The number of targets and their locations are unknown quantities of interest, to be estimated. A wireless sensor network consisting of S stationary nodes is deployed at known locations

$$\mathbf{S} = (\mathrm{S}_1,\, \mathrm{S}_2,\, \ldots,\, \mathrm{S}_S) = \left(\begin{bmatrix} S_{x_1} \\ S_{y_1} \end{bmatrix},\, \begin{bmatrix} S_{x_2} \\ S_{y_2} \end{bmatrix},\, \ldots,\, \begin{bmatrix} S_{x_S} \\ S_{y_S} \end{bmatrix} \right) \tag{4.3.2}$$

to perform this estimation task. There is one transient event occurring at each target location. Each node is able to observe these transient events by detecting the acoustic signals arising from them and measure the bearings to the targets and the TOAs of the received acoustic signals. The acoustic signal emission times are denoted as

$$\mathbf{t}^{\mathrm{e}} = (t_1^{\mathrm{e}},\, t_2^{\mathrm{e}},\, \ldots,\, t_N^{\mathrm{e}}) \tag{4.3.3}$$

For notational simplicity, let us denote

$$\mathbf{\Phi} = \left[\boldsymbol{\phi}_1^T\, \boldsymbol{\phi}_2^T \ldots\, \boldsymbol{\phi}_N^T \right]^T \tag{4.3.4}$$

where

$$\boldsymbol{\phi}_i = \left[T_{x_i} \; T_{y_i} \; t_i^{\mathrm{e}} \right]^T \tag{4.3.5}$$

denotes the unknown 3-dimensional parameter of i-th target.

If the transient events are separated significantly in time, the measurements from the same event will be close in time and the measurements from different events will also be separated significantly in time, and then the target locations can be estimated one at a time using the algorithm presented in 4.2.7. Therefore, we assume a more challenging situation that the transient events are close in time. In this case, the data association between the measurements and the targets has to be addressed before the network can fuse the measurements from a common origin to estimate the corresponding target location.

It is assumed that all measurements fall within a short time window W. Let m_ℓ denote the number of measurements (one measurement is defined as a vector consisting of both a bearing and a TOA due to one acoustic signal in this context) obtained by the ℓ-th sensor within the time window W. The j-th measurement received by the ℓ-th sensor, if it is from the i-th target at t_i^{e}, is

$$\begin{aligned} \mathbf{z}_{\ell j} &= \mathbf{h}_\ell \left(\boldsymbol{\phi}_i \right) + \mathbf{w}_{\ell j} \quad i = 1, \ldots, N; \\ \ell &= 1, \ldots, S; \quad j = 1, \ldots, m_\ell \end{aligned} \tag{4.3.6}$$

where $\mathbf{w}_{\ell j}$ is a zero mean white Gaussian measurement noise with a known diagonal co-

variance matrix R_ℓ and

$$\mathbf{h}_\ell\left(\boldsymbol{\phi}_i\right) = \begin{bmatrix} \theta_{\ell i} \\ t_{\ell i} \end{bmatrix} = \begin{bmatrix} \arctan\left[\dfrac{T_{y_i} - S_{y_\ell}}{T_{x_i} - S_{x_\ell}}\right] \\ t_i^{\mathrm{e}} + \dfrac{\sqrt{(T_{x_i} - S_{x_\ell})^2 + (T_{y_i} - S_{y_\ell})^2}}{c} \end{bmatrix} \tag{4.3.7}$$

where t_i^{e} is the unknown emission time of the acoustic signal from i-th target and c is the known speed of sound.

To incorporate false alarms, we denote a clutter target (with index 0) as $\boldsymbol{\phi}_0$. A false measurement detected by the ℓ-th sensor consists of a bearing θ_0, which is uniformly distributed in the field of view of the ℓ-th sensor, and its arrival time t_0, which is uniformly distributed in the interval $[0, W]$. The number of false alarms from each sensor is assumed to be a Poisson random variable with mean

$$N_{\mathrm{fa}} = \lambda_{\mathrm{fa}} \Phi W \tag{4.3.8}$$

where Φ is the range of the field of view and is assumed to be the same for each sensor and λ_{fa} can be interpreted as the spatial-temporal density.

The likelihood function [4] of the target parameter[4] (location and emission time) based on the measurement $\mathbf{z}_{\ell j}$ is

$$\Lambda(\boldsymbol{\phi}_0; \mathbf{z}_{\ell j}) \triangleq p(\mathbf{z}_{\ell j}|\boldsymbol{\phi}_0) = p(\theta_0)p(t_0) = \frac{1}{\Phi W} \tag{4.3.9}$$

$$\begin{aligned} &\Lambda(\boldsymbol{\phi}_i; \mathbf{z}_{\ell j}) \triangleq p(\mathbf{z}_{\ell j}|\boldsymbol{\phi}_i) = |2\pi R_\ell|^{-\frac{1}{2}} \\ &\cdot \exp\left\{ -\frac{1}{2}\left[\mathbf{z}_{\ell j} - \mathbf{h}_\ell(\boldsymbol{\phi}_i)\right]' R_\ell^{-1}\left[\mathbf{z}_{\ell j} - \mathbf{h}_\ell(\boldsymbol{\phi}_i)\right] \right\} \quad i \neq 0 \end{aligned} \tag{4.3.10}$$

[4] If the source is clutter, it has no emission time, only an arrival time.

where (4.3.9) is the probability density function (pdf) of a clutter-origin measurement (a false alarm), and (4.3.10) is the pdf of a real measurement from a true target with unknown $\boldsymbol{\phi}_i$.

The problem is to estimate N and $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_i,\, i = 1, \ldots, N\}$ (therefore knowing $\mathbf{T} = \{\mathrm{T}_i,\, i = 1, \ldots, N\}$) given the complete set of observations $\mathbf{Z} = \{\mathbf{z}_{\ell j}, \ell = 1, \ldots, S; j = 1, \ldots,\, m_\ell\}$ in the presence of missed detections and false alarms and without the knowledge of the true data association.

4.3.2 Poisson Point Process Measurement Modeling

Assume the number of targets, N, is given. The number of measurements m_ℓ and $\{\mathbf{z}_{\ell j}, j = 1, 2, \ldots, m_\ell\}$ obtained at the ℓ-th sensor is jointly modeled as a realization of a Poisson Point Process (PPP) [17]. The measurement set at the ℓ-th sensor is denoted as

$$\boldsymbol{\psi}_\ell = \{m_\ell,\, \mathbf{z}_{\ell 1},\, \mathbf{z}_{\ell 2},\, \ldots,\, \mathbf{z}_{\ell m_\ell}\} \tag{4.3.11}$$

In this case, the points $\mathbf{z}_{\ell j}$ occur in the space $\mathbb{S} = \{(\theta, t) : \theta \in \Phi, t \in [0, W]\}$ and their order is irrelevant. The PPP is fully parameterized by its spatial intensity function

$$\mu_\ell(\mathbf{z}) = \sum_{i=0}^{N} p_i^{\mathrm{d}}\, g_{\ell i}(\mathbf{z}) \tag{4.3.12}$$

where p_i^{d} is the probability of detection for the real target i ($i \neq 0$) and is assumed to be the same at each sensor and with abuse of notation

$$p_0^{\mathrm{d}} = N_{\mathrm{fa}} \tag{4.3.13}$$

is the expected number of false alarms at each sensor; the density $g_{\ell i}(\mathbf{z})$ is the conditional[5] pdf of a measurement $\mathbf{z}$ obtained by the ℓ-th sensor given that it is associated with the i-th target and is given by

$$g_{\ell i}(\mathbf{z}) = \frac{1}{\Phi W} \qquad i = 0 \tag{4.3.14}$$

$$g_{\ell i}(\mathbf{z}) = \mathcal{N}\left(\mathbf{z};\, \mathbf{h}_\ell(\boldsymbol{\phi}_i),\, R_\ell\right) \qquad i = 1, \ldots, N \tag{4.3.15}$$

For notational simplicity, we denote

$$\mathbf{p}^{\mathrm{d}} = \left[p_0^{\mathrm{d}}\, p_1^{\mathrm{d}} \ldots\, p_N^{\mathrm{d}}\right]^T \tag{4.3.16}$$

which is assumed to be unknown and therefore the set of parameters to be estimated is expanded to $\boldsymbol{\theta} = \left[\boldsymbol{\Phi}^T\, \mathbf{p}^{\mathrm{d}T}\right]^T$ for a given N.

The number of points in the PPP is a Poisson random variable with mean $\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}$, that is, the probability mass function (pmf) of m_ℓ is

$$\begin{aligned} p(m_\ell) &= \frac{\left(\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}\right)^{m_\ell}}{m_\ell\,!} \exp\left\{ -\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z} \right\} \\ &= \frac{\left(\sum_{i=0}^{N} p_i^{\mathrm{d}}\right)^{m_\ell}}{m_\ell\,!} \exp\left(-\sum_{i=0}^{N} p_i^{\mathrm{d}}\right) \end{aligned} \tag{4.3.17}$$

The m_ℓ points are defined as independent and identically distributed (i.i.d.) samples of a

[5] $g_{\ell i}(\mathbf{z} \mid \boldsymbol{\phi}_i)$ will be used when the conditioning needs to be explicitly indicated.

random variable with probability density function

$$p(\mathbf{z}) = \frac{\mu_\ell(\mathbf{z})}{\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}} = \frac{\sum_{i=0}^{N} p_i^{\mathrm{d}}\, g_{\ell i}(\mathbf{z})}{\sum_{i=0}^{N} p_i^{\mathrm{d}}} \tag{4.3.18}$$

The joint pmf-pdf of $\boldsymbol{\psi}_\ell$ from (4.3.11) is

$$p\left(\boldsymbol{\psi}_\ell \,|\, \boldsymbol{\theta}\right) = \exp\left(-\sum_{i=0}^{N} p_i^{\mathrm{d}}\right) \prod_{j=1}^{m_\ell} \mu_\ell\left(\mathbf{z}_{\ell j} \,|\, \boldsymbol{\theta}\right) \tag{4.3.19}$$

where the conditioning (dependency) on $\boldsymbol{\theta}$ will be explicitly indicated hereafter. The factorial term $m_\ell\,!$ in (4.3.17) is canceled out because there are $m_\ell\,!$ permutations of an ordered list of measurements. Let $\boldsymbol{\Psi}$ denote the set of all measurement sets (from the S sensors), i.e.,

$$\boldsymbol{\Psi} = \left\{\boldsymbol{\psi}_1,\, \boldsymbol{\psi}_2,\, \ldots,\, \boldsymbol{\psi}_S\right\} \tag{4.3.20}$$

The conditional independence of the S measurement sets yields

$$p\left(\boldsymbol{\Psi} \,|\, \boldsymbol{\theta}\right) = \prod_{\ell=1}^{S} p\left(\boldsymbol{\psi}_\ell \,|\, \boldsymbol{\theta}\right) \tag{4.3.21}$$

Therefore, we can find the maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ by maximizing (4.3.21).

