Thesis for the Degree of Doctor of Philosophy

Multiple Object Localization based on Acoustic Signals in Wireless Sensor Networks

Viet-Hung Dang

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Korea

February, 2012

Multiple Object Localization based on Acoustic Signals in Wireless Sensor Networks

Viet-Hung Dang

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Korea

February, 2012

Multiple Object Localization based on Acoustic Signals in Wireless Sensor Networks

by

Viet-Hung Dang

Advised by

Professor Sungyoung Lee

Submitted to the Department of Computer Engineering

and the Faculty of the Graduate School of

Kyung Hee University in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Dissertation Committee:

Professor TaeChoong Chung, Ph.D. . . . .

Professor Young-Koo Lee, Ph.D. . . . .

Professor Dong Han Kim, Ph.D. . . . .

Professor Brian J. d'Auriol, Ph.D. . . . .

Professor Sungyoung Lee, Ph.D. . . . .

Multiple Object Localization based on Acoustic Signals in Wireless Sensor Networks

by

Viet-Hung Dang

Submitted to the Department of Computer Engineering
on Jan 6, 2012, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

A great deal of research achievements for object localization in Wireless Sensor Networks (WSNs) have been obtained in recent years. In order for a WSN-based system to locate the objects, two localization stages must be carried out: locating the sensors' positions because the collected data without position information is meaningless, and then locating the objects' positions. This thesis focuses on solving the problems of both stages of localization.

For sensor localization, the problem has been considered thoroughly and a lot of solutions have been proposed for many years. Nevertheless, its interesting challenges in terms of cost-reduction, accuracy improvement, scalability, and distributed ability design have encouraged us to develop a new algorithm, the Push-pull Estimation (PPE). It is a geometry based algorithm, in which the differences between measurements and current calculated distances are modeled into forces, dragging the sensors or nodes close to their actual positions. Based on very few known-location sensors or beacons, PPE can pervasively estimate the coordinates of many unknown-location sensors. Each unknown-location sensor, with given pairwise distances, could independently estimate its own position through remarkably uncomplicated calculations. Characteristics of the algorithm are examined through analyses and simulations to demonstrate that it has advantages over those of previous works in dealing with the above challenges.

For object localization, we focus on solving the problem of positioning multiple moving objects based on acoustic signals. It is because acoustic source localization has many important applications particularly for military tracking foreign objects. Even though WSNs have been developed for long time, this localization remains a big incompletely solved problem. A system for source localization must have the ability to deal with the recorded convolved mixture signals while minimizing the high communication and computation cost. This work introduces a distributed design for positioning multiple independent moving sources based on acoustic signals in which we focus on utilizing the relative information of magnitudes recorded at different sensors. The sensors perform pre-processing on the sensed data to capture the most important information before compressing and sending the extracted data to the base. At the base, the data is uncompressed and the source locations are inferred via two clustering stages and an optimization method.

Analysis and simulation results lead to the conclusion that in both stages, our system provides good accuracy and needs neither much communication nor complex computation in a distributed manner. For source positioning, it works well when there exists high noise with Rayleigh multipath fading under Doppler effect and even when the number of independent sources is greater than the number of microphone sensors.

Thesis Supervisor: Sungyoung Lee
Title: Professor

Acknowledgments

I would like to express my thanks to many people who have directly and indirectly helped me so that I can complete the study of Ph.D course. First and foremost, I especially thank my supervisor and mentor Professor Sungyoung Lee, who has guided and supported me throughout the time I am in KyungHee University with warm encouragement and insightful advices. He has led me to the interest of Wireless Sensor Networks by his relentless enthusiasm. I also thank Professor Young-Koo Lee for his great helps during my study in KyungHee University.

I sincerely appreciate other Professors who were in charge of the classes I have taken during the study. I am particularly much influenced by Professor Gon Khang with his profound understanding of linear analysis, Professor Brian J. d'Aurial with his guide to achieve the recursive view of functional programming, Professor Yun-Hee Kim with her lectures on regressions, Professor TaeChoong Chung with his first opening on Artificial Intelligence algorithms to me and Professor Tae-Seong Kim with his lessons on Pattern Recognition. Their wisdom and guidance have contributed greatly to consolidating and widening my knowledge from background to higher levels. I have much enjoyed my time in their classes where the passion of study has been extended.

I would like to especially thank my thesis committee whose useful comments have helped to improve and complete this thesis and its organization. I also especially thank my friends Ba-Vui Le, Nguyen Duc Thang and Hoang Huu Viet who helped me to conduct data collection for the experiments in cold nights without hesitation; Pham The Anh and La Vinh-The who lent and supported the hardware.

My next thanks are for many friends in Ubiquitous Computing Laboratory: Tran Minh Tam, Phan Tran Ho Truc, Xuan Hung Le, Le Viet Duc, La Vinh-The, Ba-Vui Le, Nguyen Anh Tu, Pham The Anh, Luong Thi Thu Phuong, Trinh Thi Lan Anh, Dinh Dong Luong, Nguyen Duc Thanh Nhan, Asad Masood Khattak, Zeeshan Pervez, Muhammad Fahim, Manhyung Han, Yong-Koo Han, Seong Jin Cho, etc., who have been very good, enthusiastic and helpful lab-mates. I especially thank my roommate Huynh Le Minh and my close friends Ngo Anh Vien and Nguyen Duc Thang for their encouragement and helps. They are the ones who I can always share the burden of stress and have the convenience of co-working. I also thank other Vietnamese friends in the campus during my study: Ngo Thanh An, Le Dinh Phong, Le-Nam Tran, Vien Quoc Tuan, Ly Hoang Vu, Le Tuan Anh, Hoang Huu Viet, Dang Trung Sinh, Tran Nhat Huy, Nguyen Minh Tri, Le Duc Anh, etc. for guiding, helping and sharing the abroad time.

Last but not least, this thesis would not have been completed if there had not been the constant support, boundless love and encouragement from my family in Vietnam. The motivation is also contributed by the love and belief of a wonderful girl who is waiting for my graduation. I want to dedicate this thesis to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless sensor networks (WSNs) are sets of sensors spatially deployed mainly for many monitoring applications. One of the most popular purposes of tracking using WSNs is the determination of object locations. It is necessary especially in the military when WSNs have been deployed to detect foreign, or unwelcome, objects and their locations within the deployed region. In order to deal with the mentioned problem, a prerequisite work must be considered. That is the problem of sensor localization in WSNs so that the recorded signals are marked with positions and the object positioning task is performed. In fact, WSNs are used for applications of observing, tracking, and controlling. Observing includes collecting data like parameters of temperature, humidity, vibration, pressure, traffic, event counting, etc. Tracking uses the observed data to follow and sometimes to predict changing data where tracking moving objects is an example. Meanwhile, controlling pertains to decisions in response to data provided by observing and tracking, like cutting power, sending instructions to control traffic, putting out fires, sending alarms, releasing missiles to detected enemy devices, and so on. Evidently, applications of WSNs are all related to the physical locations of sensed data where object tracking/positioning is just one specific example. This means sensor localization plays an important role and it will take half of this thesis focus. The thesis taxonomy is depicted in Fig.1.1. The thesis concentrates on two main parts: sensor localization and object localization. Our contribution for the former issue is Push-pull Estimation (PPE) method and the one for the later issue is call Acoustic multiple object positioning

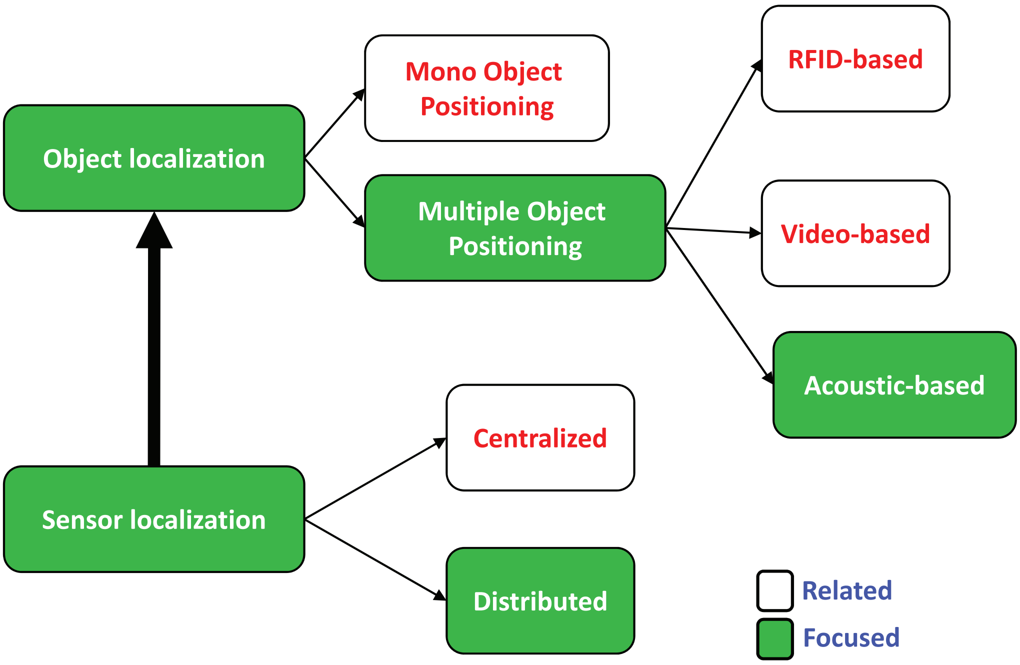


Figure 1.1: Thesis's taxonomy.

(AMOP) system.

1.1 Sensor localization and schemes

The position of a sensor, or the prerequisite information, can be achieved easily with an integrated Global Positioning System (GPS) module. However, there is one problem with GPS making localization algorithms necessary in practical implementation. It is the expensive cost of GPS devices on the sensors, causing the total cost considerably high when the number of sensors in a WSN is large (an integrated GPS module at the time being is around 40$ [97][98] and the number of nodes in WSNs can be up to hundreds or even thousands). As a result, only few nodes (a.k.a. beacons, anchors or pilots) have the known locations with GPS modules. The locations of the remaining sensors (called normal nodes) are automatically estimated based on the measurements to their neighbors using a localization algorithm. Generally, sensor localization categorizations are usually made depending on the kind of input data: range-based or range-free; by what methodology the algorithm is performed: centralized or decentralized; and whether the result is gained directly or through iterations: multilateration or successive refinement. Short introductions on these concepts are discussed below.

a/ Range-based and range-free data

The range-based scheme uses input pair-wise distance data. These measurements can be inferred from such data as received signal strength (RSS), time of arrival (TOA), time of difference of arrival (TDOA), and angle of arrival (AOA) [61]. Measurement error in RSS suffers from fading and shadowing, leading to poor accuracy of localizations using this scheme. Meanwhile, TOA and TDOA also use radio frequency (RF) and acoustic signal as RSS does, but the ranges are estimated based on the time delay of propagation through the environment (TOA) or the time discrepancy of an incoming signal at two different nodes (TDOA) [78]. These measurement methods are more accurate and easier to analyze with the popular Gaussian noise model and, as a trade-off, are more expensive than RSS systems.

For range-free schemes, the input data of localization algorithms is the connectivity information between nodes and their neighbors or the information of who are in the communication range [70][76]. A node can also estimate the distances to other nodes by counting the shortest paths (minimum numbers of connections) from itself to other nodes and multiplying this value with the mean hop distance [58]. The mean hop distance may be calculated by averaging all of the hops that make up the shortest paths between anchors. Ranges based on connectivity do not require complicated hardware or high interface power in normal nodes. However, disadvantages of this scheme are: errors of measurements are inevitable; the number of anchors should be high; and the distribution of nodes should be uniform [58].

Since range-based algorithms and most range-free algorithms work on the estimated ranges of pair-wise distances, we do not distinguish between these two schemes. Instead, we concentrate on solving the localization problem with the calibrated measurements. Push-pull Estimation (PPE) actually can be seen as a method using the range-based scheme. It is built to deal with not only Gaussian error but with any unbiased error as well. PPE can also solve the problem of biased error measurement input if a de-biasing function is found.

b/ Centralized and decentralized methodologies

A centralized localization is executed by a single computer. It therefore can easily deal with data combinations and perform complicated computations like matrix operations or eigenvalue, eigenvector calculations of a large matrix [54][62][99]. However, the cost of communications becomes dominant when all of the input data enters into one node, and all of the output data has to exit from only that node. Moreover, the protocols for this kind of method are hard to be implemented. The running time could be long if the complexity of the algorithm is high, especially when the number of nodes in the network becomes large. In other words, the scalability of centralized algorithms is limited. Another variation of the centralized algorithm is related to clustering methods where the whole network is divided into groups, and each group has a central computer to run the algorithm [104]. Yet the disadvantages of this method are difficulties in defining how the groups are divided. The central computer of a group would have strong computing hardware compared to that of a normal node, so it must be carefully deployed so that network communication and computation are effective.

To overcome above disadvantages, researchers have been seeking other methods which can be run distributedly on the simple computers of normal nodes [15][47][58][70][76]. Algorithms of this manner are called decentralized, distributed, pervasive or collaborative localizations. Real-world applications prefer this type of algorithm in which the big load of computations is shared, the calculating time is reduced considerably, and communication bottle-neck is no longer a problem. Our proposed PPE is designed to avoid all of the disadvantages of centralized or clustering-based centralized algorithms. It is purely a distributed method performed by all of the normal nodes in the network.

c/ Multilateration and successive refinement manner

Multilateration is usually used in distributed algorithms. It takes at least three input range measurements (in 3-D case, four is the least) to calculate a node's coordinates [49][90]. The ratio of the anchor number to the total node number for this manner is often high because multilateration requires the exact positions of the reference nodes.

Meanwhile, successive refinement obtains the estimated coordinates of all normal nodes through iterations. If it is designed in a decentralized method, each node has to calculate the estimated position, send this information to the network, and receive new estimated positions from other nodes. These tasks are performed repeatedly until convergence. Successive refinement in a distributed methodology is usually more difficult to design, and its convergence is surely a problem [61]. However, it reduces the communication cost and makes the number of anchors not necessary to be as high as in multilateration. In our proposed algorithm, successive updating is used in all sub-stages of the method.

In this work, sensor localization is solved with a proposed method which is range-based and distributed with successive refinement manner. The assumption is that the distance measurements are given (range-based) and unbiased, then the proposed method works robustly with simple calculation and guaranteed convergence.