4.3.3 Data Association Modeling

Since the intensity function (4.3.12) is a mixture of uniform or Gaussian pdf and the association is unknown, we model the latent association variables as conditionally independent random variables

$$\kappa_{\ell j} \in \{0, 1, 2, \ldots, N\} \tag{4.3.22}$$

that identify which component generated $\mathbf{z}_{\ell j}$. Here $\kappa_{\ell j} = 0$ indicates that the measurement is generated by the clutter. The set of latent variables for the ℓ-th sensor is denoted as

$$\kappa_{\ell} = \{\kappa_{\ell 1}, \ldots, \kappa_{\ell m_{\ell}}\} \tag{4.3.23}$$

such that the complete set of latent variables for all sensors is

$$\boldsymbol{\kappa} = \{\kappa_1, \ldots, \kappa_S\} \tag{4.3.24}$$

The latent association variables may be regarded as "marks" associated with each of the points in the PPP. If we define a mark space

$$\mathbb{M} \triangleq \{0, 1, 2, \ldots, N\} \tag{4.3.25}$$

then the marked measurement set at the ℓ-th sensor denoted by

$$\boldsymbol{\psi}_{\ell}^{\mathrm{M}} = \{m_{\ell}, (\mathbf{z}_{\ell 1}, \kappa_{\ell 1}), \ldots, (\mathbf{z}_{\ell m_{\ell}}, \kappa_{\ell m_{\ell}})\} \tag{4.3.26}$$

represents a realization of the marked[6] PPP for the ℓ-th sensor on the product space $\mathbb{S} \times \mathbb{M}$. It can be shown that the intensity function of $\boldsymbol{\psi}_\ell^{\mathrm{M}}$ is

$$\mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa \,|\, \boldsymbol{\theta}) = p_\kappa^{\mathrm{d}}\, g_{\ell\kappa}(\mathbf{z}) \tag{4.3.27}$$

The joint probability density function of $\boldsymbol{\psi}_\ell^{\mathrm{M}}$ is, similarly to (4.3.19), given by

$$\begin{aligned} p\left(\boldsymbol{\psi}_\ell^{\mathrm{M}} \,|\, \boldsymbol{\theta}\right) = \exp\left(-\sum_{\kappa=0}^{N} \int_{\mathbb{S}} \mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa \,|\, \boldsymbol{\theta})\, d\mathbf{z}\right) \\ \cdot \prod_{j=1}^{m_\ell} \mu_\ell^{\mathrm{M}}(\mathbf{z}_{\ell j}, \kappa_{\ell j} \,|\, \boldsymbol{\theta}) \end{aligned} \tag{4.3.28}$$

Let us denote the marked measurement sets from all sensors as

$$\boldsymbol{\Psi}^{\mathrm{M}} = \left\{\boldsymbol{\psi}_1^{\mathrm{M}},\, \boldsymbol{\psi}_2^{\mathrm{M}},\, \ldots,\, \boldsymbol{\psi}_S^{\mathrm{M}}\right\} \tag{4.3.29}$$

The conditional independence of these S marked measurement sets yields the pmf-pdf for $\boldsymbol{\Psi}^{\mathrm{M}}$ as

$$p\left(\boldsymbol{\Psi}^{\mathrm{M}} \,|\, \boldsymbol{\theta}\right) = \exp\left(-S \sum_{i=0}^{N} p_i^{\mathrm{d}}\right) \prod_{\ell=1}^{S} \prod_{j=1}^{m_\ell} p_{\kappa_{\ell j}}^{\mathrm{d}}\, g_{\ell\kappa_{\ell j}}\left(\mathbf{z}_{\ell j} \,|\, \boldsymbol{\theta}\right) \tag{4.3.30}$$

where we have used the fact

$$\sum_{\kappa=0}^{N} \left(\int_{\mathbb{S}} p_\kappa^{\mathrm{d}}\, g_{\ell\kappa}(\mathbf{z}|\mathbf{T}, \mathbf{t}^{\mathrm{e}})\, d\mathbf{z}\right) = \sum_{i=0}^{N} p_i^{\mathrm{d}} \tag{4.3.31}$$

[6]The superscript of $\boldsymbol{\psi}_\ell^{\mathrm{M}}$ indicates that the associations are known, i.e., "marked".

TABLE 4.4.1: Centralized EM algorithm.

1: Initializes $\boldsymbol{\theta}^{(0)}$
2: **for** $n = 1, 2, ...$ **do** until convergence
3: **E step** Evaluate

$$Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}^{(n-1)}) = \sum_{\boldsymbol{\kappa}} p(\boldsymbol{\kappa} \,|\mathbf{Z},\, \boldsymbol{\theta}^{(n-1)}) \ln p(\boldsymbol{\Psi}^{\mathrm{M}} \,|\, \boldsymbol{\theta}) \tag{4.4.1}$$

4: **M step** Evaluate $\boldsymbol{\theta}^{(n)}$ as

$$\boldsymbol{\theta}^{(n)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}^{(n-1)}) \tag{4.4.2}$$

5: **end for**

Dividing (4.3.30) by (4.3.21) leads to the density of the marks conditioned on the observed measurements and the unknown parameters

$$p(\boldsymbol{\kappa} \,|\, \mathbf{Z}, \boldsymbol{\theta}) = \prod_{\ell=1}^{S} \prod_{j=1}^{m_\ell} p_\ell \left(\kappa_{\ell j} \,|\, \mathbf{z}_{\ell j}, \boldsymbol{\theta}\right) \tag{4.3.32}$$

where

$$p_\ell \left(\kappa_{\ell j} \,|\, \mathbf{z}_{\ell j}, \boldsymbol{\theta}\right) = \frac{p^{\mathrm{d}}_{\kappa_{\ell j}}\, g_{\ell \kappa_{\ell j}} \left(\mathbf{z}_{\ell j} \,|\, \boldsymbol{\theta}\right)}{\mu_\ell(\mathbf{z}_{\ell j} \,|\, \boldsymbol{\theta})} \tag{4.3.33}$$

4.4 Centralized Algorithm

4.4.1 Centralized EM Algorithm

Given the joint distribution $p\left(\boldsymbol{\Psi}^{\mathrm{M}} \,|\, \boldsymbol{\theta}\right)$ over observed $\boldsymbol{\Psi}$ and latent variables $\boldsymbol{\kappa}$, governed by the parameter $\boldsymbol{\theta}$, the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ from the likelihood function $p\left(\boldsymbol{\Psi} \,|\, \boldsymbol{\theta}\right)$ can be found by the standard (named as centralized hereafter) EM algorithm [19] as shown in Table 4.4.1.

Evaluation of the E step decomposes into two terms

$$Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}^{(n-1)}) = Q_p + Q_\phi \tag{4.4.3}$$

where

$$\begin{aligned} Q_p = Q(\mathbf{p}^{\mathrm{d}} \,|\, \boldsymbol{\theta}^{(n-1)}) &= -S \sum_{i=0}^{N} p_i^{\mathrm{d}} \\ &+ \sum_{\ell=1}^{S} \sum_{j=1}^{m_\ell} \sum_{i=0}^{N} \ln\left(p_i^{\mathrm{d}}\right) \alpha_{\ell j i}^{(n-1)} \end{aligned} \tag{4.4.4}$$

$$Q_\phi = Q(\boldsymbol{\Phi} \,|\, \boldsymbol{\theta}^{(n-1)}) = \sum_{\ell=1}^{S} \sum_{j=1}^{m_\ell} \sum_{i=0}^{N} \ln\left(g_{\ell i}(\mathbf{z}_{\ell j} \,|\, \boldsymbol{\phi}_i)\right) \alpha_{\ell j i}^{(n-1)} \tag{4.4.5}$$

where

$$\begin{aligned} \alpha_{\ell j i}^{(n-1)} &= p_\ell\left(\kappa_{\ell j} = i \,|\, \mathbf{z}_{\ell j}, \boldsymbol{\theta}^{(n-1)}\right) \\ &= \frac{p_i^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \,|\, \boldsymbol{\phi}_i^{(n-1)})}{\sum_{i=0}^{N} p_i^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \,|\, \boldsymbol{\phi}_i^{(n-1)})} \end{aligned} \tag{4.4.6}$$

The M step involves two separate maximizations with respect to $\mathbf{p}^{\mathrm{d}}$ and $\boldsymbol{\Phi}$. From the Karush-Kuhn-Tucker (KKT) conditions [36], we have

$$p_i^{\mathrm{d}(n)} = \begin{cases} \frac{1}{S} \sum_{\ell=1}^{S} \sum_{j=1}^{m_\ell} \alpha_{\ell j 0}^{(n-1)} & \text{if } i = 0 \\ \min\left\{1, \frac{1}{S} \sum_{\ell=1}^{S} \sum_{j=1}^{m_\ell} \alpha_{\ell j i}^{(n-1)}\right\} & \text{if } i \neq 0 \end{cases} \tag{4.4.7}$$

Since Q_ϕ in (4.4.5) can be further decomposed into $N+1$ terms, the parameters of each

target can be estimated independently as

$$\boldsymbol{\phi}_i^{(n)} = \arg\max_{\boldsymbol{\phi}_i} \sum_{\ell=1}^{S} \sum_{j=1}^{m_\ell} \ln\left(g_{\ell i}\left(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i\right)\right) \alpha_{\ell j i}^{(n-1)} \tag{4.4.8}$$

4.5 Distributed Algorithm

Note that it is possible to have a distributed implementation of the centralized EM algorithm if (i) every node has the consensus on the initialization and (ii) every node has the consensus on the parameter estimates at the end of each M step. The second condition can readily be satisfied if an averaging consensus based distributed ADMM is applied to solve (4.4.8), which is a nonlinear least squares problem, and a distributed averaging consensus algorithm is applied to obtain (4.4.7). However, it is not trivial to have the same initialization for $\boldsymbol{\theta} = \left[\boldsymbol{\Phi}^T \, \mathbf{p}^{\mathrm{d}T}\right]^T$ among all sensors, especially for the component $\boldsymbol{\Phi}$. Simulation results show that it is good enough to initialize each p_i^{d} to be 1. Whereas, equal initialization at some pre-fixed values (for instance, zero vectors) for $\boldsymbol{\Phi}$ could result in the convergence of the EM algorithm to estimates that are very different from the desired MLE.

There are two possible initialization approaches in a single target localization scenario. Assume that the data association is known and no missed detections or false alarms occur, we want to localize a single target using the algorithm in Table 4.2.1. The first approach is to initialize the target location at each node using only its bearing measurement. The average distance from the wireless sensor network to the target is assumed to be D. Given a range R (probably unknown in a real scenario), each node initializes the target location along the measured line of sight in the direction towards the target randomly with a distance (between the initialized target location and the node itself) being uniformly distributed in

TABLE 4.5.1: RMSE using different initializations for distributed localization of a single target.

	Centralized	Intersection	Random $R = 30$	Random $R = 60$
RMSE (m)	1.7036	1.7733	3.2988	5.2915

[D-0.5R, D+0.5R]. The second approach is to obtain the LOS information (bearing and sensor location) from one of its neighbors and use the intersection of two lines of sight as the initial target location estimate.