1.2 Acoustic based object localization

Object localization plays an important role in military application to position foreign objects in the deployed region and recognize if they are enemies. Unlike the active devices that emit radio or ultrasonic signals to detect objects based on the reflected waves, sensors in WSNs are usually passive and only record the signals from objects. Thus, most developed tracking techniques with passive sensors are based on known communications channels. In other words, they are only suitable for localizing objects that are designed to be monitored, not for the very important security and defense application of localizing foreign objects. These objects have no prior channels to communicate with the system and the only detected signals that the system can capture are images and sounds. For image signals, however, cameras cannot be deployed randomly or casually in a large number due to high costs, view blockage and the limited angle of view. Moreover, the communication cost would be very high for image transmission while the computation for positions needs a centralized method. Therefore, the easiest and most convenient way to monitor an object is using acoustics. Although localizing objects based on the emitted acoustic signals has been considered, many works focus on one-source tracking [2][12][13][37]. This thesis describes a full design for positioning multiple moving objects that emit sound signals and the analysis of its performance. The design overcomes the drawbacks of both previous works and the previous version of the system. The input data is not simply the convolved mixtures, which are just the combinations of signals experiencing different time delays, but is more complicated mixtures in which Doppler effect and Rayleigh multi-path fading co-exist. Signals that propagate in free environment like air, water, or vacuum usually take different paths to receivers because of shadowing and reflecting effects. Some works consider this kind effect of acoustic signal with a reverberation model which is often used for indoor settings [23][26][36][82][83]. Our work pays attention on outdoor localization, so the multi-path fading is model with the popular Young model often used in telecommunications. Meanwhile, extracted information from the mixtures in this thesis is not of time-delay differences or direction of arrivals but of the ratios between Received Signal Strengths (RSS) from each object to different sensors. We remark that we are the first to address the multi-object positioning problem by focusing on the ratios of source magnitudes extracted from the complicated convolved mixture data which includes Doppler effect and Rayleigh multi-path fading. For multiple sources positioning, or even just for separation, the previous works just deal with stand-still sources and all require high-cost for both data communications and algorithm computation [1][10][49][52][71][75][105]. In addition, the sensors deployed in the area are not the arrays of microphones as in many previous works [9][27][40][46][68][92] but the isotropic acoustic recorders as in [7][33]. That means in this thesis, the main feature used for source positioning is not the directions from the sensors to the sources. In fact, we try to extract the distance relations from the ratios of pairs of frequency component to pairs of sensors. The details will be presented in later Chapters. More importantly, aiming for a low-cost method, we make the method applicable into WSNs in a distributed way, where the whole computation load is shared on the sensors with a low communication cost for data collection. Distributed approaches are categorized into *data decomposition*, *process decomposition* or *data-and-process decomposition*, depending on how the algorithm is shared on different computers [4]. In this regard, our system is designed for working in *process and data decomposition* manner whose details are discussed in Chapter 3, Section 3.3. Sensed data is pre-processed, compressed and sent to the base. The base computer decompresses the data and extracts the information of ratios between the energies of each dominant frequency component (f-component) collected at different sensors using a clustering method on the frequency domain. These ratios are then used to estimate the positions of all dominant f-components, and the source location estimations are computed by clustering these f-component positions.

1.3 Problem statement

This thesis tries to deal with the problem of positioning multiple moving source. It considers the scenario which is popular in security application and military. Imagine there is a wide area that a military unit wants to monitor in order to detect unwelcome foreign objects. That unit sends a group of soldiers to the area to deploy monitoring sensors. They come there and quickly throw the sensors randomly in the vast area while driving on a vehicle or on a plane before withdrawing back to the base. Because of the expensive cost of GPS module mentioned above, only a small portion in number of the sensors have this module and all the rest of sensors must locate themselves under the range-based scheme. Now the sensors start to communicate with their neighbors within the communication ranges to setup a network. They also estimate the distances to their neighbors, several of which are location-known due to the installed GPS modules. The location-unknown sensors then run a program to locate themselves and send their locations to the military base by transferring packages of information via other sensors. But it is not just that since the more important mission is to locate the objects that emit sound signals within the monitoring area. Each sensor may be programmed to wake up when sound signals are detected at some threshold level and begin collecting the sound data. The wireless sensor network then, by some way, must locate the sources moving in it. The first problem is: how the sensors can locate themselves to transfer this information to the base given the estimated ranges to their neighbors and to several beacons. The second problem is how they can work with the base to locate multiple moving sources while the data load is big due to high sampling frequency and the data is the convolved mixtures with Doppler effect. The two later Chapters will alternately present proposed one method to each problem with full description and analysis. Chapter 2 is for sensor localization and Chapter 3 is for the next mission of positioning the multiple moving sources. Both methods in both stages work in a distributed manner which we can see in later descriptions.

1.4 Related works

Previous works on object localization usually rely on the assumption that the locations of sensors are available and neglect the importance of how to get this information [2][3] [12][13][37][40][96]. Other works for estimating sensor location information only focus on sensor localization [47][57] [61]. That means previous works pay attention to solving either sensor localization or source localization, but not both. That leads to the problem of combining these two tasks into one whole work. Designing modules for a sensor will be more efficient when the whole task of both sensor localization and object positioning stages are considered. The reason is that the designer can utilize both hardware and software modules of one stage for the another. For example, if the sensor has the microphone module to position the object via acoustic signals, then the TOA or TDOA schemes [49][99] for sensor localization should be used. In addition, sensors in WSNs usually communicate with their neighbors for the purposes of transferring data and maintenance the networks, the RSS schemes [15][47][50][61] can also be utilized to enhance the sensor localization accuracy. This thesis gives contributions in both stages: the prerequisite information and its related problem of sensor localization will be considered via a proposed method with full analysis called Push-Pull Estimation [17], and the locations of multiple moving object will be determined by the Acoustic Multiple Object Positioning (AMOP) [19] system. In the following part, we separately introduce what previous works have been achieved for both sensor localization and acoustic based object localization.

Related works of sensor localization

For sensor localization, pairwise distance is usually used for position estimating, even range-free methods. Besides few methods like RF-based fingerprint matching [102] and grid-scanning [32][76] that do not use range estimation, the other range-free methods, to improve accuracy, estimate distances by counting the hops between nodes and then using these estimates as their inputs. For these ones, distance-vector based (DV-based) positioning [58] is one good way for approximating the ranges. Apparently, pairwise distance is quite necessary not only with range-based schemes but also in range-free schemes. In both of these schemes, the statistical model of pairwise measurement is usually assigned to the very popular Gaussian model. This model represents the most unpredicted noise (maximum entropy) and has a parametric function which is easy for analysis. However, it is only appropriate for TOA, TDOA, and AOA measurements [99]. In WSNs, where RF signals are always available, RSS is likely the cheapest and most convenient means for range-based schemes. Its error model, however, is not typical Gaussian but a log-normal distribution, or Gaussian in the log-domain [15][47][50][61]. In [50], although Malaney does not give solutions, he presents how log-normal fading models influence location accuracy and also gives the mathematical analysis for the combination model of multipath fading and log-normal power shadowing. Another technique using parametric channel models is proposed in [87] to reduce error by using the information of many parameters to refine the node locations. Recently, an approach in [57] has used least square (LS) and maximum likelihood (MLE) algorithms to locate an unknown-coordinate node with references to anchors only.

The mentioned localizations are either "centralized and successive refinement" or "decentralized and non successive refinement." Since successive refinement methods usually give higher accuracy and decentralized methods are more practical, the combination has gained more and more research interest. Patwari and Costa first propose a learning-based algorithm [63], a candidate with the ability to find the eigenvalues and eigenvectors of a sparse and symmetric matrix. Then, they introduce a distributed weighted-multidimentional scaling approach (dwMDS [15]) developed from classical multidimentional scaling (MDS), a centralized and more complex algorithm. In fact, MDS is such a novel tool that many of its variations have been developed for node localization in WSNs. Iterative MDS (IT-MDS) and simulated annealing MDS (SA-MDS) discussed in [8] are two of these variations in which simulated annealing is a famous method imitating the metal cooling process to find the optimal state. Fastmap and MDS are combined in [44] to give a two-stage algorithm in which Fastmap provides the initial coarse input and MDS does the distributed gradient descent for more accuracy like in [15]. Another MDS method based on connectivity only is presented in [38][73][74]. Generally, MDS variations require a big load of computation, especially when the smallest eigenvalues and eigenvectors are used to make MDS work distributedly [20]. They also result in poor accuracy, thus the scalability of the methods is still an issue. A distributed MLE with better accuracy [62] has been introduced, but this increased accuracy occurs only when the algorithm has good first guess input data (raw estimation). In [47], another method uses MDS to calculate the initial data and then uses MLE to produce the refined result of localization, making the computation and communication costs even higher. In this thesis, we introduce a distributed successive localization. At each iteration on a normal node, only few simple computations are needed. Moreover, the communication cost can be reduced remarkably with an insignificant accuracy trade-off as it can be seen in later chapters on analysis and simulations. One remarkable point is that PPE is unlike other methods which base on available algorithms, thus it needs to be analyzed fully. The analyses in later sections prove that PPE can obtain the global convergence even it is performed distributedly. It is a convergent and robust algorithm with any unbiased measurement input. Therefore, it can deal with not only Gaussian model but the log-normal model with a de-biasing function as well. In fact, PPE can deal with all three most popular and widely-used measurement models which are mentioned in details in Section 2.1.5, see Fig.2.8. That means, under the assumption that the pairwise distances are given and unbiased, PPE can be applied with simple calculation and guaranteed convergence.

Related works of acoustic based object positioning

Technically, most current acoustic approaches which utilize passive sensors in WSNs can only deal with one tracked object [2][12][13][37][49][69][80][88][92], and few works have studied multiple object tracking problem [3][40]. Moreover, the mentioned methods are strongly based on the techniques for one target tracking. The common focus of those techniques is on the Direction of Arrival (DoA), or the relative angles between sound sources and sensor arrays [23][26][86]. After that, particle filter [5][86], Kalman filter [30][103] or hidden Markov model [53][101] may be used to enhance the tracked routes of the sources [3][96].

For general method based on DoA schemes, generalized cross correlation would be applied to compute the time delay estimation (TDE). This is the most popular variant for the acoustic source localization and tracking problem where DoA values are estimated based on the sensor arrays. In many cases, the sensor array can have only two microphone sensors and the signal's DoA is computed based on the differences between these pairs of recording microphones. This scheme is actually good only for the scenario in which a single source is monitored by several arrays of sensors, calculation of localization is usually based indirectly on maximizing the cross correlation [92] or directly on Time Differences Of Arrival (TDOA) [49]. In order to compress the reverberation or multipath fading, several TDE works have been proposed and can be found in

[9][25]. Another set of methods related to the direction are relying on beamforming (BF) [13][64]. The microphone sensors of an array are organized and tuned in order to focus on some particular positions in the monitoring area and to measure the averaged power. The directions to the sources are obtained by the detection of maximum power when the BF tries to scan all over the monitoring space (usually 2-D space). In other words, the beamforming is steered electronically by the array of microphone sensors to locate the source's position based on the detected power. The assumption in those methods is that the beamforming power is large at the focused positions. However, the methods usually have local maxima and the result accuracy suffers as a consequence. Except for the BF technique [40][64] used in the passive towed array sonar system, the numbers of tracked sources used in previous works are small, two is common and at most three [3][68]. Towed array sonar system is an array of sensors deployed along a tow which is usually considered to be straight. Based on delay times and linear phase differences recorded at sensors, it gives proper directions of arrival signals if the sources are far from the tow but it gives poor range estimation [27][40]. Such sonar system, after all, is a well developed DoA sensor array. Actually, both mentioned TDE based and BF based methods are just for positioning in which the prediction based on the past and present observations are not considered. To fill in this neglect, particle filters (PF) [95] and Kalman filters (KF) [30][85][103], which provide dynamic models via time steps, have been applied. These two approaches are often used along with some schemes where the positioning task is performed in advance with TDE or BF, then they will enhance the system performance by considering the model of the objects' moving in terms of speed, direction, statistical history of moving, etc. To give a proper name, PF and KF methods can be referred to as tracking stage, which is based on some positioning stage beforehand. The positioning stage is carried out by using such methods as TDOA, TDE, BF, and so forth.

In this work, we do not use the nodes each of which has a sensor array because this kind of node also has the problem when being randomly deployed. Instead we design a node with an isotropic microphone, so randomly deploying sensors is not an issue and the load of sensed data is also reduced. The direction of arrival information is then not the feature for solving the source localization because of the change of the input data. For the framework in which a large number of sources to be located with the isotropic sensors, we would intuitively consider some techniques that can be applied to the problem, including Principal Component Analysis (PCA) and Independent Component Analysis (ICA) techniques [34] which are capable of solving a class of Blind Source Separation (BSS) problems [55]. When the sources are recovered, the different time delays from sources to sensors can be obtained for location estimating. Nevertheless, computing the delay from sources to sensors proves to be difficult since the signal from a source takes different time delays to reach the sensors, so the observed data is a set of convolved mixtures. Although convolved mixture ICAs have been developed to deal with this kind of complex data, their disadvantages include the excessive computation cost since the Finite Impulse Response (FIR) Linear Algebra model must be used [42][79]. The necessity of a centralized manner makes the communication load become so big that it is difficult, if not impossible, to apply convolved mixture ICAs to WSNs. Moreover, related works on blind source separation so far can not deal with the data that has Doppler effect, not to mention the interference of high noise and Rayleigh multi-path fading. The reason is previous works on multiple source positioning so far has just been for stand-still sources [41][46][56][81][93], thus the Doppler effect does not exist in their signal models. Dealing with convolved mixtures under the influence of Doppler effect and Rayleigh multi-path fading for localization, which has not mentioned before, will be discussed in this work.

We have introduced the early version of the system design for source positioning based on ICA technique on the frequency domain in order to eliminate the influence of time delays and to extract the ratios between original source energies for position estimating [18]. Our preliminary method can deal with convolved mixture data [14][79] and avoid many drawbacks of convolved mixture ICAs on both time and frequency domains [69][89][96]. Nevertheless, it can only be used for positioning still sources and does not adequately tolerate the noise. It fails to recover the frequency images, or magnitude spectral images, of moving sources when Doppler effect causes different spectral shifts on the frequency domain especially when Rayleigh multi-path fading is experienced [28]. Using ICA on the data of frequency images, we observe that independent sources allow for replacing the high-cost and low-reliability ICA techniques with clustering methods. The positioning can be solved relying on the information of magnitude ratios each of which is calculated from energies of an f-component at different sensors. Object localization method based on this information has not appeared in any previous works before either and will be presented in the later chapters. For multiple moving source localization, this thesis does not pay attention to tracking stage but to positioning instead. That means the tracking which considers the past and present states, the velocity and the direction of moving to enhance the estimation accuracy is not the focus of this work and will be left as the future improvement work.

Chapter 2

Push-pull Estimation (PPE) for Sensor localization

This chapter describes the idea of Push-pull Estimation (PPE) along with the annotations [17]. Then the analyses of the proposed method are presented to highlight its advantages in terms of distributed ability, scalability, low computation cost, low communication cost, accuracy, etc. The simulation results are in the first Section of Chapter 4 for the purposes of verifying the analyses and comparing the results with those of other contemporary successive refinement approaches.

2.1 Push-pull Estimation (PPE)

2.1.1 Annotation and description of PPE

Our proposed algorithm [45] is based on geometry in which the errors of measurements are modeled into pushing and pulling forces. The influence of these forces leads a node to the point where all of the forces are balanced or, in other words, where the errors are minimized and canceled out through the averaging mechanism. The algorithm is named the Push-pull Estimation because of this original concept. PPE is designed to meet the basic requirement that if the range measurement is shorter than the calculated distance between two nodes, these two nodes pull each other, and vice versa. In addition, the force magnitude should be a monotone function of the discrepancy between the measured and the calculated distances in order for PPE to work and converge. The bigger is the discrepancy, the higher is the influence it gives to reduce itself. This idea is demonstrated in Fig.2.1 where node i needs to update its coordinates to the balanced position. Node j causes a pushing force on i because the range measurement between i and j is longer than the current calculated distance between them, and contrarily, node k exerts a pulling force on i. It should be noted that in the concept of this model, we give normal nodes the moving ability and consider location updating as a process in which normal nodes move under the influence of forces caused by their neighbors. When a normal node attempts to locate its balanced position, it considers the other nodes to be still.

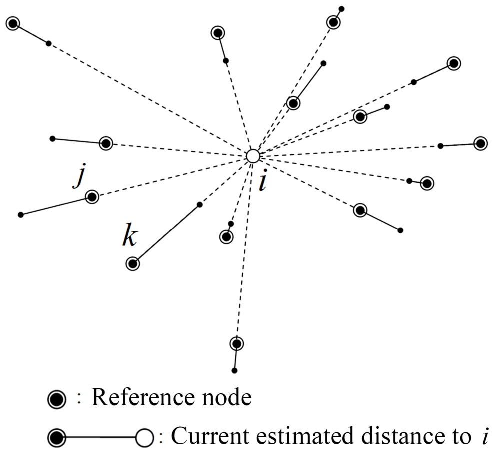
Assuming that the total node number of a 2-D deployed sensor network is N, $N = n + m$ where n is the number of normal nodes and m is the number of anchors, $m << n$. Let $\mathbf{x} = \{x_i : i = 1..N\}$, $x_i \in \mathbf{R}^2$, be the true coordinate vector of normal sensors $\{x_i : i = 1..n\}$ and anchors $\{x_i : i = (n+1)..N\}$. The true distance d_{ij} is the Euclidean distance between i and j,

$$d_{ij} = \|x_j - x_i\| \quad ; i, j = 1..N, \tag{2.1}$$

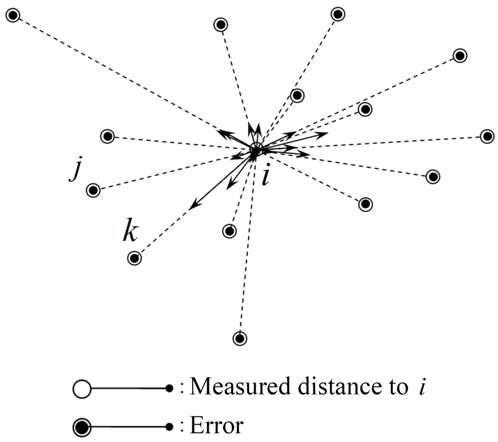
so, $d_{ii} = 0$ and $d_{ij} = d_{ji}$.

We used the range-based scheme for PPE input by assuming that with a certain methodology, unbiased measurements from a node to its neighbors and several anchors are available. This is normal in most range-based localized WSNs. The ranges can also be estimated through hops [58][90], and the anchors may have a large broadcasting range. Then the pair-wise measurement or the measured distance between node i and j is $\delta_{ij} : i, j = 1..N$; $\delta_{ij} = \delta_{ji}$, $\delta_{ii} = 0$ and $\delta_{ij} = 0$ if the range from i to j is too large, or out of range. Since the error of measurement is always proportional to the real distance, δ_{ij} has the following form [44]

$$\delta_{ij} = d_{ij} + d_{ij}.noise_{ij} \quad , i, j = i..N \tag{2.2}$$



(a)



(b)

Figure 2.1: (a) Node i and measurement errors from i to its reference nodes. (b) The errors are modeled as pull-push forces.

where $noise_{ij} \sim \mathcal{K}\left(0, \sigma_n^2\right)$, some zero-mean distribution. We also define the current estimated coordinate vector to be $\tilde{\mathbf{x}} = \{\tilde{x}_i : i = 1..N\}$, $\tilde{x}_i \in \mathbf{R}^2$, certainly we always have $\tilde{x}_j = x_j, j = (n+1)..N$. The localization problem can be succinctly stated: the input data of the algorithm are measurements δ_{ij} and anchors' locations $\{x_j : j = (n+1)..N\}$; the algorithm will give an initial $\tilde{\mathbf{x}}$ and will update each $\tilde{x}_i$ of $\tilde{\mathbf{x}}$ so that the difference between $\tilde{\mathbf{x}}$ and $\mathbf{x}$ is as small as possible, $\tilde{\mathbf{x}}$ then is the solution to the problem. Let the current calculated pair-wise ranges be $\tilde{d}_{ij}$, then

$$\tilde{d}_{ij} = \|\tilde{x}_j - \tilde{x}_i\| \quad , i, j = 1..N. \tag{2.3}$$

Now the distributed PPE algorithm for a normal node i is presented with three phases. In each phase node i uses iterations to produce a balanced position by updating position $\tilde{x}_i$ with the two following equations:

$$\overrightarrow{F_i}^{(p)} = \frac{1}{M_i^{(p)}} \sum \overrightarrow{f_{ij}}^{(p)}, \tag{2.4}$$

$$\tilde{x}_i \leftarrow \tilde{x}_i + \alpha^{(p)} \overrightarrow{F_i}^{(p)}. \tag{2.5}$$

$\alpha^{(p)}$ in (2.5) is the movement rate and $\overrightarrow{F_i}^{(p)}$ is the mean-force caused by a set of $M_i^{(p)}$ related nodes. These parameters vary in different phase p of PPE.

- Phase 1: Raw estimation:

Node i needs at least three different beacon positions and the measurements from itself to these beacons. By computing (2.4) and (2.5), i gets its raw estimated location where the sum of forces caused by the related beacons is balanced. At each iteration, these forces are

$$\overrightarrow{f_{ij}}^{(1)} = \left(\tilde{d}_{ij} - \delta_{ij}\right) \overrightarrow{e_{ij}}, \tag{2.6}$$

where j is the index of beacons related to i, and $\overrightarrow{e_{ij}}$ is the unit vector pointing from $\tilde{x}_i$ to $\tilde{x}_j$ to indicate the direction of $\overrightarrow{f_{ij}}$. The initial position of $\tilde{x}_i$ is chosen as the mean position of these beacons. Choosing the initial position is empirical and will be mentioned later in Section 4.1.1. $M_i^{(1)}$ is the number of beacons related to i, see Fig.2.2.

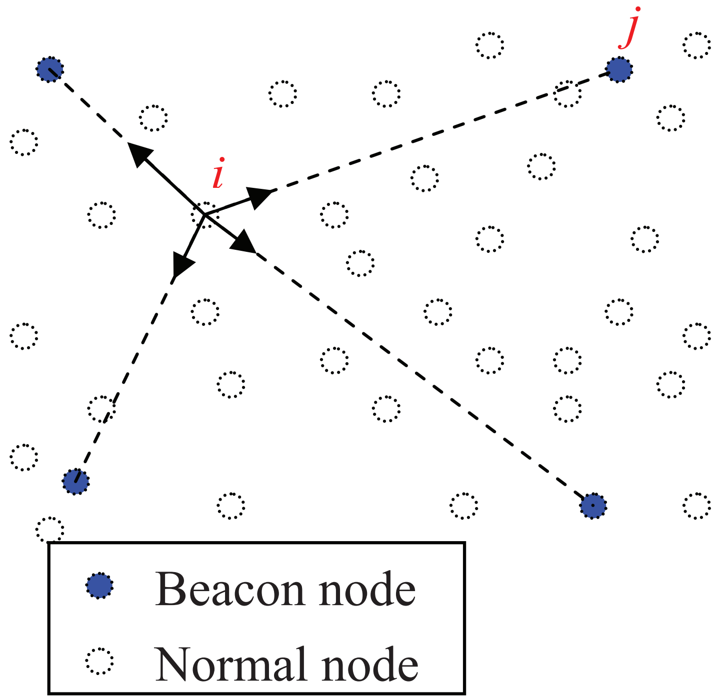


Figure 2.2: Phase1 reference nodes are the beacons that node i refers to

- Phase 2: Pre-refinement:

After normal nodes complete the raw estimation phase, node i takes the measurements to all of its related nodes, consisting of related neighbors and beacons (see Fig.2.3), and their current updated positions for determining its new balanced position. For normal neighbor j, $\tilde{x}_j$ is the current estimated position and for related beacon j, it is the actual position. At each round the individual forces are

$$\overrightarrow{f_{ij}}^{(2)} = \left(\tilde{d}_{ij} - \delta_{ij}\right)\overrightarrow{e_{ij}}. \tag{2.7}$$

For both phase 2 and phase 3, after a certain number of iterations, node i must send out its current updated position to its normal neighbors and get the new updated positions of the neighbors for better location computing.

- Phase 3: Refinement:

This phase is used last to improve result accuracy because the error of measured distance is proportional to the actual distance. It has the same behavior as phase 2 with the same related nodes, or $M_i^{(3)} = M_i^{(2)}$. The difference is the way node i calculates the forces caused by all of its related nodes. $\overrightarrow{f_{ij}}^{(3)}$ is obtained by dividing $\overrightarrow{f_{ij}}^{(2)}$ by $\tilde{d}_{ij}$,

$$\overrightarrow{f_{ij}}^{(3)} = \left(\frac{\tilde{d}_{ij} - \delta_{ij}}{\tilde{d}_{ij}}\right)\overrightarrow{e_{ij}} = \left(1 - \frac{\delta_{ij}}{\tilde{d}_{ij}}\right)\overrightarrow{e_{ij}}. \tag{2.8}$$

j in phase 3 is the same as that in phase 2, the index of all nodes related to i. The stop condition for all three phases is when the magnitude of the mean-force on i is less than some small positive value or when a given maximum number of iterations is reached. Note that $\tilde{d}_{ij} \neq 0$ in all three phases, $\tilde{d}_{ij} = 0$ means that i does not have the measured distance to j and so will not use j as a related node for updating its position.

For convenience of description and further analysis, we add the following common formula

$$\overrightarrow{f_{ij}^{(p)}} = \left(\frac{\tilde{d}_{ij} - \delta_{ij}}{L_{ij}}\right)\overrightarrow{e_{ij}}, \tag{2.9}$$

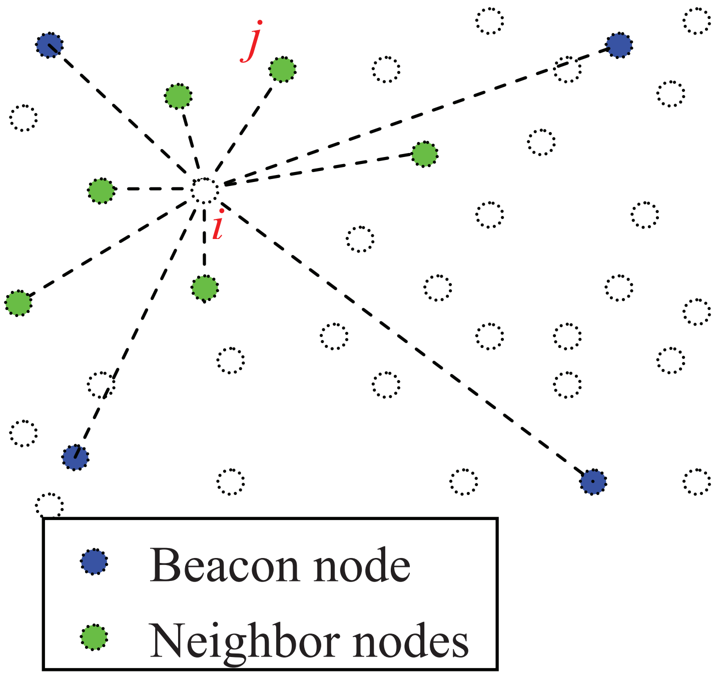


Figure 2.3: In phase 2 and phase 3, a node has the same reference nodes which are the beacons and the neighbors that it refers to

where

$$L_{ij} = \begin{cases} 1 & \text{for phase 1 and phase 2} \\ \tilde{d}_{ij} & \text{for phase 3} \end{cases} . \tag{2.10}$$

Clearly, if $\delta_{ij} < \tilde{d}_{ij}$, then $\overrightarrow{f_{ij}}$, having the same direction with vector $\overrightarrow{e_{ij}}$, will be a pulling force given by node j on node i; and vice-versa, if $\delta_{ij} > \tilde{d}_{ij}$, $\overrightarrow{f_{ij}}$ will be a pushing force. If $\delta_{ij} = \tilde{d}_{ij}$, the force of j on i is zero.

This force model is the core idea of the algorithm. When a normal node, say node i, is at a position where the forces around it are not balanced, i will move under the force influence to a new position where the sum-force tends to be weaker. While i is moving, at any new position, the sum-force changes in both of magnitude and direction, so the path of movement is actually a curve. Since we can not make a continuous curve, we will use step-wise movements over iterations to update the estimation. In addition, $\alpha^{(p)}\overrightarrow{F_i}^{(p)}$ must be a quantity in the same unit as that of d_{ij} or δ_{ij}. We can define $\alpha^{(p)}$ with a more specific formula

$$\begin{cases} \alpha^{(1)} = \alpha_1 \\ \alpha^{(2)} = \alpha_2 \\ \alpha^{(3)} = \alpha_3 \overline{\tilde{d}_{ij}} \end{cases} , \tag{2.11}$$

where α_i is a non-unit quantity and is small enough to guarantee convergence; and $\overline{\tilde{d}_{ij}}$ is the mean value of the distances measured from i to its neighbors and can be replaced by other parameters provided that the replaced ones have the same unit of length. Determining the value of $\alpha^{(p)}$ so that the step is small enough to guarantee convergence is difficult. If $\alpha^{(p)}$ is too small, too many iterations are needed. If it is too large, the algorithm may not converge. There are a few approaches for adjusting $\alpha^{(p)}$ to overcome this dilemma. For consistency with the goal of low cost, we choose the *line search* technique by doubling $\alpha^{(p)}$ until $\left\|\overrightarrow{F_i}^{(p)}\right\|$ starts to increase.