Table 4.5.1 lists the root mean square error (RMSE) of the target location (averaged over 100 Monte-Carlo runs) using different initialization approaches to localize a single target at the location $(8.7\,\mathrm{m},\ 99.6\,\mathrm{m})$ in a scenario given in Section 4.6. It shows that the performance of the distributed algorithm with LOS intersection initialization is almost as good as the centralized algorithm, which assumes all bearing measurements available at a fusion center and also uses intersection initialization. With a random initialization based on some knowledge, which is likely unavailable, the distributed algorithm converges to local minimum point of (4.2.53) with $h(x)$ being an $\arctan$ function.

For a multiple target localization scenario with unknown data association, the random initialization approach will be worse. Therefore, in a similar way as what we did at the fusion center in a centralized fusion algorithm, each node obtains an initial position estimate for each target that is very close to the final MLE, by associating its local measurements with those from its neighbors using the sequential m-best 2-D assignment algorithm [3]. Another important reason that we choose the sequential m-best 2-D algorithm over the random initialization approach is that the position estimates obtained are completely observable with corresponding covariance matrices, which allows the use of the association method described in Section 4.2.5 to reach the consensus on the initialization.

If the probability of detection is low or the false alarm rate is high, then it is possible that

the initial estimated numbers of targets at various nodes are different. Some nodes could have estimated more targets than there actually are due to false alarms, whereas other nodes could have estimated less due to missed detections. Section 4.5.1 discusses how to reach the consensus on the number of candidate targets[7] among all nodes.

Section 4.5.2 assumes that each node has an initial set consisting of the same number of target estimates which correspond to the same group of candidate targets, and discusses how to reach the consensus on on target-estimate association, that is, for a given ordered set of candidate targets, each node should know the association between its estimates and the targets. Note that the initial estimated value of parameter $\mathbf{\Phi}$ could still be different from node to node. However, the consensus on target-estimate association requires all the nodes have exactly the same set of target estimates.

In Section 4.5.3, we develop a distributed algorithm that assumes all the nodes have the same set of target estimates. The consensus on target-estimate association is required for convergence of the algorithm.

4.5.1 Consensus on the Number of Targets

If the probability of detection is low or the false alarm rate is high, then it is possible that the initial estimated numbers of targets at various nodes are different. Some nodes could have estimated more targets than there should be, whereas other nodes could have estimated less. In this subsection, we extend the problem solved in Section 4.2.3 to the case when missed detections and/or false alarms exist and develop a distributed set averaging consensus algorithm to expand some or all sets so that we end up with sets of estimates for the same number of candidate (real or false) targets. Each sensor gradually modifies

[7]The concept of candidate target is discussed in detail in Section 4.5.1.

its own set by performing the association test presented in Section 4.2.5 with the sets of its neighbors.

Let us denote the initial set of estimates with corresponding covariance matrices at node v_i as

$$\hat{\mathbf{\Phi}}_i = \{\hat{\boldsymbol{\phi}}_{i1}, \hat{\boldsymbol{\phi}}_{i2}, \ldots, \hat{\boldsymbol{\phi}}_{iN_i}\} \tag{4.5.1}$$

$$\mathcal{Q}_i = \{Q_{i1}, Q_{i2}, \ldots, Q_{iN_i}\} \tag{4.5.2}$$

where each $\hat{\boldsymbol{\phi}}_{ik}$ corresponds to one candidate target and the number of candidate targets N_i is probably distinct for different nodes v_i. Assume that there are N_c candidate targets with parameters

$$\mathbf{\Phi}_c = \{\boldsymbol{\phi}_1,\ \boldsymbol{\phi}_2,\ \ldots,\ \boldsymbol{\phi}_{N_c}\} \tag{4.5.3}$$

of which only N parameters correspond to real targets and the remaining $N_c - N$ parameters correspond to false targets. The number N_c will only be known at the end of the algorithm. For any target parameter in $\mathbf{\Phi}_c$, there is at most one estimate $\hat{\boldsymbol{\phi}}_{ik}$ at node v_i.

Figure 4.5.1 illustrates the concept of candidate target. In this example, both sensors have detected true targets at coordinates $(10\,\text{m},\ 10\,\text{m})$ and $(20\,\text{m},\ 20\,\text{m})$, therefore each of these is a candidate target. Sensors 1 and 2 each also have an additional estimated target around $(31\,\text{m},\ 31\,\text{m})$ and $(29\,\text{m},\ 11\,\text{m})$, respectively. In this case, we assume that these two estimates fail to be associated. Therefore, two additional targets, which are assumed to be at coordinates $(30\,\text{m},\ 30\,\text{m})$ and $(30\,\text{m},\ 10\,\text{m})$, are also candidate targets.

Referring back to the same context in Section 4.2.5, the sets $\mathbf{\Phi}_c$ and $\hat{\mathbf{\Phi}}_i$ defined in (4.5.3) and (4.5.1) play the same roles as X and $\hat{X}_i$ in (4.2.33) and (4.2.34), respectively. The in-

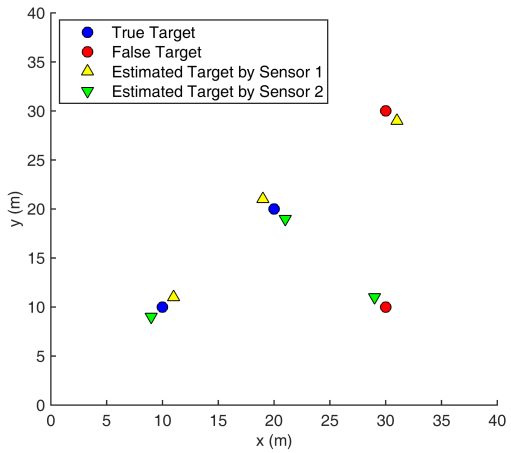


FIGURE 4.5.1: An illustrative example: each sensor has three estimates, there are four candidate targets.

dependent estimation error assumption is valid only when two estimates have no common source of error [5]. In the case that two neighboring nodes each have one estimate for the same target, it is quite likely that these two estimates are obtained using some common measurements and therefore they have correlated errors. Since it is difficult to calculate the cross covariance, we will use (4.2.26) as an approximation. This approximation only applies to true target location estimates that are supposed to be associated, and will not affect the decisions involving estimates that belong to false targets. In our approach, the independence assumption (of the errors in the estimated target locations at different sensors) is used only to build the consensus. We do not "fuse" the corresponding covariances, which pertain to errors that are dependent; fusing them under independence assumption would indeed be optimistic and unreasonable.

One iteration of the distributed set averaging consensus algorithm is described next. At iteration t, node v_i expands $\hat{\mathbf{\Phi}}_i(t)$ sequentially with each neighboring node $v_j \in \mathcal{N}_i$.

Firstly, for a given significance level α, the following generalized 2-D assignment prob-

lem

$$\min_{\rho_{k\ell}} \sum_{k=0}^{N_i} \sum_{\ell=0}^{N_j} \rho_{k\ell}\, d_{k\ell}(t) \tag{4.5.4}$$

subject to

$$\sum_{\ell=0}^{N_j} \rho_{k\ell} = 1 \text{ for all } k = 1, 2, \ldots, N_i \tag{4.5.5}$$

$$\sum_{k=0}^{N_i} \rho_{k\ell} = 1 \text{ for all } \ell = 1, 2, \ldots, N_j \tag{4.5.6}$$

$$\rho_{k\ell} \in \{0, 1\} \text{ for all } k = 0, 1, \ldots, N_i \text{ and } \ell = 0, 1, \ldots, N_j \tag{4.5.7}$$

where, similarly as in (4.2.38) and (4.2.39), with the addition of a dummy estimate $\hat{\boldsymbol{\phi}}_{i0}$ at each node υ_i, the distance between two estimates are defined as

$$d_{k\ell}(t) = \begin{cases} \left(\hat{\boldsymbol{\phi}}_{ik}(t) - \hat{\boldsymbol{\phi}}_{j\ell}(t)\right)^T \left(P_{ik}(t) + P_{j\ell}(t)\right)^{-1} \\ \cdot \left(\hat{\boldsymbol{\phi}}_{ik}(t) - \hat{\boldsymbol{\phi}}_{j\ell}(t)\right) \quad \text{if } k > 0 \text{ and } \ell > 0 \\ F^{-1}_{\chi^2_n}(1-\alpha) \quad \text{if } k = 0 \text{ or } \ell = 0 \end{cases} \tag{4.5.8}$$

Next, $\hat{\boldsymbol{\Phi}}_i(t)$ could be expanded based on the solution $\rho_{k\ell}$ to the assignment problem. If

$$\rho_{0\ell} = 1 \tag{4.5.9}$$

which means that the estimate $\hat{\boldsymbol{\phi}}_{j\ell}(t)$ is not associated with any estimate at node υ_i, then $\hat{\boldsymbol{\Phi}}_i(t)$ is expanded to $\hat{\boldsymbol{\Phi}}_i(t) \cup \{\hat{\boldsymbol{\phi}}_{j\ell}(t)\}$. If there are multiple estimates that are not associated, then they are all used to expand $\hat{\boldsymbol{\Phi}}_i(t)$.

The algorithm terminates when every node set has the same number of estimates and

no set can be expanded further.

Note that if a sensor does not have a position estimate for target i, it will "copy" a position estimate for target i from one of its neighbors. If a sensor has a position estimate for a false target, then all its neighbors need to "copy" this estimate so that every sensor has a position estimate for the same false target. Since the total number of target estimates across all the sensors is a finite number $\sum_{i=1}^{S} N_i$, where S is the number of sensors, and each iteration expands at least one set, the algorithm will be terminated in a finite number of iterations.

4.5.2 Consensus on the Target-Estimate Association

Suppose that the initial sets of target estimates, either obtained directly via the assignment algorithm across all nodes or by means of the method described in Section 4.5.1, have the same number of target estimates.

The local variable of $\boldsymbol{\theta}_\ell$ has components $\boldsymbol{\Phi}_\ell$ and $\mathbf{p}_\ell^{\mathrm{d}}$. We initialize $\mathbf{p}_\ell^{\mathrm{d}}$ as a vector of ones. The component $\boldsymbol{\Phi}_\ell = \begin{bmatrix} \boldsymbol{\phi}_{\ell 1}^T \; \boldsymbol{\phi}_{\ell 2}^T \; \ldots \; \boldsymbol{\phi}_{\ell N}^T \end{bmatrix}^T$ will be initialized using the set obtained via the sequential assignment algorithm denoted by

$$\Phi_\ell = \{\varphi_{\ell 1},\, \varphi_{\ell 2},\, \ldots,\, \varphi_{\ell N}\} \tag{4.5.10}$$

There are $N\,!$ ways of initialization for node ℓ. We want to find a permutation for each set Φ_ℓ

$$\pi_\ell(\Phi_\ell) = \{\varphi_{\ell\pi_\ell(1)},\, \varphi_{\ell\pi_\ell(2)}, \ldots,\, \varphi_{\ell\pi_\ell(N)}\} \tag{4.5.11}$$

such that for any $k = 1,\, 2\, \ldots, N$, the set of estimates $\{\varphi_{\ell\pi_\ell(k)},\ \ell = 1, 2, \ldots, S\}$, one from

each sensor, corresponds to the same target. For this purpose, we can apply the algorithm in Section 4.2.3 on the sets $\Phi_\ell,\ \ell = 1, \ldots, S$, and when the algorithm terminates, we have all the sets equal.

We initialize $\mathbf{\Phi}_\ell$ as

$$\mathbf{\Phi}_\ell = \left[\varphi^T_{\ell\pi_\ell(1)}\ \varphi^T_{\ell\pi_\ell(2)} \cdots\ \varphi^T_{\ell\pi_\ell(N)}\right]^T \tag{4.5.12}$$

where, letting $i = \pi_\ell(k)$

$$\varphi_{\ell i} = \left[T_{x_i}\ T_{y_i}\ t_i^{\mathrm{e}}\right]^T \tag{4.5.13}$$

is ordered such that

$$T_{x_i} \leq T_{x_j},\ \forall\ i \leq j \tag{4.5.14}$$

$$T_{y_i} \leq T_{y_j},\ \forall\ i \leq j \text{ and } T_{x_i} = T_{x_j} \tag{4.5.15}$$

The use of the ordering rules (4.5.14) and (4.5.15) to label targets makes sense only when the sets of the estimates from all nodes are the same.

4.5.3 The EM and AC Based Distributed ADMM Algorithm

The centralized EM algorithm provides a method to solve the following optimization problem

$$\min_{\boldsymbol{\theta}} \left\{-\ln p\left(\mathbf{\Psi}\middle|\boldsymbol{\theta}\right)\right\} \tag{4.5.16}$$

where

$$\ln p\left(\boldsymbol{\Psi}\middle|\boldsymbol{\theta}\right)=\sum_{\ell=1}^{S}\ln p\left(\boldsymbol{\psi}_{\ell}\middle|\boldsymbol{\theta}\right) \tag{4.5.17}$$

To develop a distributed algorithm to solve the above problem, we consider an equivalent formulation with equality constraints between local variables $\boldsymbol{\theta}_{\ell}$ and a global variable $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\ldots,\boldsymbol{\theta}_S}\quad\sum_{\ell=1}^{S}-\ln p\left(\boldsymbol{\psi}_{\ell}\middle|\boldsymbol{\theta}_{\ell}\right) \tag{4.5.18}$$

subject to

$$\boldsymbol{\theta}_1=\boldsymbol{\theta}_2=\ldots=\boldsymbol{\theta}_S=\boldsymbol{\theta} \tag{4.5.19}$$

The augmented Lagrangian is

$$\begin{aligned}L_{\rho}\left(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\ldots,\boldsymbol{\theta}_S,\boldsymbol{\theta},\lambda\right)=\sum_{\ell=1}^{S}\Big[&-\ln p\left(\boldsymbol{\psi}_{\ell}\middle|\boldsymbol{\theta}_{\ell}\right)\\&+\lambda_{\ell}^{T}\left(\boldsymbol{\theta}_{\ell}-\boldsymbol{\theta}\right)+\frac{\rho}{2}\left\|\boldsymbol{\theta}_{\ell}-\boldsymbol{\theta}\right\|_2^2\Big]\end{aligned} \tag{4.5.20}$$

Following the similar derivations as presented in Section 4.2.7, we can obtain the ADMM

TABLE 4.5.2: EM and averaging consensus based distributed ADMM algorithm.

Node υ_ℓ initializes $\boldsymbol{\theta}_\ell^1$ by a sequential m-best 2-D assignment algorithm and $\lambda_\ell^1 = 0$
Compute $\boldsymbol{\theta}^1 = \frac{1}{S}\sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^1$ by a distributed averaging consensus algorithm
for $k = 1, 2, \ldots$ **do** until convergence
    for all υ_ℓ **do**
        Compute $\boldsymbol{\theta}_\ell^{k+1}$ via (4.5.21) by a local EM algorithm
        Compute $\boldsymbol{\theta}^{k+1} = \frac{1}{S}\sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^{k+1}$ by a distributed averaging consensus algorithm
        Compute λ_ℓ^{k+1} via (4.5.23)
    end for
end for

steps, which are in a distributed form, as

$$\boldsymbol{\theta}_\ell^{k+1} = \arg\min_{z} \sum_{\ell=1}^{S} \Big[-\ln p\left(\boldsymbol{\psi}_\ell \middle| \boldsymbol{\theta}_\ell\right) + \lambda_\ell^{kT}(\boldsymbol{\theta}_\ell - \boldsymbol{\theta}^k) + \frac{\rho}{2}\left\|\boldsymbol{\theta}_\ell - \boldsymbol{\theta}^k\right\|_2^2 \Big] \tag{4.5.21}$$

$$\boldsymbol{\theta}^{k+1} = \frac{1}{S}\sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^{k+1} \tag{4.5.22}$$

$$\lambda_\ell^{k+1} = \lambda_\ell^k + \rho\left(\boldsymbol{\theta}_\ell^{k+1} - \boldsymbol{\theta}^{k+1}\right) \tag{4.5.23}$$

Based on the above ADMM steps, we obtain an EM and averaging consensus based distributed algorithm as summarized in Table 4.5.2. Each node υ_ℓ stores and updates two vectors $\boldsymbol{\theta}_\ell$ and λ_ℓ. At iteration k=1, each node initializes a local parameter estimate $\boldsymbol{\theta}_\ell^1$ and reaches the consensus on the global variable $\boldsymbol{\theta}^1$ via a distributed averaging consensus algorithm. The local dual variable is initialized as $\lambda_\ell^1 = 0$. During the k-th iteration, each node updates the local variable $\boldsymbol{\theta}_\ell^{k+1}$ via (4.5.21), which is solved by the local EM algorithm as in Table 4.5.3 because of the term $\ln p\left(\boldsymbol{\psi}_\ell \middle| \boldsymbol{\theta}_\ell\right)$. Next, each node obtains $\boldsymbol{\theta}^k$

TABLE 4.5.3: Local EM algorithm at node υ_ℓ to find $\boldsymbol{\theta}_\ell^{k+1}$.

1: Initialization

$$\boldsymbol{\theta}_\ell^{(0)} = \boldsymbol{\theta}_\ell^k \tag{4.5.24}$$

2: **for** $n = 1, 2, ...$ **do** until convergence

3: **E step**

$$\alpha_{\ell ji}^{(n-1)} = p_\ell\left(\kappa_{\ell j} = i \middle| \mathbf{z}_{\ell j}, \boldsymbol{\theta}_\ell^{(n-1)}\right) = \frac{p_{i\ell}^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell}^{(n-1)})}{\sum_{i=0}^{N} p_{i\ell}^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell}^{(n-1)})} \tag{4.5.25}$$

$$Q(\boldsymbol{\theta}_\ell|\boldsymbol{\theta}_\ell^{(n-1)}) = \sum_{\kappa_\ell} p(\kappa_\ell \big| \boldsymbol{\psi}_\ell, \boldsymbol{\theta}_\ell^{(n-1)}) \ln p(\boldsymbol{\psi}_\ell^{\mathrm{M}} \big| \boldsymbol{\theta}_\ell) = Q(\mathbf{p}_\ell^{\mathrm{d}}) + Q(\boldsymbol{\Phi}_\ell)$$

$$Q(\mathbf{p}_\ell^{\mathrm{d}}) = -\sum_{i=0}^{N} p_{i\ell}^{\mathrm{d}} + \sum_{j=1}^{m_\ell}\sum_{i=0}^{N} \ln\left(p_{i\ell}^{\mathrm{d}}\right) \alpha_{\ell ji}^{(n-1)} \tag{4.5.26}$$

$$Q(\boldsymbol{\Phi}_\ell) = \sum_{j=1}^{m_\ell}\sum_{i=0}^{N} \ln\left(g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell})\right) \alpha_{\ell ji}^{(n-1)} \tag{4.5.27}$$

4: **M step**

$$\mathbf{p}_\ell^{\mathrm{d}(n)} = \arg\min_{\mathbf{p}_\ell^{\mathrm{d}}} -Q(\mathbf{p}_\ell^{\mathrm{d}}) + \lambda_{p\ell}^{kT}(\mathbf{p}_\ell^{\mathrm{d}} - \mathbf{p}^{\mathrm{d}k}) + \frac{\rho}{2}\left\|\mathbf{p}_\ell^{\mathrm{d}} - \mathbf{p}^{\mathrm{d}k}\right\|_2^2 \tag{4.5.28}$$

$$\boldsymbol{\Phi}_\ell^{(n)} = \arg\min_{\boldsymbol{\Phi}_\ell} -Q(\boldsymbol{\Phi}_\ell) + \lambda_{\phi\ell}^{kT}(\boldsymbol{\Phi}_\ell - \boldsymbol{\Phi}^k) + \frac{\rho}{2}\left\|\boldsymbol{\Phi}_\ell - \boldsymbol{\Phi}^k\right\|_2^2 \tag{4.5.29}$$

5: **end for**

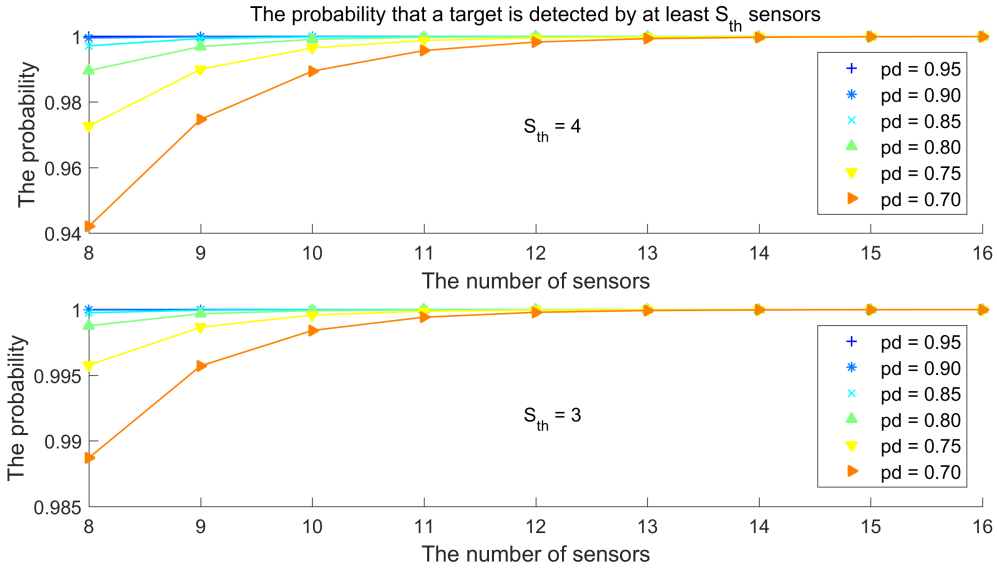


FIGURE 4.5.2: The probability that a target is detected by at least S_{th} (3 or 4) sensors for varied values for the number of sensors and the probability of detection (pd).

via a distributed averaging consensus algorithm, and subsequently, updates its local dual variable λ_i^{k+1} using (4.5.23), which concludes the k-th iteration.

In the local EM algorithm, the dual variable λ_ℓ is partitioned as

$$\lambda_\ell = \begin{bmatrix} \lambda_{\phi\ell} \\ \lambda_{p\ell} \end{bmatrix} \tag{4.5.30}$$

with respect to the components $\mathbf{\Phi}_\ell$ and $\mathbf{p}_\ell^{\text{d}}$ of $\boldsymbol{\theta}_\ell$.