For evaluating the accuracy of PPE in later sections, we use the root mean square of the position errors.

$$RMS = \sqrt{\frac{1}{n}\sum_{k} \|\tilde{x}_k - x_k\|^2}, \tag{2.12}$$

where k is the index for a normal node.

2.1.2 PPE's convergence analysis

We analyze PPE's convergence by proving the following statements:

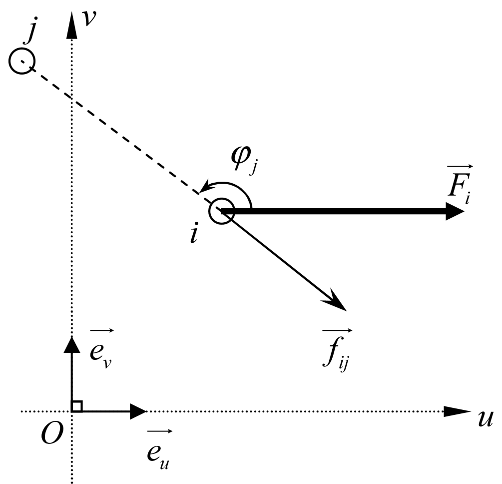
i) "In all three phases, a node will always find a balanced position via the force mechanism."

This statement implies that the force described in (2.9) converges the location of a normal node i to a balanced point, and therefore i would never get stuck in infinite loop. In this part, we try to prove that when a node moves along the mean force influencing on it, the mean force magnitude will decrease. Therefore, it finally gets to a position where the mean force is zero, or the position where the forces around make the position a balanced point. We decompose the mean-force $\overrightarrow{F_i}$ ($\overrightarrow{F_i} = \overrightarrow{F_i^{(1,2)}}$) into two orthogonal sub-components $\overrightarrow{F_u}$ and $\overrightarrow{F_v}$ in a coordinate system uOv in which $\overrightarrow{e_u}$ and $\overrightarrow{e_v}$ are the basis vectors. In order to simplify the analysis, we choose $\left(\widehat{\overrightarrow{F_i}, \overrightarrow{Ou}}\right) = 0$ and let $\varphi_j = \left(\widehat{\overrightarrow{F_i}, \overrightarrow{ij}}\right)$ (see Fig.2.4). For phase 1 and phase 2, the mean-force is rewritten:

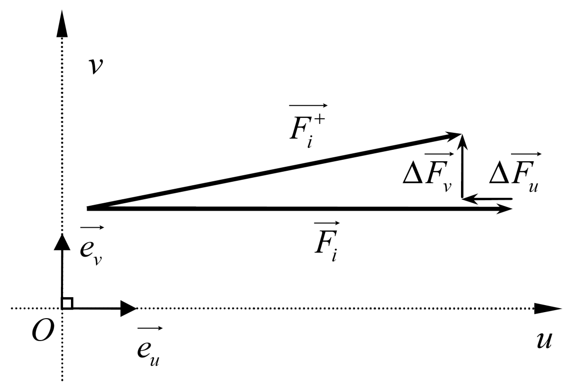
$$\overrightarrow{F_i} = \frac{1}{M_i^{(p)}} \sum_j \left(\tilde{d}_{ij} - \delta_{ij}\right)\overrightarrow{e_{ij}}, \tag{2.13}$$

$$\overrightarrow{F_i} = \overrightarrow{F_u} + \overrightarrow{F_v}, \tag{2.14}$$

$$\begin{aligned} \overrightarrow{F_i} = {} & \frac{\overrightarrow{e_u}}{M_i^{(p)}} \sum_j \left(\begin{array}{c} \left(\sqrt{(u_i - u_j)^2 + (v_i - v_j)^2} - \delta_{ij}\right) \times \\ \cos\varphi_j \end{array} \right) \\ & + \frac{\overrightarrow{e_v}}{M_i^{(p)}} \sum_j \left(\begin{array}{c} \left(\sqrt{(u_i - u_j)^2 + (v_i - v_j)^2} - \delta_{ij}\right) \times \\ \sin\varphi_j \end{array} \right). \end{aligned} \tag{2.15}$$



(a)



(b)

Figure 2.4: (a) The mean-force in coordinate system uOv. (b) The mean-force decreases in magnitude when a node moves along it.

Now we analyze how the mean-force changes when i moves along $\overrightarrow{F_i}$'s direction such a small distance that φ_j is considered to be unchanged. It should be noted that although the v-component $\overrightarrow{F_v} = 0$, when i moves along $\overrightarrow{F_i}$, $\overrightarrow{F_i}$ becomes $\overrightarrow{F_i^+}$, and the v-component of $\overrightarrow{F_i^+}$ is not zero. Partial derivation of equation (2.15) reveals

$$\begin{aligned}\frac{\partial \overrightarrow{F_i}}{\partial u_i} =& \frac{\overrightarrow{e_u}}{M_i^{(p)}} \sum_j \left(\frac{(u_i - u_j)\cos\varphi_j}{\sqrt{(u_i-u_j)^2 + (v_i - v_j)^2}} \right) \\ &+ \frac{\overrightarrow{e_v}}{M_i^{(p)}} \sum_j \left(\frac{(u_i - u_j)\sin\varphi_j}{\sqrt{(u_i-u_j)^2 + (v_i - v_j)^2}} \right)\end{aligned}, \tag{2.16}$$

$$\begin{aligned}\frac{\partial \overrightarrow{F_i}}{\partial u_i} =& \left(\frac{1}{M_i^{(p)}} \sum_j -\cos^2\varphi_j \right) \overrightarrow{e_u} \\ &+ \left(\frac{1}{M_i^{(p)}} \sum_j -\cos\varphi_j \sin\varphi_j \right) \overrightarrow{e_v}.\end{aligned} \tag{2.17}$$

Equation (2.17) means that when i moves along $\overrightarrow{F_i}$ a very small distance Δu ($\Delta u > 0$), $\overrightarrow{F_i}$ changes by an amount of $\Delta\overrightarrow{F_i}$, which is composed by two orthogonal components $\Delta\overrightarrow{F_u}$ and $\Delta\overrightarrow{F_v}$ (see Fig.2.4),

$$\Delta\overrightarrow{F_i} = \Delta\overrightarrow{F_u} + \Delta\overrightarrow{F_v}. \tag{2.18}$$

$$\begin{aligned}\Delta\overrightarrow{F_i} =& \left(\frac{1}{M_i^{(p)}} \sum_j -\cos^2\varphi_j \right) \Delta u_i \overrightarrow{e_u} \\ &+ \left(\frac{1}{M_i^{(p)}} \sum_j -\cos\varphi_j \sin\varphi_j \right) \Delta u_i \overrightarrow{e_v}.\end{aligned} \tag{2.19}$$

The new updated mean-force on i becomes

$$\overrightarrow{F_i^+} = \overrightarrow{F_i} + \Delta\overrightarrow{F_i}. \tag{2.20}$$

Because of the fact that $\sum_j -\cos^2\varphi_j < 0$, the inner product $\left\langle \Delta\overrightarrow{F_u}, \overrightarrow{e_u} \right\rangle < 0$, or $\Delta\overrightarrow{F_u}$ always has a counter-direction to $\overrightarrow{F_i}$. In addition, $\left\|\Delta\overrightarrow{F_v}\right\| \ll \left\|\overrightarrow{F_i}\right\|$. As a result, for phase 1 and phase 2 (see Fig.2.4b):

$$\left\|\overrightarrow{F_i^+}\right\| \approx \left\|\overrightarrow{F_i} + \Delta\overrightarrow{F_u}\right\| < \left\|\overrightarrow{F_i}\right\|. \tag{2.21}$$

In the same manner, in phase 3, we infer the change amount of $\overrightarrow{F_i^{(3)}}$,

$$\begin{aligned} \frac{\partial\overrightarrow{F_i^{(3)}}}{\partial u_i} &= \frac{1}{2M_i^{(3)}}\left(\sum_j\left(-\frac{\delta_{ij}}{\tilde{d}_{ij}^2}\right)\cos^2\varphi_j\right)\overrightarrow{e_u} \\ &+ \frac{1}{2M_i^{(3)}}\left(\sum_j\left(-\frac{\delta_{ij}}{\tilde{d}_{ij}^2}\right)\sin\varphi_j\cos\varphi_j\right)\overrightarrow{e_v}. \end{aligned} \tag{2.22}$$

The u-component $\Delta\overrightarrow{F_u^{(3)}}$ is also opposite in direction with $\overrightarrow{F_i^{(3)}}$. Therefore, (2.21) holds for all three phases of PPE. This means that when i moves under PPE's force effect, the mean-force's magnitude gets smaller and smaller. As a result, i's location eventually converges to a balanced position.

One can easily see that PPE is basically an optimization method for a normal node in which the objective function to minimize is the length of the mean-force,

$$\tilde{x}_i = \arg\min_{\tilde{x}_i}\left\|\overrightarrow{F_i}^{(p)}\right\|. \tag{2.23}$$

Aiming at a low-cost algorithm, we have avoided much of the complicated load of the gradient descend method which provides the optimal direction of movement. By letting a normal node gradually move along the mean-force, our method is simpler and involves only few simple computations. The proof above guarantees the minimization.

ii) "Estimation of a normal node in phase 1 is unbiased."

In this phase, the reference nodes of a normal node i are the anchors. Since $E\{\delta_{ij}\} = d_{ij}$, the expectation of the mean-force is

$$E\left\{\overrightarrow{F_i^{(1)}}\right\} = \frac{1}{M_i^{(1)}} \sum_j \left(\tilde{d}_{ij} - d_{ij}\right)\overrightarrow{e_{ij}} \tag{2.24}$$

where j is the index of the known-location nodes. This equation implies that the expectation of the mean-force is built upon the correct measurements from the known-location nodes. Each individual force $\overrightarrow{f_{ij}}$ tries to move i to the circle C_{ij}, having the radius d_{ij}, to minimize its own magnitude (see Fig.2.5). As the result, i's position converges to the intersection point i_0 of these circles C_{ij}, or

$$E\{\tilde{x}_i\} = x_i. \tag{2.25}$$

iii) "Estimations of a normal node in phase 2 and phase 3 are unbiased."

In these two phases, every normal node has its raw location estimation and uses all of its reference nodes to better the estimation. Since the algorithm mainly deals with probability issues, we not only consider the force effects to be specifically caused by the nodes but also consider the force effects caused by sub-areas. We assume that the whole deployed area is divided into small sub-areas, in each of which the node number is large and the node distribution is uniform. Fig.2.6 illustrates this idea: the actual positions are the black round nodes, e.g., k_0, while the current estimated positions are the smaller black square nodes, e.g., k. The solid links between them are the current errors of estimations. What we want to do here is to prove that even with these errors of estimations, the balanced position of node i under PPE mechanism is its actual position i_0.

Let the mean force on i caused by the area S_K be

$$\overrightarrow{F_{iK}} = \frac{1}{n_k} \sum_k \left(\frac{\tilde{d}_{ik} - \delta_{ik}}{L_{ik}}\right) \frac{\overrightarrow{ik}}{\tilde{d}_{ik}}. \tag{2.26}$$

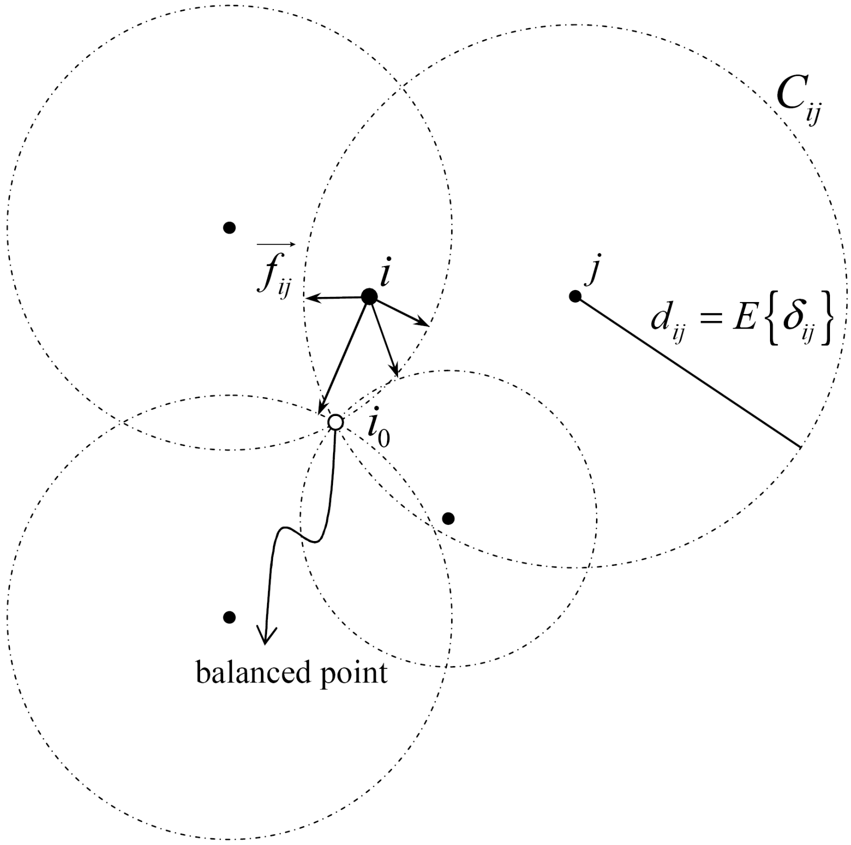


Figure 2.5: When the measurements have no error and the reference nodes' locations are known, the balanced point of a node is its actual position.