4.5.4 Determination of the Number of Real Targets

The Bayesian information criterion (BIC) was used in our previous work [25] for the Poisson measurement generation model assumption because this assumption leads to a cardinality selection problem formulation, which is similar to the K-means clustering problem

and BIC is one of the widely used and trusted approaches [49] to determine the number of clusters (the number of targets in our case).

Here, we assume a Bernoulli measurement generation model, which is more realistic than the Poisson model in the multiple transient emitter localization problem. Therefore, we used the likelihood function (binomial, in view of the Bernoulli model) based thresholding approach to determine the number of targets.

In the distributed algorithm, the estimated probability of detection $\hat{p}_i^{\mathrm{d}}$ will converge to the value in (4.4.7) for $n = 10$. Assume that the true probability of detection is high (say, above 0.9) and the number of nodes is large, we expect that most of the nodes have a measurement associated with a particular target. Therefore, for a real target estimate, $\hat{p}_i^{\mathrm{d}}$ is likely to end up with a value close to 1. For a false target estimate, $\hat{p}_i^{\mathrm{d}}$ is likely to end up with a value close to 0, since only a few nodes have a measurement associated with a false target (which is the "same" across sensors, i.e., approximately at the same location). Based on this difference between real targets and false targets, it is reasonable to assume that there is a threshold value of $\hat{p}_i^{\mathrm{d}}$ that can be used to classify targets into either real or false.

If the number of sensors is known and the probability of detection is also known, then one can calculate the probability that a target is detected by at least S_{th} sensors. Figure 4.5.2 plots this probability for a range of values for the number of sensors and the probability of detection. Since even at $p^{\mathrm{d}} = 0.7$, the probability that a target is detected by at least 3 sensors is greater than 0.995 in most cases. We use the threshold value $S_{\mathrm{th}} = 3$. The corresponding threshold value of $\hat{p}_i^{\mathrm{d}}$ is

$$p_{\mathrm{th}}^{\mathrm{d}} = 0.3 \tag{4.5.31}$$

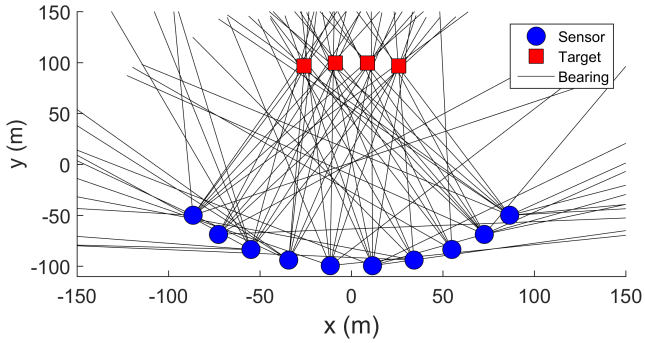


FIGURE 4.6.1: A scenario with 10 targets and 4 sensors.

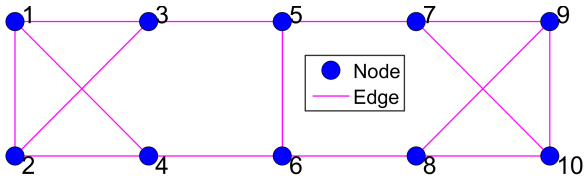


FIGURE 4.6.2: The graph model of the wireless sensor network in Figure 4.6.1.

when $S = 10$ as in the simulation study. Therefore, we classify the targets with $\hat{p}_i^{\mathrm{d}}$ greater than 0.3 as real targets and otherwise the targets are deemed as false.

4.6 Simulation Results

4.6.1 Scenario

Assume there are four targets ($N = 4$). The emission times of the acoustic events at the target locations are 0.4 s, 0.3 s, 0.1 s and 0.2 s, respectively. The speed of the acoustic signal is assumed to be 342 m/s. The measurement noise covariance matrix is

$$R_\ell = \begin{bmatrix} 7.6 \times 10^{-5} & 0 \\ 0 & 1 \times 10^{-4} \end{bmatrix} \tag{4.6.1}$$

TABLE 4.6.1: CRLB and MSE with and without TOA measurements.

	Bearing	Bearing and TOA
CRLB (m^2)	2.6655	2.6464
MSE (m^2)	2.6396	2.6290

i.e., the bearing standard deviation is $\sqrt{76}$ mrad $= 0.5^{\circ}$ and the TOA measurement standard deviation amounts to 10 ms, assumed to be the same for all sensors. The probability of detection for the targets is assumed to be 0.9 at all sensors. The time window W is chosen to be 1 s and the field of view of each sensor is from 0 to π. The density of the false alarms is set to be 1.27 s^{-1}radian^{-1} such that the expected number of false alarms (N_{fa}) at each sensor is 4, which is equal to the number of real targets. Figure 4.6.1 shows one example using a wireless sensor network with 10 sensors numbered from left to right in an ascending order, which is represented by the graph model shown in Figure 4.6.2, to localize these 4 targets. Each node has three neighbors.

In the simulation, the targets and the sensors are located such that the angle between two LOS from two neighboring targets to any sensor is 5°, which is 10 times the standard deviation of LOS measurement noise, i.e. there are no unresolved measurements.

4.6.2 The significance of TOA measurements

The TOA measurements play an important role in the data association. The ghosting effect using bearing-only measurements is no longer present due to the additional estimation of a common signal emission time for the measurements associated with a single target. Here, we look at the improved estimation accuracy provided by the TOA measurements on top of the bearing-only measurements.

Assume that the data association is known and no missed detection or false alarms

occurs, we want to localize a single target at the location $(8.7\,\mathrm{m},\ 99.6\,\mathrm{m})$ with all measurements available at a fusion center. Table 4.6.1 shows the Cramér-Rao lower bound (CRLB) and MSE of the target location using bearing-only measurements and bearing with TOA measurements. It shows that the improvement of the location estimation due to the additional TOA information is insignificant.

This implies that the TOA information should be only used in the sequential m-best assignment algorithm to obtain initial target estimates. Within the local EM algorithm, we can use only bearing measurements to reduce computational workload without significantly degrading the estimation accuracy.

4.6.3 Performance Metrics

In the following sections, we evaluate our distributed algorithm by two real-valued metrics for each Monte-Carlo run instead of averaging over all Monte-Carlo runs. These two metrics are the cardinality error for the number of targets and the root mean square (RMS) position error averaged over all targets. The latter is obtained by globally associating each location estimate to the nearest targets.

The cardinality error for the number of targets

Given the true number of targets N_{t} and the estimated number of targets $\hat{N}$, the cardinality error is defined as

$$\tilde{N} = N_{\mathrm{t}} - \hat{N} \tag{4.6.2}$$

The RMS position error

Given the set of true positions of N_{t} targets

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_{N_{\mathrm{t}}}, y_{N_{\mathrm{t}}})\} \tag{4.6.3}$$

and the set of estimated positions of $\hat{N}$ targets

$$\{(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), \ldots, (\hat{x}_{\hat{N}}, \hat{y}_{\hat{N}})\} \tag{4.6.4}$$

there are three cases. Let Π_N denote all permutations of the set $\{1, 2, \ldots, N\}$.

Case 1: $N_{\mathrm{t}} = \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS_p} = \min_{\pi \in \Pi_{N_{\mathrm{t}}}} \sqrt{\frac{1}{N_{\mathrm{t}}} \sum_{i=1}^{N_{\mathrm{t}}} \left[(x_i - \hat{x}_{\pi(i)})^2 + (y_i - \hat{y}_{\pi(i)})^2 \right]} \tag{4.6.5}$$

Case 2: $N_{\mathrm{t}} < \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS_p} = \min_{\pi \in \Pi_{\hat{N}}} \sqrt{\frac{1}{N_{\mathrm{t}}} \sum_{i=1}^{N_{\mathrm{t}}} \left[(x_i - \hat{x}_{\pi(i)})^2 + (y_i - \hat{y}_{\pi(i)})^2 \right]} \tag{4.6.6}$$

Case 3: $N_{\mathrm{t}} > \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS_p} = \min_{\pi \in \Pi_{N_{\mathrm{t}}}} \sqrt{\frac{1}{\hat{N}} \sum_{i=1}^{\hat{N}} \left[(\hat{x}_i - x_{\pi(i)})^2 + (\hat{y}_i - y_{\pi(i)})^2 \right]} \tag{4.6.7}$$

Note that we need to combine these two real-valued metrics (4.6.2 and one of 4.6.5 – 4.6.7) in order to have a complete evaluation of the algorithm performance.

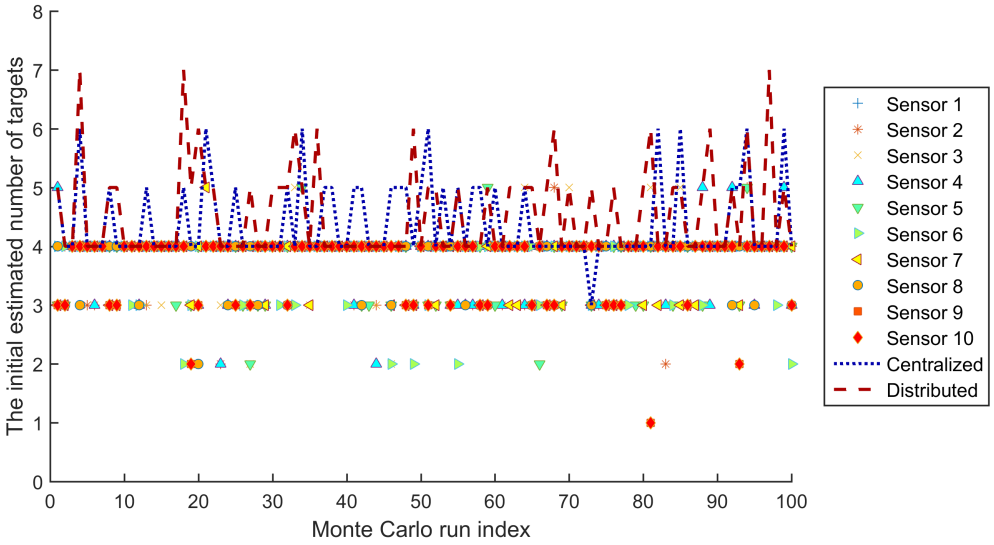


FIGURE 4.6.3: The initially estimated number (the truth is 4) of targets by individual sensors, the centralized EM algorithm and the EM and AC based distributed ADMM algorithm.

4.6.4 Performance of the EM and AC based distributed ADMM algorithm

For the algorithm evaluation, the target measurements are generated according to a Bernoulli measurement model, specifically, one measurement from each target is generated for each sensor with a probability $p_{\rm d}$ or nothing with a probability $1-p_{\rm d}$. The false alarms are generated for each sensor according to the Poisson model (4.3.8) and (4.3.9).

Note that the values of the probability of detection, $p_{\rm d}$, and the expected number of false alarms, $N_{\rm fa}$, are required to generate the target measurements. However, the EM and AC based distributed ADMM algorithm do not need to know the values of $N_{\rm fa}$ and $p_{\rm d}$. They adapt to these values by "learning them".

We used 100 Monte-Carlo runs to evaluate the performance of our distributed algorithm and make comparisons with a modified version of the centralized algorithm in [25]. Both used the same threshold (4.5.31) to determine the number of targets.

Figure 4.6.3 shows the number of targets initially estimated by each sensor using the sequential m-best 2-D assignment algorithm on the measurements of its own and its one-hop neighbors. It can be observed that this number is different from sensor to sensor because of the missed detections and false alarms, which is the motivation for the development of the distributed set consensus algorithm described in the Sections V-A and V-B. In the same plot, the centralized algorithm (denoted by "Centralized") obtained the initial estimated number of targets by using the sequential m-best 2-D assignment algorithm on the measurements from all sensors. In contrast, the distributed algorithm obtained the initial estimate (the same for all sensors) of the number of targets via the distributed set consensus algorithm and this estimate is also the estimated number of candidate targets. Since the centralized and distributed algorithms use different initialization approaches, the initially estimated number of targets is different for the two algorithms. False targets appear in 40 runs, where the estimated number of candidate targets is greater than the true number of targets.

Figure 4.6.4 shows the number of iterations required for the convergence of the different iterative algorithms presented in this chapter. All the algorithms terminate in a few iterations. The EM and AC based distributed algorithm, being itself an iterative algorithm, consists of three steps, two of which are iterative algorithms themselves (steps on Lines 5 and 6 in Table 4.5.2). By close examination, we found that average number of iterations for these two algorithms is around 3 and 9, respectively. Since local communication only occurs at the AC step (Line 6 in Table 4.5.2), the average number of communications for each sensor is approximately 50.

Figure 4.6.5 plots the number of targets estimated by the centralized and distributed algorithms before and after thresholding. Since the initialization is different for these two EM algorithms, the estimated number of targets is slightly different. In the shooter local-

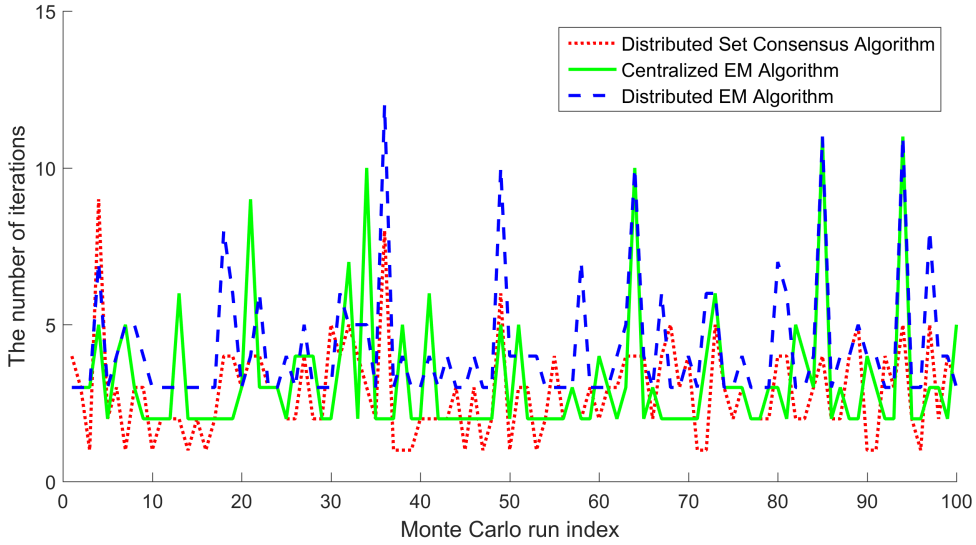


FIGURE 4.6.4: The number of iterations of the distributed set consensus algorithm, the centralized EM algorithm and the EM and AC based distributed ADMM algorithm.

ization application, the priority is to avoid any missed target and then try to avoid as many false targets as possible. There are two possible sources for the false targets in the final solution. One is that the false alarm rate is high, which can inevitably cause the presence of some false targets. The other is that a target is split into two close targets due to the association test. While the former may cause confusion in the decision making, we may prioritize the targets based on the estimated p^{d} such that the low $\hat{p}^{\mathrm{d}}$ targets have the low priority. The latter may be solved by looking at whether two close targets with low estimated probabilities of detection have their sum close to 1.

The top plot in Figure 4.6.6 shows the RMS position error (averaged over all targets) for different cases before we remove the predicted false targets. The "Known Association", which refers to the situation when we know the number of targets and the association between measurements and targets, is meant to serve as a baseline or a lower bound (which is unachievable). In this case, the position estimates can be obtained separately for each

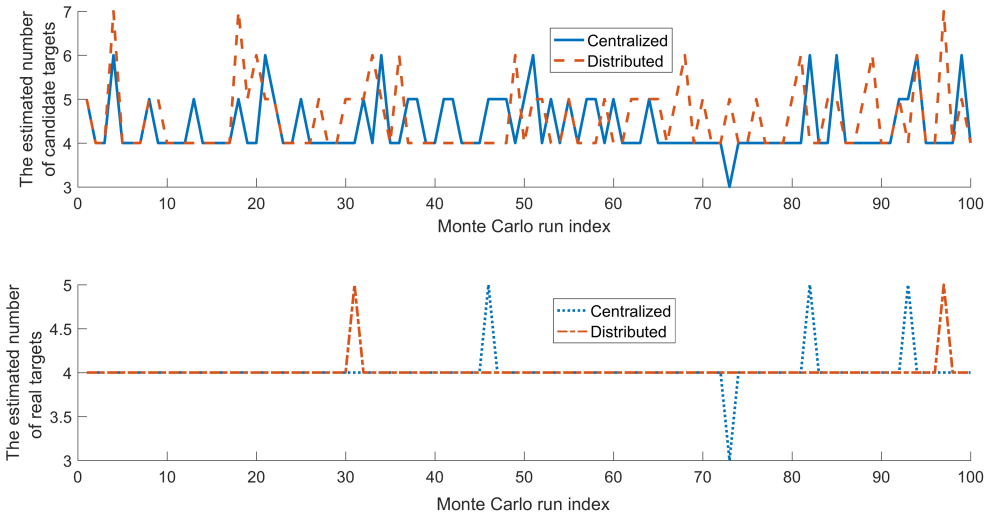


FIGURE 4.6.5: The number of targets (the truth is 4) estimated by the centralized and distributed algorithms before (top plot) and after (bottom plot) removing false targets using the threshold (4.5.31).

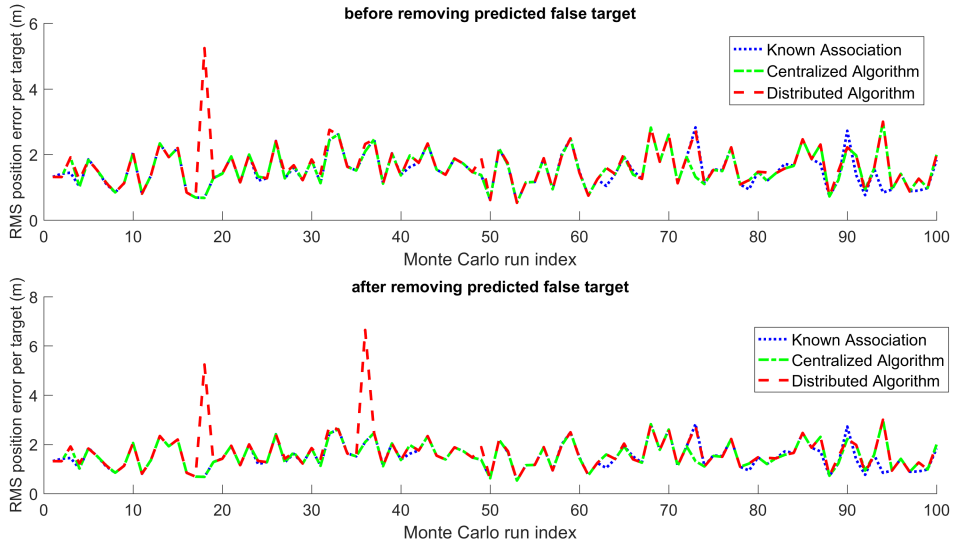


FIGURE 4.6.6: The RMS position error per target evaluated by assuming known target-measurement association, using the centralized EM algorithm and the EM and AC based distributed ADMM algorithm before and after removing the false targets.

TABLE 4.6.2: The final (after removing low pd targets) RMS ratio among assuming known association (KA), the centralized algorithm (C) and the distributed algorithm (D).

Interval	$(0.9, 0.99)$	$[0.99, 1.01]$	$(1.01, 1.1)$	$[1.1, 1.3)$	$[1.3, 1.5)$	$[1.5, 8]$
D versus C	3	84	3	6	1	3
C versus KA	12	75	3	4	2	2
D versus KA	15	60	6	9	6	4

target by solving a nonlinear least squares problem, and subsequently the position error can easily be obtained. From the same plot, it can be observed that the distributed algorithm yields the same position error as the centralized algorithm most of the time. While the baseline serves as a lower bound in most cases, it is interesting to note that the performance of the centralized algorithm or the distributed algorithm is better than the baseline in a few situations, which is due to "useful" false measurements.

The bottom plot in Figure 4.6.6 shows the RMS position error (averaged over all targets) for different cases after we remove the low $\hat{p}^{\mathrm{d}}$ false targets. This is also a measure of accuracy of the final position estimates provided by the centralized and distributed algorithm. For a clearer comparison, the range of the RMS ratio of the distributed algorithm over the centralized algorithm is also shown as in Table 4.6.2. While the distributed algorithm can produce a higher error than the centralized algorithm occasionally, in most cases (84%), it yields practically the same localization result as the centralized algorithm. It is interesting to note that the distributed algorithm can be slightly better than the centralized algorithm due to a different initialization.

4.7 Conclusion

This chapter considers passive localization of multiple transient emitters using a wireless sensor network and develops a distributed algorithm, which relies solely on local com-

munications between one-hop neighboring sensors. A distributed implementation of the centralized EM based algorithm is not possible unless the consensus on the initial set of estimates can be reached among all sensors. It is shown by simulation that even with the knowledge of data association, we need to carry out the initialization carefully because of the bearing measurements. Random initializations based on individual sets of bearing measurements could converge to a local minimum, therefore it is necessary to use the bearing measurements from neighboring sensors. As in the centralized EM based algorithm, each node uses a sequential m-best 2-D assignment algorithm on measurements from itself and its neighbors to obtain an initial set of target estimates.

Since initially estimated target set can be different from node to node in terms of both the cardinality (this happens when the probability of detection is low or the false alarm rate is high) and the values of estimates (since each sensor uses different measurements for initialization), we developed a distributed set consensus algorithm to reach consensus on the number of candidate targets before each node can reach consensus on the target-estimate association so that a proper initialization is obtained for the EM and AC based distributed ADMM algorithm. Since a Bernoulli measurement generation model is a more realistic assumption as it reflects best the physical process of measurement generation, we presented a likelihood function based thresholding technique to determine the number of targets.

Simulation results show that the EM and AC based distributed ADMM algorithm converges very fast and yields the target location estimates that are almost as good as those of the centralized algorithm. The estimated probability of detection is shown to be able to effectively distinguish real targets from false targets.