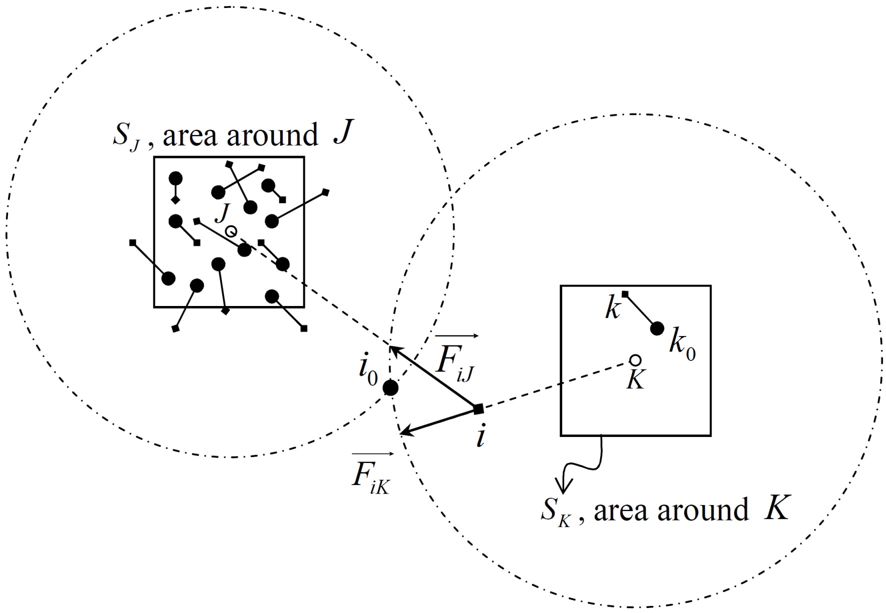


Figure 2.6: The force effect of a sub-area on a node is equivalently the force effect caused by the center position of this sub-area with the correct measurement.

$$E\left\{\overrightarrow{F_{iK}}\right\} = E\left\{\left(\frac{\tilde{d}_{ik} - d_{ik} - d_{ik}.noise_{ik}}{L_{ik}}\right)\frac{\overrightarrow{iK} + \overrightarrow{Kk}}{\tilde{d}_{ik}}\right\}. \tag{2.27}$$

K is the center point of S_K, and n_k in (2.26) is the number of nodes in S_K. In (2.27), $noise_{ik}$ and $\overrightarrow{Kk}$ are independent of each other and of d_{ij}, $\tilde{d}_{ij}$. $E\left\{\overrightarrow{Kk}\right\} = 0$ since the estimation of the previous phase is unbiased. Equation (2.27) is rewritten as

$$E\left\{\overrightarrow{F_{iK}}\right\} = E\left\{\left(\frac{\tilde{d}_{ik} - d_{ik}}{\tilde{d}_{ik}L_{ik}}\right)\right\}\overrightarrow{iK} \tag{2.28}$$

$$E\left\{\overrightarrow{F_{iK}}\right\} = E\left\{\left(\frac{ik - i_0k_0}{\tilde{d}_{ik}L_{ik}}\right)\right\}\overrightarrow{iK}. \tag{2.29}$$

Assume that the estimation errors of nodes in S_k are small so that k still has a uniform distribution in S_K, or $E\{ik\} = E\{ik_0\}$.

- If $iK = i_0K$, then $E\{ik_0 - i_0k_0\} = 0$. Since $\tilde{d}_{ik}L_{ik}$ is a positive and bounded value, $E\left\{\overrightarrow{F_{iK}}\right\} = 0$.

- If $iK > i_0K$, then $E\{ik_0 - i_0k_0\} > 0$, and $E\left\{\overrightarrow{F_{iK}}\right\}$ becomes a pull force with the direction of $\overrightarrow{iK}$.

- Conversely, if $iK > i_0K$, then $E\left\{\overrightarrow{F_{iK}}\right\}$ becomes a push force. Hence, the balanced point of i belongs to the circle $C(K, i_0K)$.

Combining the effects of other sub-areas on i, we find that i_0 becomes the unique balanced position of i. With the previous proof that PPE leads i to a balanced point, we can conclude that the estimations of i in phase 2 and phase 3 are unbiased.

In these two phases, a normal node acts as if it is affected by forces caused by groups of reference nodes, where each group plays the role of a beacon. With more reference beacons, phase 2 has a better accuracy of estimation than does phase 1 (see Section 2.1.3). In addition, it should be noted that the assumption that k has a uniform distribution in S_K is not a tight one. However, through iterations, each node updates and achieves a better and better estimation, so the assumption becomes tighter and tighter. Consequently, the expectation of the balanced point of a normal node i is the very i_0, and PPE drags i toward i_0 even when the current estimations of the reference nodes are experiencing errors. In other words, one node's updating is independent of the others' and global convergence is obtained. This means that the distributed PPE is a robust convergent algorithm.

2.1.3 Necessity of PPE's phases

Raw estimation, i.e. initial input data, is the same requirement for most successive localizations, unless the Procrustes algorithm is used [44][47]. Procrustes is the method used in the last module of localization to fix a flipped or rotated result. However, it usually needs all of the node location estimations, leading to a centralized mechanism and the associated problems, which all distributed approaches aim to avoid. The raw estimation can be any one of the low cost algorithms so far. However, multilateral localizations have either complicated calculation (localizing the most likely area) or a high number of anchors [51][63][65][76][87][94]. Contemporary successive algorithms are surely more complex; in [47], MDS is used to obtain this raw estimation. Clearly, PPE's phase 1 plays an important role for the later phases based only on simple computations. Moreover, by using phase 1, we can utilize the same hardware and software modules because the structures of all of three phases are similar. This makes the method more practical.

In all three phases, if the reference node number is infinitely large, the node's balanced position will be at its actual position. Unfortunately, the limited reference node number $M_i^{(p)}$ and the errors of measurements make the mean-force non-zero at the correct position i_0, drive i out of i_0, and rise PPE's error (see Fig.2.7a). Obviously, the larger is the variance of the mean-force's magnitude, the larger is the error in the final result. Therefore, to compare the performances of the three phases, we examine the variance of $\left\|\overrightarrow{F_i^{(p)}}\right\|$ at position i_0 when all of the reference nodes are at the correct positions.

Let $\left\|\overrightarrow{F_i^{(p)}}\right\| = F_i^{(p)}$, $\left\|\overrightarrow{f_{ij}^{(p)}}\right\| = f_{ij}^{(p)}$ and decompose each individual force $\overrightarrow{f_{ij}^{(p)}}$ into components as in Fig.2.7b, so $\overrightarrow{f_{ij}^{(p)}} = \left(f_{ij}^{(p)}\cos\theta_j, f_{ij}^{(p)}\sin\theta_j\right)$. Remarking that $E\left\{f_{ij}^{(p)}\right\} = 0$, $E\left\{F_i^{(p)}\right\} =$

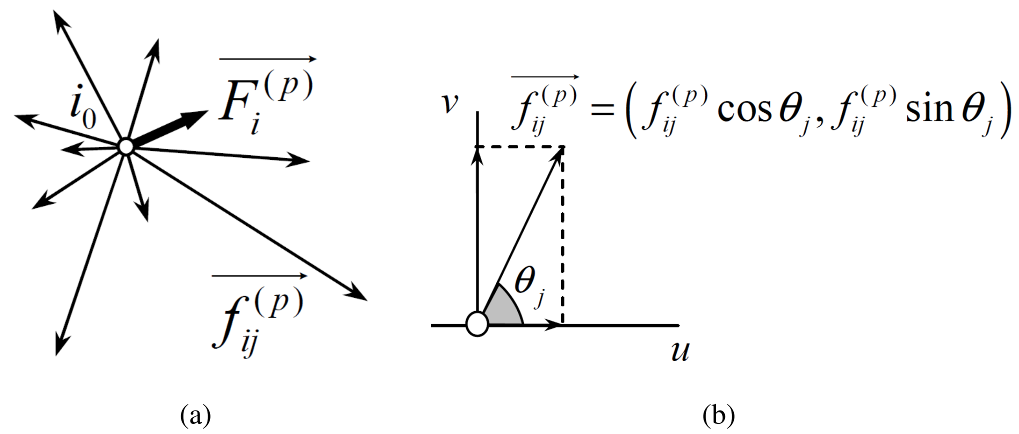


Figure 2.7: (a) The mean-force is not a zero vector at the correct position because measurements have errors and the number of reference nodes is limited. (b) A force is decomposed into horizontal and vertical components.

0, (unbiased measurement error and unbiased estimation respectively), and $f_{ij}^{(p)}$ and θ_j are two independent variables, we have

$$\overrightarrow{F_i^{(p)}} = \frac{1}{M_i^{(p)}} \sum_j \overrightarrow{f_j}, \tag{2.30}$$

$$\overrightarrow{F_i^{(p)}} = \left(\frac{1}{M_i^{(p)}} \sum_j f_{ij}^{(p)} \cos\theta_j, \frac{1}{M_i^{(p)}} \sum_j f_{ij}^{(p)} \sin\theta_j \right) \tag{2.31}$$

$$\mathrm{var}\left(F_i^{(p)}\right) = E\left\{ \left(F_i^{(p)}\right)^2 \right\} - E^2\left\{F_i^{(p)}\right\} \tag{2.32}$$

$$\begin{aligned} &\mathrm{var}\left(F_i^{(p)}\right) = \\ &E\left\{ \left(\frac{1}{M_i^{(p)}} \sum_j f_{ij}^{(p)} \cos\theta_j \right)^2 + \left(\frac{1}{M_i^{(p)}} \sum_j f_{ij}^{(p)} \sin\theta_j \right)^2 \right\} \end{aligned} \tag{2.33}$$

$$\text{var}\left(F_i^{(p)}\right) = \frac{\text{var}\left(f_{ij}^{(p)}\right)}{M_i^{(p)}}. \tag{2.34}$$

Equation (2.34) implies that the variance of $F_i^{(p)}$ does not depend on the directions of the forces caused by neighbors, but only on the reference node number and the variance of the individual force. It also guarantees that phase 2 is necessary to improve phase 1's result, which is obtained only with anchors ($M_i^{(1)} < M_i^{(2)}$).

In order to show the necessity of phase 3 over phase 2, we compare the variances of the individual forces in these two phases because the number of reference nodes in these phases are the same, $M_i^{(2)} = M_i^{(3)}$. From (2.4) and (2.11), we have

$$\begin{cases} \text{var}\left(f_j^{(2)}\right) = \text{var}\left(\alpha_2\left(\delta_{ij} - d_{ij}\right)\right) = \text{var}\left(\alpha_2 d_{ij}.noise_{ij}\right) \\ \text{var}\left(f_j^{(3)}\right) = \text{var}\left(\alpha_3 \overline{d_{ij}}.\frac{\delta_{ij} - d_{ij}}{d_{ij}}\right) = \text{var}\left(\alpha_3 \overline{d_{ij}}.noise_{ij}\right) \end{cases} \tag{2.35}$$

Obviously, in each phase, the balanced position of a node is the same with any α_i provided that α_i is small enough so that PPE can obtain convergence. Therefore the comparison is meaningful only when the forces have equivalent magnitudes, or

$$\sum_j \alpha_2 d_{ij}.noise_{ij} \approx \sum_j \alpha_3 \overline{d_{ij}}.noise_{ij} \Leftrightarrow \alpha_2 \approx \alpha_3. \tag{2.36}$$

Then,

$$\begin{aligned} \text{var}\left(f_j^{(2)}\right) &= \alpha_2^2\left[E\left\{\left(d_{ij}.noise_{ij}\right)^2\right\} - E^2\left\{d_{ij}.noise_{ij}\right\}\right] \\ &= \alpha_2^2\left[E\left\{d_{ij}^2\right\} E\left\{\left(noise_{ij}\right)^2\right\} - 0\right] \end{aligned} \tag{2.37}$$

$$\begin{aligned} &\Leftrightarrow \text{var}\left(f_j^{(2)}\right) = \alpha_2^2\left(\text{var}\left(d_{ij}\right) + E^2\left\{d_{ij}\right\}\right)\text{var}(noise_{ij}) = \\ &= \alpha_2^2 E^2\left\{d_{ij}\right\}\text{var}\left(noise_{ij}\right) + \alpha_2^2 \text{var}\left(d_{ij}\right).\text{var}\left(noise_{ij}\right)\ . \end{aligned} \tag{2.38}$$

Meanwhile,

$$\text{var}\left(f_j^{(3)}\right) = \alpha_2^2 \overline{d_{ij}}^2 \text{var}\left(noise_{ij}\right) = \alpha_2^2 E^2\left\{d_{ij}\right\}\text{var}\left(noise_{ij}\right). \tag{2.39}$$

Equations (2.38) and (2.39) lead to

$$\mathrm{var}\left(f_j^{(3)}\right) < \mathrm{var}\left(f_j^{(2)}\right). \tag{2.40}$$

The variance of the mean-force is reduced in the refinement phase. Consequently, the result of estimation in phase 3 is better than that in phase 2 when the measurement error is proportional to the real distance. Confirmations are conducted via simulation in Section 4.1.3.

2.1.4 Distributed ability and scalability

In any of the three mentioned phases, PPE works distributedly in the way that the computation load of localization is divided into sub-tasks performed by the normal nodes of the network. Statements proved in Section 2.1.2 assure the convergence of the whole result, even if the reference positions are not correct. Hence, normal nodes do not require synchronized updating while the localization is being performed. In other words, PPE is a distributed algorithm.

One important point is that the lack of minor information can be accepted by PPE. Removed or added nodes will not affect much a node's position updating as long as the number of these nodes is small compared to the number of reference neighbors. The reason is that each individual force contributes to the mean-force an amount proportional to ${}^{1}\big/_{M_i^{(p)}}$. Some nodes, including newly added ones, even do not need to obtain the distances to the beacons beforehand because they can use the estimated positions of the reference neighbors. Besides, each node only uses the information of its nearest anchors and neighbors, so the scalability of a network using PPE is assured.

In any distributed successive refinement localizations, a node needs to know its neighbors' current positions in order to obtain a higher estimation at every iteration. The communication cost would significantly increase, whereas network designers continually try to reduce and replace it with computation cost. Fortunately, PPE is a good way to soothe the burden of communication cost. Proven statements guarantee that a node's updated result is not affected much by those of others. Hence, it is unnecessary for a node to send out requests to its neighbors for reference updated positions after every iteration. Instead, the communication cost can be reduced when a normal node sends the request task once after every R internal iterations on it, or in other words, the task is skipped $R-1$ times. We call R the *cycle request number*. Consequently, the communication cost decreases by R-fold.

$$PPE\_Communication\_Cost = \frac{C.Cost}{R}, \tag{2.41}$$

where $C.Cost$ is the total communication cost of a net work when $R=1$. This again verifies PPE's distributed ability and scalability for practical implementation with light computation and communication. A simulation set in Section 4.1.1 shows how little a node's iteration result depends on those of the others; and another in Section 4.1.3 illustrates the slight trade-off of accuracy for significant communication cost.

2.1.5 De-biasing function and RSS scheme

The noise model in this work does not have to be a specific one which can be represented by a mathematical formula. The only constraint of the noise model is a zero-mean distribution or unbiased measurements.