Chapter 5

Conclusions

In Chapter 2, We investigated several configurations with bistatic range and range rate measurements for collision warning. It turned out that a multistatic configuration is needed to provide good observability of the target, which is useful for collision warning. The minimum number of the transmitters required is three in a 3-D scenario and two in a 2-D scenario. We also implemented an ML estimator in both types of scenarios using the ILS technique and showed that the estimator can be considered as statistically efficient through Monte Carlo simulations for the scenarios considered. Based on the ML estimator, the collision warning was approached in two different ways. The first method is formulating the collision as a hypothesis testing problem using a generalized likelihood function, where the efficiency of the CPA time is also verified. The second method is a Bayesian formulation focusing on the time of CPA modeled as a random variable. Only the multistatic configuration with three transmitters is reliable for collision warning because the multistatic configuration with two transmitters based on the same target and ownship altitude assumption turns out to be prone to false alarms. When the minimum distance in the

Bayesian approach is the same as the safety margin in the likelihood based approach, both algorithms yield very similar collision warning performance.

In Chapter 3, the problem of localizing an unknown number of stationary transient emitters using passive sensors in the presence of missed detections and false alarms is investigated. Each measurement is based on one detection by a passive sensor and consists of a time of arrival and a bearing. It is assumed that measurements within a short time interval have to be associated before estimation. Both a Bernoulli measurement model and a Poisson measurement model are considered for each target. These two measurement models lead to two different proposed problem formulations: one is an S-D assignment problem and the other is a cardinality selection problem. The former can be solved by the Lagrangian relaxation algorithm reliably when the number of sensors is small. The sequential m-best 2-D (SEQ[m(2-D)]) assignment algorithm, which is resistant to the ghosting problem due to the estimation of the emitter signal's emission time, is developed to solve the problem when the number of sensors becomes large. Simulation results show that the SEQ[m(2-D) assignment algorithm is efficient for real time processing with reliable associations and estimates. In the cardinality selection formulation, a list of measurements is modeled as either realizations of a random variable with a uniform-Gaussian mixture (UGM) density or a Poisson point process (PPP). Because of an efficient way of incorporating false alarm rate, the UGM formulation is shown to be a useful alternative to the PPP formulation. Simulation studies show that both UGM and PPP formulations, which are based on the expectation maximization algorithm, require the right initial estimates to yield reliable localization results.

Chapter 4 investigates the problem of deploying a network of passive sensors to estimate the positions of an unknown number of stationary transient emitters. Since a completely connected network, which has a link between every pair of nodes, is not feasible

because of the power and bandwidth constraints, we developed a distributed algorithm that relies only on local communications between neighboring sensors. This distributed algorithm requires information diffusion within the network with the goal that every node achieves all target location estimates as accurate as a fusion center with centralized access to all information. The locations of the emitters are not completely observable by any single sensor since bearings and times of arrival with origin uncertainty are the only available measurements. These measurements are modeled as a realization of a Poisson point process at each sensor. The problem is formulated as a constrained optimization problem, which is solved via an alternating direction method of multipliers in a distributed manner based on the expectation maximization and averaging consensus algorithms. Consensus on the number of candidate targets as well as the inter-node estimate association are addressed so that the distributed algorithm converges to the maximum likelihood estimate. A likelihood function based approach using the estimated probability of detection is presented to determine the number of targets. Simulation results show that the distributed algorithm converges very fast and the root mean square error of target locations is almost as small as that obtained using the centralized algorithm. It is also shown that one can accurately determine the number of targets using the estimated probability of detection.

Proof of Proposition 1

For notational simplicity, let us denote

$$a_i = \sum_{\ell=1}^{N_{\rm s}} \sum_{j=1}^{n_\ell} w_{\ell ji}^{(n-1)}, i = 0, 1, \ldots, N \tag{.0.1}$$

$$p_i = p_{\rm d}(\mathrm{T}_i), i = 1, 2, \ldots, N \tag{.0.2}$$

$$\mathbf{p} = [p_1, p_2, \ldots, p_N] \tag{.0.3}$$

$$h_i(\mathbf{p}) = p_i - 1 \tag{.0.4}$$

Substitute (3.4.4) into (3.4.24), the problem becomes

$$\begin{aligned} \underset{\mathbf{p}}{\text{maximize}} \quad & f(\mathbf{p}) = a_0 \ln N_{\rm fa} + \sum_{i=1}^{N} a_i \ln p_i - (\sum_{i=0}^{N} a_i) \ln(\sum_{i=1}^{N} p_i + N_{\rm fa}) \\ \text{subject to} \quad & h_i(\mathbf{p}) \leq 0,\ i = 1, \ldots, N \end{aligned} \tag{.0.5}$$

Let μ_i be a Lagrange multiplier corresponding to $p_i \leq 1$ and $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_N]$. The Lagrangian is

$$L(\mathbf{p},\, \boldsymbol{\mu}) = f(\mathbf{p}) + \sum_{i=1}^{N} \mu_i (0 - h_i(\mathbf{p})) \tag{.0.6}$$

From the KKT conditions, the optimal values of $\mathbf{p}$ and $\boldsymbol{\mu}$ satisfy the following system of equations and inequalities

$$\begin{cases} 0 = \dfrac{\partial f}{\partial p_i} - \mu_i \dfrac{\partial h_i}{p_i} = \dfrac{a_i}{p_i} - \dfrac{\sum_{i=0}^{N} a_i}{\sum_{i=1}^{N} p_i + N_{\text{fa}}} - \mu_i \qquad i = 1, 2, \ldots, N & (.0.7) \\ 0 = \mu_i h_i(\mathbf{p}) = \mu_i (p_i - 1) \qquad i = 1, 2, \ldots, N & (.0.8) \\ 0 \leq \mu_i \qquad i = 1, 2, \ldots, N & (.0.9) \end{cases}$$

We need to break the analysis into cases according to (.0.8).

Case 1: If

$$\mu_i = 0 \qquad i = 1, 2, \ldots, N \tag{.0.10}$$

then (.0.7) is simplified to

$$p_i = \frac{a_i (\sum_{k=1}^{N} p_k + N_{\text{fa}})}{\sum_{i=0}^{N} a_i} \tag{.0.11}$$

Summing over i from 1 to N, we have

$$\sum_{i=1}^{N} p_i = \sum_{k=1}^{N} p_k = \frac{\sum_{i=1}^{N} a_i (\sum_{k=1}^{N} p_k + N_{\text{fa}})}{\sum_{i=0}^{N} a_i} \tag{.0.12}$$

which can be simplified to

$$\sum_{k=1}^{N} p_k = \frac{\sum_{i=1}^{N} a_i N_{\text{fa}}}{a_0} \tag{.0.13}$$

Substituting (.0.13) into (.0.11), we have

$$p_i = \frac{a_i N_{\text{fa}}}{a_0} \tag{.0.14}$$

The feasibility of this solution depends on whether $\frac{a_i N_{\text{fa}}}{a_0}$ is greater than 1. Let

$$\mathcal{S} = \{i \,|a_i N_{\text{fa}} > a_0\} \tag{.0.15}$$

If the set $\mathcal{S}$ is empty, (.0.14) will be the optimal solution for $\mathbf{p}$. If the set $\mathcal{S}$ is not empty, then we must have

$$\begin{cases} 0 = \mu_i h_i(\mathbf{p}) = \mu_i(p_i - 1) & i \notin \mathcal{S} \quad (.0.16) \\ 0 \leq \mu_i & i \notin \mathcal{S} \quad (.0.17) \\ 0 = p_i - 1 & i \in \mathcal{S} \quad (.0.18) \\ 0 < \mu_i & i \in \mathcal{S} \quad (.0.19) \end{cases}$$

Case 2: If

$$\begin{cases} 0 = \mu_i & i \notin \mathcal{S} \quad (.0.20) \\ 1 = p_i & i \in \mathcal{S} \quad (.0.21) \end{cases}$$

then (.0.7) is simplified to

$$p_i = \frac{a_i(\sum_{k \notin \mathcal{S}} p_k + |\mathcal{S}| + N_{\text{fa}})}{\sum_{i=0}^{N} a_i}, \qquad i \notin \mathcal{S} \tag{.0.22}$$

Summing over i which is not in the set $\mathcal{S}$ and solving for $\sum_{i \notin \mathcal{S}} p_i$,

$$\sum_{i \notin \mathcal{S}} p_i = \frac{\sum_{i>0, i \notin \mathcal{S}} a_i(|\mathcal{S}| + N_{\text{fa}})}{\sum_{i>0, i \in \mathcal{S}} a_i + a_0} \tag{.0.23}$$

Substituting (.0.23) into (.0.22), we have

$$p_i = \frac{a_i(|\mathcal{S}| + N_{\text{fa}})}{\sum_{i>0, i\in\mathcal{S}} a_i + a_0}, \qquad i \notin \mathcal{S} \tag{.0.24}$$

Since

$$a_i N_{\text{fa}} > a_0, \qquad i \in \mathcal{S} \tag{.0.25}$$

we have

$$\sum_{i>0, i\in\mathcal{S}} a_i N_{\text{fa}} > |\mathcal{S}| a_0 \tag{.0.26}$$

$$\sum_{i>0, i\in\mathcal{S}} a_i N_{\text{fa}} + a_0 N_{\text{fa}} > |\mathcal{S}| a_0 + a_0 N_{\text{fa}} \tag{.0.27}$$

$$\frac{N_{\text{fa}}}{a_0} > \frac{|\mathcal{S}| + N_{\text{fa}}}{\sum_{i>0, i\in\mathcal{S}} a_i + a_0} \tag{.0.28}$$

Since

$$a_i N_{\text{fa}} \leq a_0, \qquad i \notin \mathcal{S} \tag{.0.29}$$

we have

$$p_i = \frac{a_i(|\mathcal{S}| + N_{\text{fa}})}{\sum_{i>0, i\in\mathcal{S}} a_i + a_0} < \frac{a_i N_{\text{fa}}}{a_0} \leq 1, \qquad i \notin \mathcal{S} \tag{.0.30}$$

which verifies the feasibility of the solution consisting of (.0.21) and (.0.24). One can

summarize the two cases as follows

$$p_i = \begin{cases} 1 & \text{if } i \in \mathcal{S} \\ \dfrac{a_i(|\mathcal{S}| + N_{\text{fa}})}{\sum_{i>0, i\in\mathcal{S}} a_i + a_0} & \text{if } i \notin \mathcal{S} \end{cases} \tag{.0.31}$$

which is equivalent to (3.4.37) because of (.0.1) and (.0.2).

Bibliography

[1] H. Akaike

A new look at the statistical model identification.

IEEE Transactions on Automatic Control, 19, (6): 716–723, Dec. 1974.

[2] B. M. Albaker, and N. A. Rahim

A survey of collision avoidance approaches for unmanned aerial vehicles.

In *International Conference for Technical Postgraduates*, Dec. 2009.

[3] J. Areta, Y. Bar-Shalom, M. Levedahl, and K. Pattipati

Hierarchical track association and fusion for a networked surveillance system.

Journal of Advances in Information Fusion, 1, (2): 140–157, Dec. 2006.

[4] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan

Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.

New York: Wiley, 2001.

[5] Y. Bar-Shalom, P. Willett, and X. Tian

Tracking and Data Fusion.