Suppose we have a function $g\left(.\right)$ that changes the distribution of variable δ_{ij} into an unbiased distribution of variable $g\left(\delta_{ij}\right)$ and $g\left(.\right)$ is a monotone increasing function so that PPE can correctly define the direction of the force, the formula (2.6), (2.7), and (2.8) now can be replaced by

$$\overrightarrow{f_{ij}}^{(1)} = \overrightarrow{f_{ij}}^{(2)} = \left(g\left(\delta_{ij}\right) - g\left(\tilde{d}_{ij}\right)\right)\overrightarrow{e_{ij}} \tag{2.42}$$

and

$$\overrightarrow{f_{ij}}^{(3)} = \left(\frac{g\left(\delta_{ij}\right) - g\left(\tilde{d}_{ij}\right)}{g\left(\tilde{d}_{ij}\right)}\right)\overrightarrow{e_{ij}}\ . \tag{2.43}$$

Phase 3 with (2.43) is necessary only when the error of $g\left(\delta_{ij}\right)$ is proportional to $g\left(d_{ij}\right)$. In reality, RF power decays proportionally to a negative exponent of the distance. As a result, the power error measured at a receiver is modeled to the log-normal distribution, $P_{ij} \sim N\left(E\left\{P_{ij}\right\}, \sigma_p^2\right)$. This log-normal is proven to be suitable for modeling the decaying power of RF signals in both practice and literature [67]. We will find the de-biasing function $g\left(.\right)$ for this case:

$$P_{ij} = P_0 - 10\log_{10}\left(\frac{\delta_{ij}}{d_0}\right), \tag{2.44}$$

$$P_{ij} = E\left\{P_{ij}\right\} + Pnoise. \tag{2.45}$$

P_{ij} is the received power in decibels at node i when node j transmits to i and $Pnoise \sim N\left(0, \sigma_p^2\right)$ while P_0 is the received power, also in decibels, from a node at a reference distance d_0. In this thesis, we do not focus on the physical parameter channel-loss, which is included in the Gaussian variable $Pnoise$ (see Appendix of [44]), and just use $Pnoise$ for the simulation of the log-normal measurement model. Rewrite (2.45):

$$10\log_{10}\left(\delta_{ij}\right) = 10\log_{10}\left(d_{ij}\right) - Pnoise. \tag{2.46}$$

Even though δ_{ij} does not have an unbiased distribution over d_{ij}, $10\log_{10}\left(\delta_{ij}\right)$ does over $10\log_{10}\left(d_{ij}\right)$. Therefore the de-biasing function in PPE to deal with the log-normal distribution is $g\left(d\right) = 10\log_{10}\left(d\right)$. Actually, the function $g\left(d\right)$ does not need to be exactly $10\log_{10}\left(d\right)$, but any of the log variations providing that $g\left(d\right)$ is a monotone increasing function and

$$g\left(\delta_{ij}\right) = g\left(d_{ij}\right) + t.N\left(0, \sigma_p^2\right), t \neq 0. \tag{2.47}$$

Then,

$$g\left(d\right) = a\log_{10}\left(d\right), \quad a \in \mathbf{R}^+. \tag{2.48}$$

For a log-normal RSS scheme, the measurement model is (from (2.46))

$$\delta_{ij} = d_{ij} 10^{Pnoise}/10, \tag{2.49}$$

and only the first two phases with (2.42) are used.

In general, PPE is a geometry based method for sensor localization. The measurement model can be of any providing that it is unbiased. When it is not, if a de-biasing function is derived, then PPE can be used properly. For three most usual situations of measurement model depicted in Fig.2.8, PPE gives good performance as one can see in the experiments in later chapter. For the first model, $\delta_{ij} = d_{ij} + noise_{ij}$, the first two phases are used. For the second model, $\delta_{ij} = d_{ij} + d_{ij}.noise_{ij}$, all three phases are used. For the log-normal distribution of the third model, the first two phases are used with the de-biasing function defined in (2.48).

2.1.6 Details of PPE

As a consequence of all PPE's characteristics and the related analysis, PPE for a normal node i now can be described in details as in Fig.2.9 and in the algorithm flows. In this description, $\overrightarrow{f_{ij}}^{(p)}$ is defined in (2.42); $g(.)$ is the de-biasing function. We choose $g(d) = d$ if the distance measurement is unbiased (both $\delta_{ij} = d_{ij} + noise_{ij}$ and $\delta_{ij} = d_{ij} + d_{ij}.noise_{ij}$) and (2.43) and $g(d) = \log_{10} d$ if it has a log-normal distribution ($\delta_{ij} = d_{ij} 10^{Pnoise}/10$,). R is the cycle request number.

Followings are the algorithm details of all three phases:

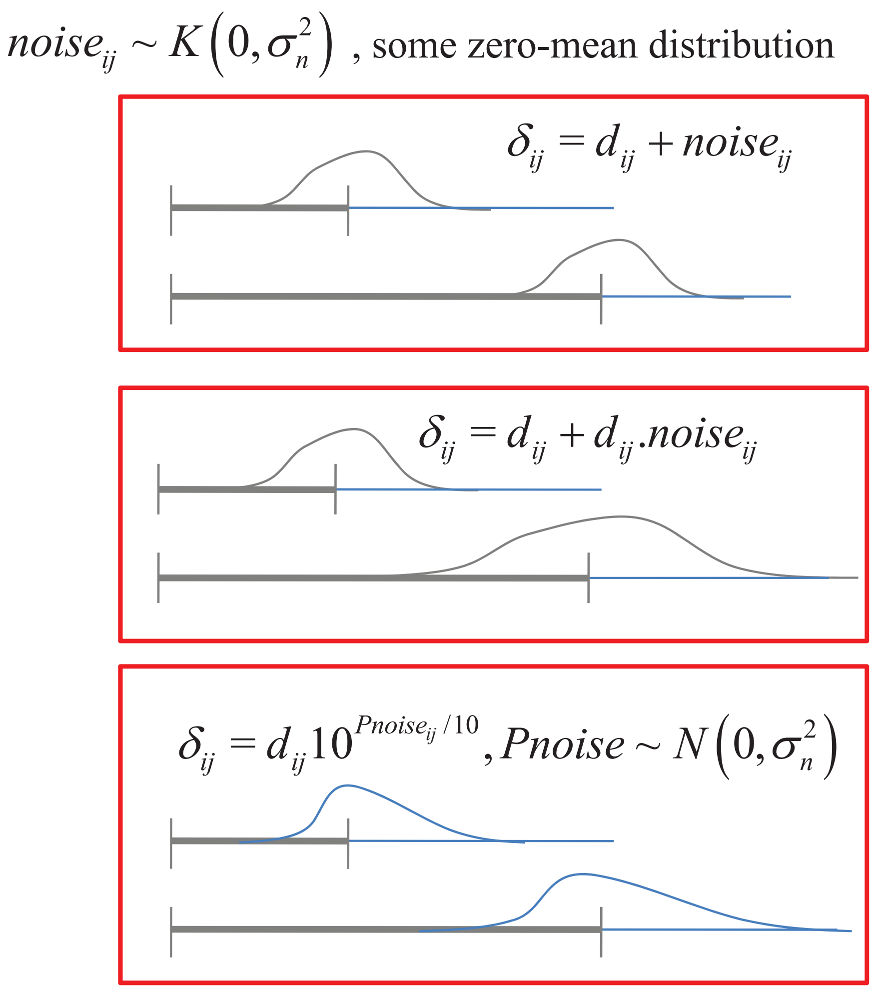


Figure 2.8: PPE's measurement noise model. (a) simple unbiased noise model [7][99], (b) proportional to distance unbiased model [43][24], (c) log-normal model [15][50]

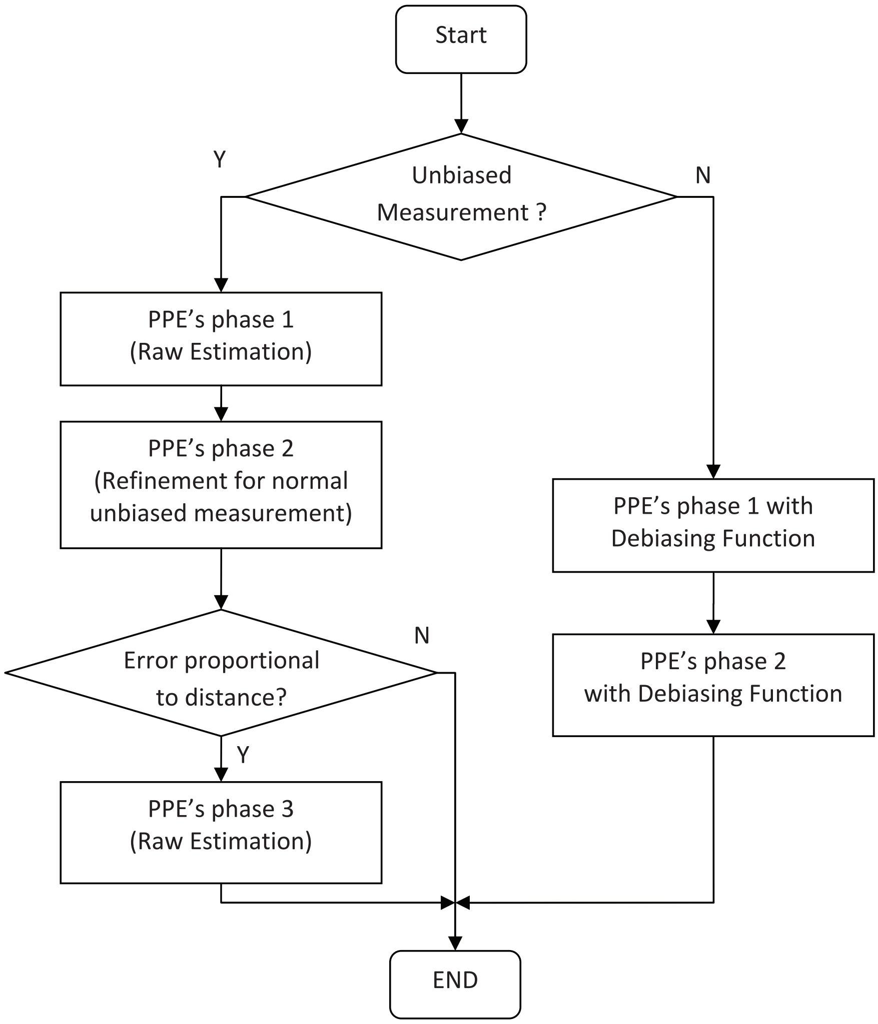


Figure 2.9: The summary flow chart of all PPE's phases and the condition for using them.

+ Send out requests for nearest anchors' coordinates x_j (at least three) and the measurements to each.

+ Set the initial position, $\tilde{x}_i = \frac{1}{M_i^{(1)}} \sum_j x_j$.

+ Set the initial moving step $\alpha^{(1)}$.

+ $k^{(1)} = 0$, $k^{(1)}$ is the iteration index.

repeat

$k^{(1)} = k^{(1)} + 1$;

$\overrightarrow{F_i}^{(1)} = \frac{1}{M_i^{(1)}} \sum_j \overrightarrow{f_{ij}}^{(1)}$;

(j is the index of the nearest **beacons**).

if $F_i^{(1)}$ *still decreases over the iterations* **then**

Double $\alpha^{(1)}$;

end

else

Halve $\alpha^{(1)}$;

Stop updating $\alpha^{(1)}$;

end

$\tilde{x}_i \leftarrow \tilde{x}_i + \alpha^{(1)} \overrightarrow{F_i}^{(1)}$;

until *(*$k^{(1)} > k_{max}^{(1)}$*) OR (*$\left\| \overrightarrow{F_i}^{(1)} \right\| < \epsilon$*)* ;

Algorithm 1: Phase 1: Raw estimation

+ $\alpha^{(2)} = \alpha^{(1)}$;

+ $k^{(2)} = 0$, $k^{(2)}$ is the iteration index.

repeat

if $mod(k^{(2)}, R) == 0$ **then**

Request the current updated coordinates of the neighbors.

Send out the current self updated position $\tilde{x}_i$ to the neighbors.

end

$k^{(2)} = k^{(2)} + 1$;

$\overrightarrow{F_i}^{(2)} = \frac{1}{M_i^{(2)}} \sum\limits_j \overrightarrow{f_{ij}}^{(2)}$;

(j is the index of all **reference nodes**).

$\tilde{x}_i \leftarrow \tilde{x}_i + \alpha^{(2)} \overrightarrow{F_i}^{(2)}$;

until *($k^{(2)} > k_{max}^{(2)}$) OR ($\left\| \overrightarrow{F_i}^{(2)} \right\| < \epsilon$)* ;

Algorithm 2: Phase 2: Pre-refinement

+ $\alpha^{(3)} = \alpha^{(2)}$;

+ $k^{(3)} = 0$, $k^{(3)}$ is the iteration index.

repeat

    if $mod(k^{(3)}, R) == 0$ **then**

        Request the current updated coordinates of the neighbors.

        Send out the current self updated position $\tilde{x}_i$ to the neighbors.

    end

    $k^{(3)} = k^{(3)} + 1$;

    $\overrightarrow{F_i}^{(3)} = \frac{1}{M_i^{(3)}} \sum\limits_j \overrightarrow{f_{ij}}^{(3)}$;

    (j is the index of all **reference nodes**).

    if $F_i^{(3)}$ *still decreases over the iterations* **then**

        Double $\alpha^{(3)}$;

    end

    else

        Halve $\alpha^{(3)}$;

        Stop updating $\alpha^{(3)}$;

    end

    $\tilde{x}_i \leftarrow \tilde{x}_i + \alpha^{(3)} \overrightarrow{F_i}^{(3)}$;

until *($k^{(3)} > k^{(3)}_{max}$) OR ($\left\| \overrightarrow{F_i}^{(3)} \right\| < \epsilon$)* ;

Algorithm 3: Phase 3: Refinement

Chapter 3

Acoustic based multi-object positioning

We have the discussions in previous Chapters about the sensor localization with the proposed method named Push-pull Estimation (PPE). The sensor localization is actually a must that provides prior information about sensor locations without which the object positioning could not be done. When the first problem is solved, we continue to present a method for the next problem to locate the moving sources, which is the ultimate goal of this thesis. This chapter introduces the idea for solving the problem of localizing multiple moving sources which is already described in the beginning of the chapter, then a full description is introduced next. In the first parts of the second Section of Chapter 4, the simulation results are for the idea illustration and the idea evaluation while in the later part, an demonstration with real data is presented.

3.1 Annotation and proposed method

Assume that there are M objects emitting continuous zero-mean acoustic signals and Q location-known sensors; these signals can be considered as $s_j(t), j = 1, .., M$. At each sensor i, the received data is denoted as $x_i(t)$. The data received at each sensor is the actual signal with continuous values of delay, similar to the model in [80], and is calculated according to

$$x_i(t) = \sum_{j=1}^{M} a_{ij} s_j(t - \tau_{ij}(t)),\ i = 1, .., Q \tag{3.1}$$

where a_{ij} is a real positive number representing the amplitude gain of the signal from source j measured at sensor i and $\tau_{ij}(t)$ is the propagation time of this signal. When the sources are fixed, the delays $\tau_{ij}(t)$ are constants. However, if the sources move, these parameters change at different time points,

$$\tau_{ij}(t) = \frac{d_{ij}(t)}{v_c}, \tag{3.2}$$

where $d_{ij}(t)$ is the immediate distance from sensor i to source j and v_c is the velocity of acoustic propagation. Source j's movement with velocity v_j causes $\tau_{ij}(t)$ to increase or decrease over time, resulting in a stretched or compressed image of the source signal on the time domain at receiver i. As a result, different shifts are caused to different f-components at the receivers. This phenomenon is known as Doppler effect [72] (chapter 17) and is expressed as

$$f_{ij} = \left(\frac{v_c}{v_c + v_j \cos(\theta_{ij}(t))} \right) f_j, \tag{3.3}$$

where f_j is some f-component of source j, f_{ij} is the shifted version of f_j received at sensor i, and $\theta_{ij}(t)$ is the immediate angle between $\overrightarrow{v_j}$ and $\overrightarrow{ij}$. Doppler effect demonstration is in Fig.3.1

The problem can be stated as: Without prior knowledge about the sources except for the received data at the sensors and the information that delayed versions of the sources are statistically independent of one another, the locations of sources must be determined.

Specifically, the problem can be seen in more details in Fig.3.2 with many challenges. The signals are compressed or stretched depending on the speeds of the sources due to Doppler effect. The collected data also includes Rayleigh multi-path fading noise. Rayleigh fading effect is caused when a signal propagates in every direction out of the source and is reflected when hitting obstacles in the setting before reaching to the recorders. As a consequence, not only the line-of sight signal to the sensor is recorded but also the reflections of the signal are recorded. The coming reflections

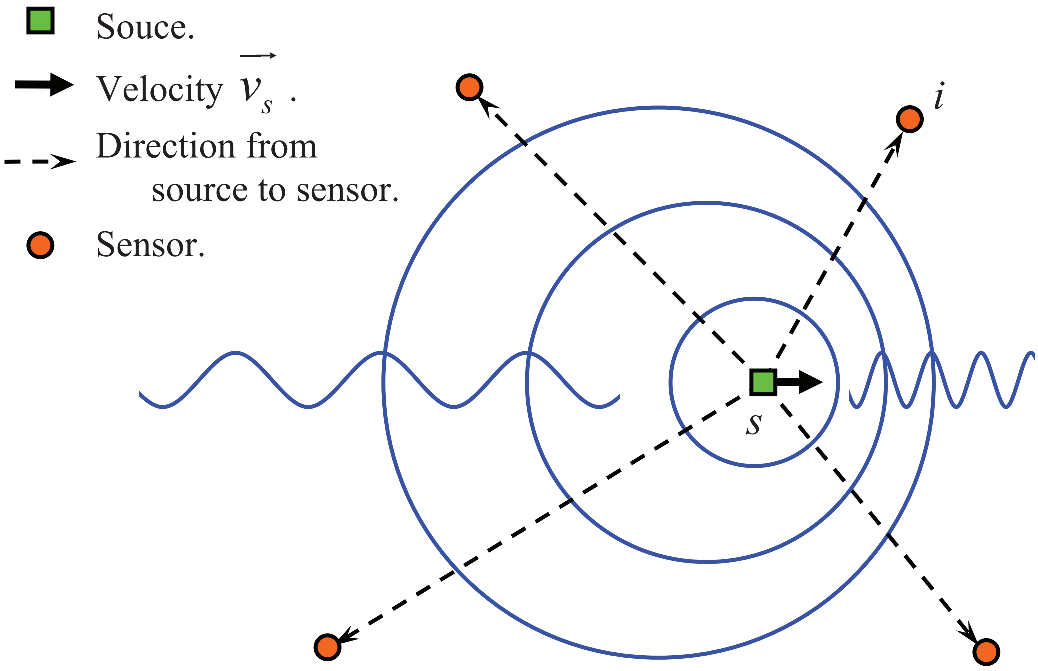


Figure 3.1: Doppler effect illustration.

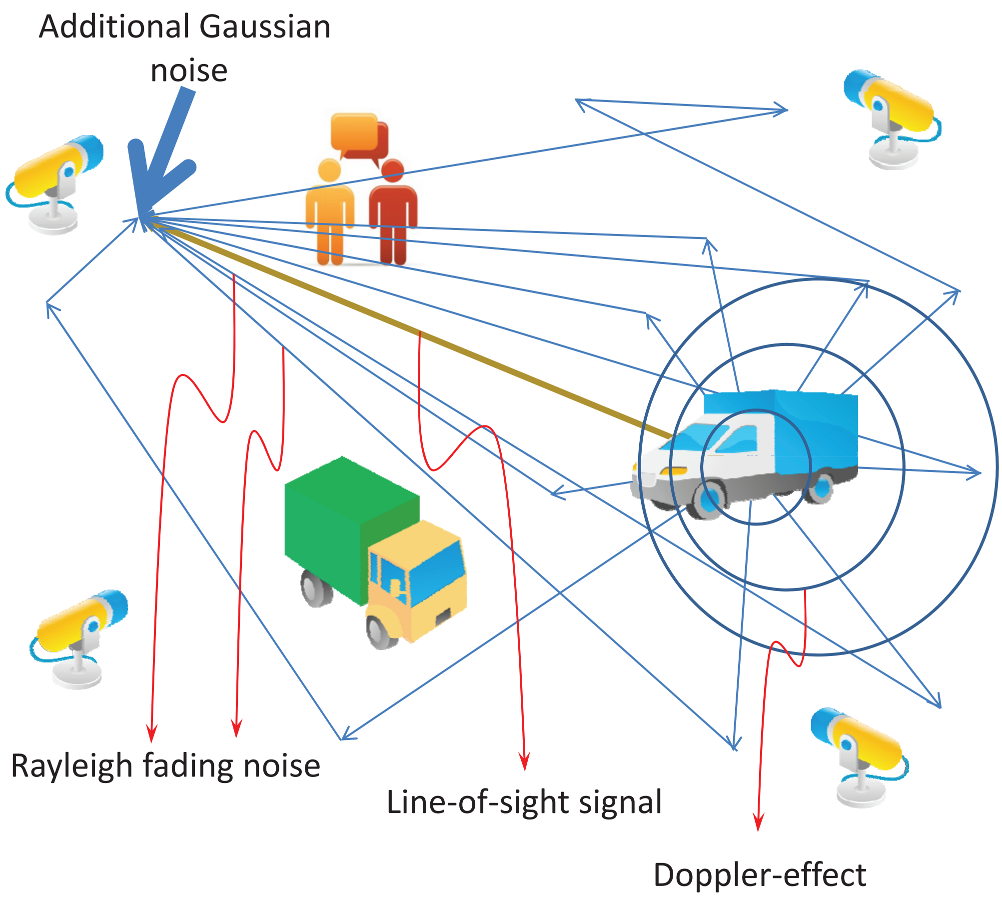


Figure 3.2: Acoustic multiple moving object problem and its challenges in terms of acoustic delays, Doppler effect, Rayleigh fading noise and environment noise.

is unpredictable in terms of amplitude and phase, making the recorded signal exceeds at some frequencies and degrades at some others. The problem is even more difficult when the sources move, that means the signal comes out of the source at different directions will be different on the time domain due to Doppler effect. Now each sensor receives a combination of signals from the sources or the data of convolved mixtures with the Rayleigh fading noise and the setting noise. In this part of source positioning, the Gaussian noise is assumed to be the noise of the environment. Fortunately, under the assumption that the source signals are independent, we can solve the problem of source positioning mainly with two clustering stages and an optimization stage.

As mentioned before, the feature $\tau_{ij}(t)$ is actually the best measurement, giving good results of object location estimation. However, it is difficult to extract this feature and a big load of communication is required to transmit all of the data to the base in order to perform the algorithm. In this thesis, we mainly focus on extracting the relative information among the magnitudes of the f-components at different sensors.

3.2 Distance information extraction

Note that if there is only one active fixed source and the others are inactive, or emitting no sound, applying Short Time Fourier Transformation (STFT) [66] to the sampled data $s_i(t_k)$ at this source and sampled data at each sensor, we obtain the same magnitude spectral images. The different parts of the STFT results are the scalar coefficients and the phase spectral images. Obviously, the time-delay τ_{ij} only affects the phase spectral image. Therefore, for multiple fixed and independent sources, if STFT is applied at each sensor for each mixture, the result is:

$$X_i(\omega) = \sum_{j=1}^{M} a_{ij} S_j(\omega) e^{-2\pi\tau_{ij}},\ i = 1, .., Q, \tag{3.4}$$

$$|X_i(\omega)| = \sum_{j=1}^{M} |a_{ij}| \, |S_j(\omega)|, \;\; i = 1, .., N. \tag{3.5}$$

As usual, the continuous form of STFT is difficult to compute and store, so the Discrete Fourier Transformation (DFT) form is the best choice of replacement. Equation (3.5) is rewritten in the discrete form as

$$|X_i(\omega_k)| = \sum_{j=1}^{M} |a_{ij}| \, |S_j(\omega_k)|, \;\; i = 1, .., Q. \tag{3.6}$$

Evidently, the magnitude spectrum data takes the form of instantaneous mixtures. The sound signals are zero-mean, mainly composed of sinusoid waves. Also, the delayed versions of source signals are statistically independent of one another. That leads to the fact that if an f-component is included in one source, it would not be present in the others. Therefore, the magnitude spectra of different $|S_j(\omega_k)|$ are orthogonal to each other or

$$|S_u\,(\omega_k)|^T \, |S_v\,(\omega_k)| = 0, \;\; u \neq v. \tag{3.7}$$

As a result, when the number of sources is less than or equal to the number of sensors, $|S_j(\omega_k)|$ in (3.6), or the the magnitude spectral image of $s_i(t_n)$ can be restored using a standard ICA [11][18][34][77]. The idea of applying ICA on Fourier domain is presented on [6][35]. However, note that ICA cannot restore the magnitudes of the original Independent Components (ICs). Instead of providing the exact $|a_{iz}| \, |S_z(\omega_k)|$, ICA results in $b_z \, |S_z(\omega_k)|$, $b_z \in \mathbf{R}$. That means ICA can restore the signal with some different scale coefficient b_z. $|X_i(\omega_k)|$ is the linear combination of orthogonal vectors $|S_j(\omega_k)|$, thus the inner product of each IC vector $|b_z| \, |S_z(\omega_k)|$ and each magnitude spectral image $|X_i(\omega_k)|$ contains the information of energy of this IC observed by sensor i.

$$(|b_z| \, |S_z(\omega_k)|)^T \, |X_i(\omega_k)| = |b_z| \, |S_z(\omega_k)|^T \sum_{j=1}^{M} |a_{ij}| \, |S_j(\omega_k)| \tag{3.8}$$

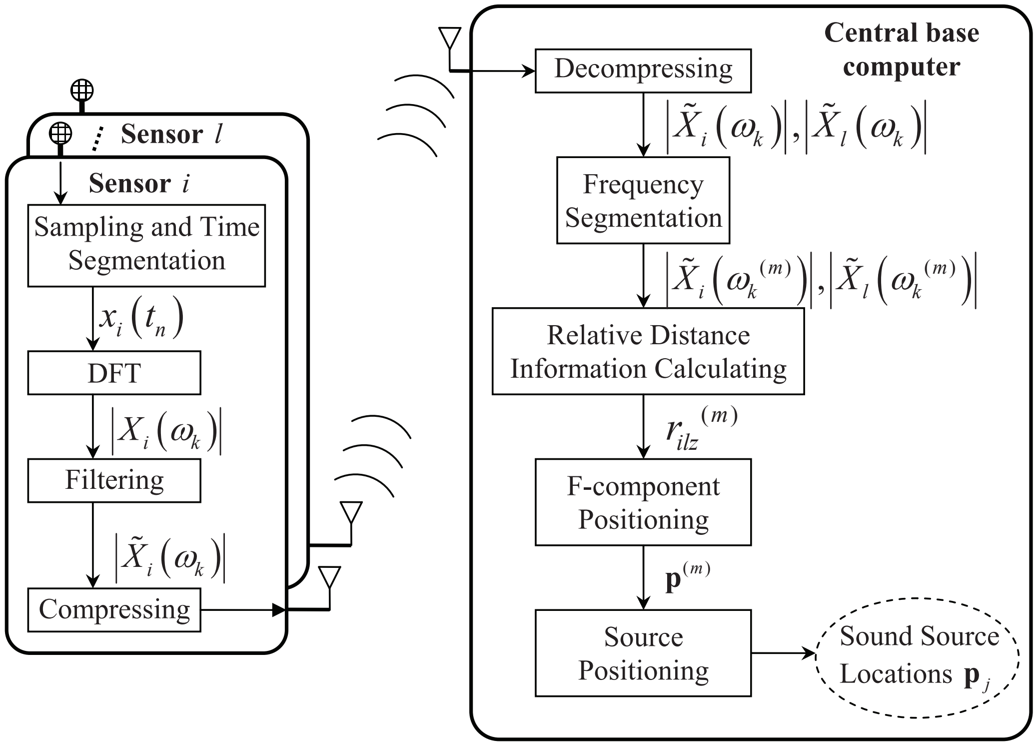


Figure 3.3: Sensor architecture and Base architecture of the proposed system.

or

$$(|b_z| \, |S_z(\omega_k)|)^T \, |X_i(\omega_k)| = |b_z a_{iz}| \, |S_z(\omega_k)|^T \, |S_z(\omega_k)| \, . \tag{3.9}$$

Therefore, for each IC z, for each pair of magnitude spectral images of observed data at sensors i and l, the ratio $\left|\frac{a_{iz}}{a_{lz}}\right|$ can be achieved as

$$\left|\frac{a_{iz}}{a_{lz}}\right| = \frac{(|b_z| \, |S_z(\omega_k)|)^T \, |X_i(\omega_k)|}{(|b_z| \, |S_z(\omega_k)|)^T \, |X_l(\omega_k)|}. \tag{3.10}$$

Meanwhile,

$$\left|\frac{a_{iz}}{a_{lz}}\right|^2 = \frac{a_{iz}^2 \, |s_j(t_n)|^T \, |s_j(t_n)|}{a_{lz}^2 \, |s_j(t_n)|^T \, |s_j(t_n)|} = \frac{E_{iz}}{E_{lz}}, \tag{3.11}$$

where E_{iz} is the energy sent by source z and received by sensor i in an interval of time. Since the absorption of gas molecules is insignificant, due to the inverse square law, the energy of sound decreases proportionally to the inverse square of the distance [72] (chapter 17). In other words,

$$\frac{E_{iz}}{E_{lz}} = \frac{\left(1/d_{iz}\right)^2}{\left(1/d_{lz}\right)^2}. \tag{3.12}$$

From (3.10), (3.11) and (3.12), we have the relationships of all pairs of distances from any tracked object j to all sensors,

$$r_{ilz} = \left|\frac{a_{iz}}{a_{lz}}\right| = \frac{d_{lz}}{d_{iz}} = \frac{\left(|b_z|\,|S_z(\omega_k)|\right)^T |X_i(\omega_k)|}{\left(|b_z|\,|S_z(\omega_k)|\right)^T |X_l(\omega_k)|},\ i \neq l. \tag{3.13}$$

Based on these relationships, the locations of all of the sources will be inferred.