Storrs, CT: YBS Publishing, 2011.

[6] C.M. Bishop

Pattern Recognition and Machine Learning.

Springer-Verlag, 2006.

[7] S. Blackman

Multiple hypothesis tracking for multiple target tracking.

IEEE Aerospace and Electronic Systems Magazine, 19, (1): 5–18, Jan. 2004.

[8] S. S. Blackman, and R. Popoli

Design and Analysis of Modern Tracking Systems.

Artech House, 1999.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein

Distributed optimization and statistical learning via the alternating direction method of multipliers.

Foundations and Trends in Machine Learning, 3, (1): 1–222, Jan. 2011.

[10] C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo

A novel 3D geometric algorithm for aircraft autonomous collision avoidance.

In *Proceedings of 45th IEEE Conference on Decision and Control*, 1580–1585, Dec. 2006.

[11] B. K. Chalise, Y. D. Zhang, M. G. Amin, and B. Himed

Target localization in a multi-static passive radar system through convex optimization.

Signal Processing, 102, (0): 207–215, 2014.

[12] C.Y. Chong

Hierarchical estimation.

In 2nd MIT/ONR Workshop on Distributed Information and Decision Systems Motivated by Naval Command-Control-Communication (C3) Problems, July 1979.

[13] C.Y. Chong, K.C. Chang and S. Mori

Comparison of optimal distributed estimation and consensus filtering.

In Proceedings of 19th International Conference on Information Fusion, Heidelberg, July 2016.

[14] C.Y. Chong, S. Mori and K.C. Chang

Distributed Multitarget Multisensor Tracking.

Multitarget-Multisensor Tracking: Applications and Advances, Volume I, Yaakov Bar-Shalom, Editor, Chapter 8, Artech House, 1990.

[15] C.Y. Chong, S. Mori, E. Tse, and R.P. Wishner

Distributed estimation in distributed sensor networks.

In Proceedings of the 1982 American Control Conference, 1982.

[16] C.Y. Chong, E. Tse, and S. Mori

Distributed estimation in networks.

In Proceedings of the 1983 American Control Conference, 1983.

[17] D. Daley, and D. Vere-Jones

An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods.

Springer-Verlag, 2003.

[18] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom

A generalized S-D assignment algorithm for multisensor-multitarget state estimation.

IEEE Transactions on Aerospace and Electronic Systems, 33, (2): 523–538, April 1997.

[19] A.P. Dempster, N.M. Laird, and D.B. Rubin

Maximum likelihood from incomplete data via the EM algorithm.

Journal of the Royal Statistical Society, Series B, 39, (1): 1–38, 1977.

[20] W. Dou, Y. Bar-Shalom, L. Kaplan and J. George

Distributed fusion algorithms for passive localization of multiple transient emitters.

Journal of Advances in Information Fusion, under review.

[21] W. Dou, Y. Bar-Shalom and P. Willett

Tracking filter initialization with Doppler-biased multistatic time-of-arrival measurements.

In *Proceedings of SPIE #9092, Signal and Data Processing of Small Targets*, June 2014.

[22] W. Dou, Y. Bar-Shalom and P. Willett

Bistatic measurement fusion from multistatic configurations for air collision warning.

Journal of Advances in Information Fusion, 10, (2): 163–182, Dec. 2015.

[23] W. Dou, Y. Bar-Shalom and P. Willett, and X. Song

Initialization and tracking using Doppler-biased multistatic time-of-arrival measurements with linear frequency modulated waveforms

In *Proceedings of 17th International Conference on Information Fusion*, July 2014.

[24] W. Dou, J. George, L. Kaplan, R. W. Osborne and Y. Bar-Shalom

Assignment and EM approaches for passive localization of multiple transient emitters.

In *Proceedings of SPIE #9842, Signal Processing, Sensor/Information Fusion, and Target Recognition XXV*, May 2016.

[25] W. Dou, J. George, L. Kaplan, R. W. Osborne, and Y. Bar-Shalom

Evaluation of fusion algorithms for passive localization of multiple transient emitters.

Journal of Advances in Information Fusion, under review.

[26] W. Dou, P. Willett and Y. Bar-Shalom

Fusion of range-only measurements from multistatic configurations for air collision warning.

In *Proceedings of IEEE Aerospace Conference*, March 2015.

[27] W. Dou, P. Willett and Y. Bar-Shalom

Configuration selection for fusion of range and Doppler measurements from multistatic radars for air collision warning.

In *Proceedings of 18th International Conference on Information Fusion*, July 2015.

[28] H. Durrant-Whyte and M. Stevens

Data fusion in decentralised sensing networks.

In Proceedings of 4th International Conference on Information Fusion, 2001.

[29] P.A. Forero, A. Cano, and G.B. Giannakis

Distributed clustering using wireless sensor networks.

IEEE Journal of Selected Topics in Signal Processing, 5, (4): 707–724, Aug. 2011.

[30] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe

Sonar tracking of multiple targets using joint probabilistic data association.

IEEE Journal of Oceanic Engineering, 8, (3): 173–184, July 1983 .

[31] J. George, and L. Kaplan

Shooter localization using a wireless sensor network of soldier-worn gunfire detection systems.

Journal of Advances in Information Fusion, 8, (1): 15–32, June 2013.

[32] J. George, L. Kaplan, S. Deligeorges, and G. Cakiades

Multi-shooter localization using finite point process.

In *Proceedings of Seventeenth International Conference on Information Fusion*, July 2014.

[33] J. George, and L. Kaplan

A finite point process approach to multi-agent localization using transient measurements.

Journal of Information Fusion, 32, (A): 62–74, Nov. 2016.

[34] D. Gu

Distributed EM algorithm for Gaussian mixtures in sensor networks.

IEEE Transactions on Neural Networks, 19, (7): 1154–1166, July 2008.

[35] V. Indelman, E. Nelson, J. Dong, N. Michael and F. Dellaert

Incremental distributed inference from arbitrary poses and unknown data association: using collaborating robots to establish a common reference.

IEEE Control Systems Magazine, 36, 2(Apr. 2016), 41–74.

[36] H.W. Kuhn, and A.W. Tucker

Nonlinear Programming.

In *Proceedings of Second Berkeley Symposium on Mathematical Statistics and Probability*, 481–492, 1951.

[37] Y. K. Kwag, and C. H. Chung

UAV based collision avoidance radar sensor.

In *IEEE International Geoscience and Remote Sensing Symposium*, 639–642, July 2007.

[38] R. Mahler

Multitarget Bayes filtering via first-order multitarget moments.

IEEE Transactions on Aerospace and Electronic Systems, 39, (4): 1152–1178, Oct. 2003.

[39] M. Malanowski, and K. Kulpa

Two methods for target localization in multistatic passive radar.

IEEE Transactions on Aerospace and Electronic Systems, 48, (1): 572–580, Jan. 2012.

[40] T.W. Martin and K.C. Chang

A distributed data fusion approach for mobile ad hoc networks.

In Proceedings of 7th International Conference on Information Fusion, 2005.

[41] J. Moller, and R.P. Waagepetersen

Statistical Inference and Simulation for Spatial Point Processes.

Chapman and Hall/CRC, 2003.

[42] J.F.C. Mota, J.M.F. Xavier, P.M.Q. Aguiar, and M. Püschel

D-ADMM: a communication-efficient distributed algorithm for separable optimization.

IEEE Transactions on Signal Processing, 61, (10): 2718–2723, May 2013.

[43] K.G. Murty

An algorithm for ranking all the assignments in order of increasing cost.

Operations Research, 16, (3): 682–687, 1968.

[44] R.D. Nowak

Distributed EM algorithms for density estimation and clustering in sensor networks.

IEEE Transactions on Signal Processing, 51, (8): 2245–2253, Aug. 2003.

[45] R. W. Osborne, III and Y. Bar-Shalom

Statistical efficiency of composite position measurements from passive sensors.

IEEE Transactions on Aerospace and Electronic Systems, 49, 4 (Oct. 2013), 2799–2806.

[46] R.W. Osborne, Y. Bar-Shalom, J. George, and L. Kaplan

Data fusion from multiple passive sensors for multiple shooter localization via assignment.

In *Proceedings of Seventeenth International Conference on Information Fusion*, July 2014.

[47] R.W. Osborne, Y. Bar-Shalom, J. George, and L. Kaplan

Statistical efficiency of target localization from angle and shockwave measurements.

Journal of Advances in Information Fusion, 9, (2): 75–89, Dec. 2014.

[48] K.R. Pattipati, S. Deb, Y. Bar-Shalom, and R.B. Washburn

A new relaxation algorithm and passive sensor data association.

IEEE Transactions on Automatic Control, 37, (2): 198–213, Feb. 1992.

[49] D. Pelleg and A.W. Moore

X-means: extending K-means with efficient estimation of the number of clusters.

In Proceedings of the Seventeenth International Conference on Machine Learning, 2000.

[50] 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, 56, (1): 350–364, Jan. 2008.

[51] G. Schwarz

Estimating the dimension of a model.

The Annals of Statistics, 6, (2): 461–464, 1978.

[52] J. Sijbers, A. J. den Dekker, P. Scheunders, D. Van Dyck

Maximum-likelihood estimation of Rician distribution parameters.

IEEE Transactions on Medical Imaging, 17, 3 (June 1998), 357–361.

[53] D. Sislak, M. Rehak, M. Pechoucek, D. Pavilicek and M. Uller

Negotiation-based approach to unmanned aerial vehicles.

In *IEEE 2006 Workshop on Distributed Intelligent Systems*, 279–284, June 2006.

[54] M. Švecová, D. Kocur, R. Zetik, and J. Rovňáková,

Target localization by a multistatic UWB radar.

In *Proceedings of 20th International Conference Radioelektronika*, April 2010.

[55] C. J. Tomlin, J. Lygeros, and S. S. Sastry

A game theoretic approach to controller design for hybrid systems.

Proceedings of the IEEE, 88, 7 (July 2000), 949–970.

[56] H. L. Van Trees, K. L. Bell, and Z. Tian

Detection Estimation and Modulation Theory, 2nd Edition, Part I, Detection, Estimation, and Filtering Theory.

New York: Wiley, 2013.

[57] J. -H. Wen, J. -S. Li, C. -Y. Yang, C. -H. Chen, and H. -C. Chen

Localization scheme of multistatic radars system based on the information of measured signal.

In *Proceedings of 9th International Conference on Broadband and Wireless Computing, Communication and Applications*, 462–466, Nov. 2014.

[58] P. R. Williams

Aircraft collision avoidance using statistical decision theory.

In *Proceedings of SPIE 1694, Sensors and Sensor Systems for Guidance and Navigation II*, July 1992.

[59] L. Xiao, and S. Boyd

Fast linear iterations for distributed averaging.

Systems and Control Letters, 53: 65–78, 2004.

[60] L. Xiao, S. Boyd, and S. Lall

A scheme for robust distributed sensor fusion based on average consensus.

In *Fourth International Symposium on Information Processing in Sensor Networks*, 63–70, Apr. 2005.

[61] S. Zhang, and Y. Bar-Shalom

Practical data association for passive sensors in 3D.

Journal of Advances in Information Fusion, 9, (1): 38–46, June 2014.