However, it should be remarked that ICA achieves poor separation in the presence of noise. Moreover, when the sources move, frequency shifts occur and ICA can no longer produce ICs $b_z\,|S_z(\omega)|$. Therefore, we extend the meaning of “independent” for sources to “being in the state in which a shifted major f-component of a source does not overlap the shifted major f-components of other sources”. The interferences of minor f-compo-nents among the sources are considered as noise. For this situation, instead of using ICA, we apply a more robust method, the main idea of this study, which involves less computation based on clustering techniques. Equation (3.6) then is rewritten as

$$|X_i(\omega_k)| = \sum_{j=1}^{M} |a_{ij}|\,|S_{ij}(\omega_k)|,\ i = 1, .., Q, \tag{3.14}$$

where $|S_{ij}(\omega_k)|$ is the discrete frequency image of the signal emitted by source j under the “view” of sensor i. Remark that for different sensors, S_j is not the same as in (3.6) due to different frequency shifts caused by Doppler effect.

Note that the method only needs the magnitude information of the source’s DFT results, not on the phase as in [59]. Now considering a specific segment on the frequency domain (ω_a, ω_b)

containing all shifted versions of some f-component of source z without any interference from other sources' shifted f-components, we have

$$\begin{aligned}\left|X_i(\omega_k^{(m)})\right| &= \sum_{j=1}^{M} |a_{ij}| \left|S_{ij}(\omega_k^{(m)})\right| \\ &= |a_{iz}| \left|S_{iz}(\omega_k^{(m)})\right|, \quad i = 1, .., Q,\end{aligned} \tag{3.15}$$

where $\omega_k^{(m)} \in (\omega_a, \omega_b)$ and m is the index of the f-component. Although this f-component has different shifted versions, its energy is unchanged. The reason is while a signal is stretched and compressed by Doppler effect on the time domain, its amplitude at the source keeps unchanged, or

$$\begin{gathered}\left|S_{iz}(\omega_k^{(m)})\right|^T \left|S_{iz}(\omega_k^{(m)})\right| = \left|S_{lz}(\omega_k^{(m)})\right|^T \left|S_{lz}(\omega_k^{(m)})\right|, \\ i \neq l.\end{gathered} \tag{3.16}$$

Based on this fact, if an f-component belongs to source z, then all relative distance relationships in (3.13) are computed according to

$$r_{ilz}^{(m)} = \left|\frac{a_{iz}}{a_{lz}}\right| = \frac{d_{lz}}{d_{iz}} = \sqrt{\frac{\left|\tilde{X}_i(\omega_k^{(m)})\right|^T \left|\tilde{X}_i(\omega_k^{(m)})\right|}{\left|\tilde{X}_l(\omega_k^{(m)})\right|^T \left|\tilde{X}_l(\omega_k^{(m)})\right|}}, \; i \neq l \tag{3.17}$$

where $\tilde{X}_i(\omega_k)$ is the result of noise filtering $X_i(\omega_k)$ and $\tilde{X}_i(\omega_k^{(m)})$ is the frequency image of $\tilde{X}_i(\omega_k)$ on the segment (ω_a, ω_b) (see Fig.3.3). This implies that, for each f-component m of a source within the frequency segment, a set of relative distance relations can be computed and the position of the source having these components can be estimated. Therefore, there are necessarily two clustering stages, one for grouping the shifted frequency components to determine the segment (ω_a, ω_b), and the other for grouping f-component positions to calculate source locations after f-component positions are computed. This is the main idea of the design which is described in more details in the next section. The advantages of this system are: (a) it is more robust than our previous system even when the sources are fixed, (b) it works well with moving sources and tolerates the co-existence of Doppler effect and Rayleigh multi-path fading, (c) it is considered to be a distributed method since the computation load is shared among the sensors and the communication cost is low, and (d) it is not constrained by the condition that the sensor number is larger than the source number.

3.3 Multi-object tracking system architecture

The design of the system for acoustic tracking [19] is depicted in Fig.3.3 based on the key idea mentioned in Subsection 3.2 about distance information. The figure describes both acoustic sensor work flow and central base computer work flow. Because of the nature of WSNs, the input data is already decomposed at the sensors and hence can be utilized to build up a *data decomposition* algorithm. It also can be seen in the figure that the source localization computing is decomposed into two different processes, one at the sensors and the other at the central base. A powerful enough computer is used at the base to solve its flow in a short time, not longer than the sum of the computing time at the sensors and data transmission time. Then although the process at the sensors must be executed before that at the base, with continuous input data, these two processes function in a pipeline manner, or the system works with *process decomposition* method. In other words, the design can be considered to be *data-and-process decomposition*. Since the data is decomposed into portions at sensors and the sensors have the same work flow, the process at the sensors obviously uses SIMD (Single Instruction on Multiple Data stream) approach [4]. Details of the work flows are following.

3.3.1 Acoustic sensor architecture

At the sensors, the acoustic signal is sampled and synchronically segmented into half-overlapped frames in the "Sampling and Time Segmentation" stage. A frame at sensor i is denoted by $x_i(t_n)$

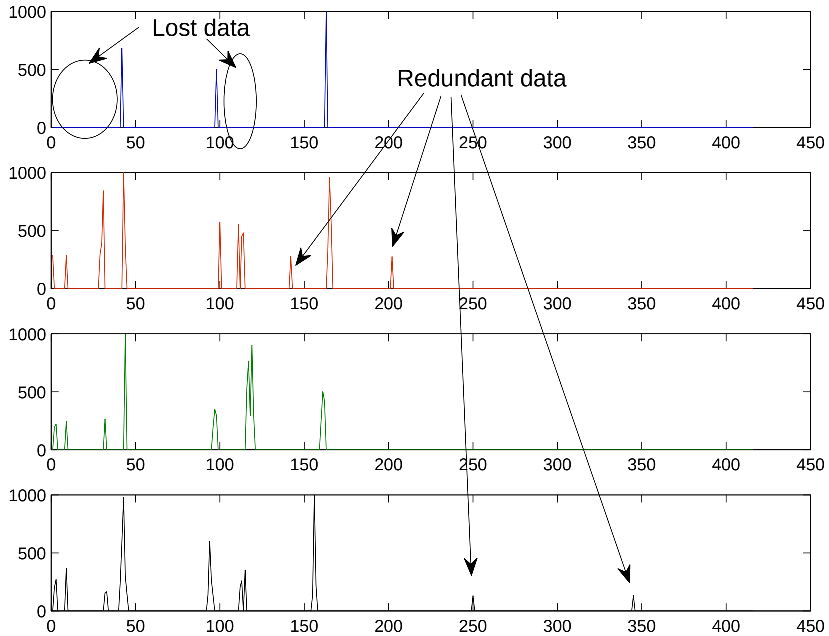


Figure 3.4: An example of data after filtering on Fourier domain

and the time length of the frame is also called time segment, denoted by T_f.

Since our method is based on the energies within frequency intervals, we need to minimize the spectral leakage effect [31] caused by limited length DFT transformation. Thus, each frame of sensed data with length L is multiplied by the Hamming window whose weights are defined as [22]

$$\begin{aligned} w_{Hamming}(k) =& 0.54 - 0.46 \cos\left(\frac{2\pi(k-1)}{L-1}\right); \\ & k = 1, .., L. \end{aligned} \tag{3.18}$$

The input data from the recorders always includes additional Gaussian noise whose spectrum is spread over the frequency domain. We can remove the noise by detecting its level and forcing all low f-components to zero. Fig.3.4 demonstrates the results after the filtering process. The filtered data may have some lost f-components and may include redundant f-components; this is tolerated by the system as we will see in later analysis. Filtering step allows for the retention of only several dominant f-components, and as a result, the amount of data to be transmitted from a sensor to the base computer is reduced considerably. Moreover, only half of the frequency image length is needed owing to the symmetric property of the image. For example, instead of sending 1638 values in every time segment of 0.2s with sampling frequency $F_s = 16.384KHz$, each sensor sends only the compressed data containing less than twenty f-components (pairs of values and indexes). This is one of the key ideas for reducing the communication cost so that the method can be implemented into WSNs.

As the trade-off to this low communication cost, the computation cost at the sensors is high with DFT transformations of lengthy frames. However, as can be seen in Fig.3.4, a sensor can skip calculating the frequency bins where the probability of major f-components' existence is low due to the feedback from the base. Only the frequency bins in the high energy ranges are computed. Therefore, the computation cost at sensors are also reduced considerably.

3.3.2 Central base computer architecture

At the central base, the flow is straightforward and consistent with what we analyzed in Section 3.3. Data received from the sensors needs to be decompressed and fed into the “Frequency-Segmentation” process. This stage marks dominated f-components as well as the segments that contain the components with the index m. Then the process “Relative Distance Information Calculate” computes a set of $r_{il}^{(m)}$ for each component. These sets are then input into the “F-component Positioning” process to estimate the output position of each dominant f-component $\vec{p}^{(m)}$. Ideally, the f-compo-nents that belong to the same source j should have the same position $\vec{p}_j$. However, factors that affect the frequency images will influence the detection result and make f-components belonging to the same source j not have the same position but have the positions that are close to the real position of source j. Some factors do not much influence the frequency image, like the measurement error due to sampling resolution, the heat noise of the electric system and the DFT frequency leak. DFT frequency leak, which depends on the length of frame (window length), obviously changes the frequency image but is reduced considerably by Hamming window multiplication. Others factors are more serious like setting noise, Doppler effect and Rayleigh multi-path fading effect. Setting noise, or background noise is usually very high and affect much the measurements. Meanwhile, Doppler effect causes the shifted f-component, which increases the probability of wrong grouping on frequency domain. Rayleigh multi-path fading is even more serious than background noise since it affects directly on the regions that cover dominant f-components. With the same level of noise, Rayleigh multi-path fading gives more changes on the frequency images than other kinds of noise. How background noise, Doppler effect and Rayleigh multi-path fading effect influence the localization can be seen in simulation results of Section 4.2. Due to those factors, f-components of a source have positions that are close to the real source position. Therefore, the final stage “Source Positioning” is necessary to cluster those $\vec{p}^{(m)}$ and estimate $\vec{p}_j$ using the averaging mechanism. Details of the three main processes are described in the following subsections.

Frequency Segmentation

"Frequency Segmentation" determines the frequency segments on which all shifted components of a dominant f-component are included. As can be seen in Fig.3.4, data from the "Decompressing" block may have some missing and redundant f-components. Frequency segmenting is actually a clustering task which groups shifted components of an dominant f-component and determines the frequency segment based on the cluster. Dopp-ler effect influences the f-components differently, the higher is the frequencies, the larger is the shift. From (3.3), an f-component of source j at f_0 has shifted versions within $(\frac{v_c}{v_c+v_j}f_0, \frac{v_c}{v_c-v_j}f_0)$. This frequency interval varies depending on f_0 on the frequency scale, however, it is fixed on the logarithmic scale. Indeed,

$$\begin{aligned}\Delta f(dB) &= \log_{10}(\frac{v_c}{v_c - v_j}f_0) - \log_{10}(\frac{v_c}{v_c + v_j}f_0) \\ &= \log_{10}(\frac{v_c + v_j}{v_c - v_j})\end{aligned} \tag{3.19}$$

Therefore, clustering is performed on the $\log_{10}(.)$ scale of the frequency image according to the criteria: (a) the width of each segment is not larger than $\Delta f(dB)$ as defined in (3.20); (b) the number of nonzero f-components within the clustered interval is greater than 2 so that the number of constraints is at least 3; and (c) the average energy of an f-component received at the sensors must be larger than the detected noise level. A sliding window with the width $\Delta f(dB)$ is used here to detect the frequency intervals that satisfy (b) and (c). With such a mechanism, the number of sources can be larger than that of sensors. The total loss of some f-components due to filtering is acceptable because a source can be positioned with only one of its f-components. In addition, the redundant f-component will hardly be taken into account, and the groups with missing data can still be considered for position estimating. The maximum number of sources now depends on the sampling frequency, the speeds of the sources and the number of dominant f-components in each source. If the minimum frequency of all f-components is f_{min}, the mean number of f-components in a source is $\overline{n_f}$. Then the possible number of sources can be up to

$$M_{max} = \frac{\left(\frac{\log_{10}\frac{F_s}{2} - \log_{10} f_{\min}}{\Delta f(dB)}\right)}{\overline{n_f}}, \tag{3.20}$$

where the numerator is the possible total number of f-components caused by all sources. If the maximum source velocity decreases, then the maximum number of sources increases. Obviously, the possible number of sources is not related to the number of sensors since the locations of sources are determined by their dominant f-components. The accuracy is improved with a large number of sensors but it is not if the number of sources is changed, providing that the shifted versions of dominant f-components of sources do not interfere with each other on the frequency domain.

F-component Positioning

All of the ratios r_{ilz} of different pairs of distances from a source to sensors are calculated (see (3.17)) in "Relative Distance Information Calculating" process before being fed into the "F-component Positioning" process. Each ratio defines a constraint, or a position curve to which the source belongs. As illustrated in Fig.3.5, if $r_{ilj}^{(m)} = 1$, the curve is the perpendicular bisector of the line segment connecting sensor i and sensor l; otherwise, the curve is a circle.

The additional noise and spectral leakage always exist in the recorded data and cannot be completely removed. Moreover, when a source is very close to a sensor, its signal may dominate at this sensor but be at a level lower than noise level at other sensors and produce large errors in the constraints. Therefore, the error in the constraints is unavoidable, and the solution for the position of f-component m should be a vector $\mathbf{p}^{(m)}$, $\mathbf{p}^{(m)} \in \mathbf{R}^2$ that compromises its constraints. We propose an objective function in a quadratic sum form for this compromise

$$\mathbf{F}_j = \sum_{i}^{Q} \sum_{l,l \neq i}^{Q-1} (d_{ij} - r_{ilj} d_{lj})^2, \ 0 < r_{ilj} < \infty \tag{3.21}$$

and the solution for source j will be

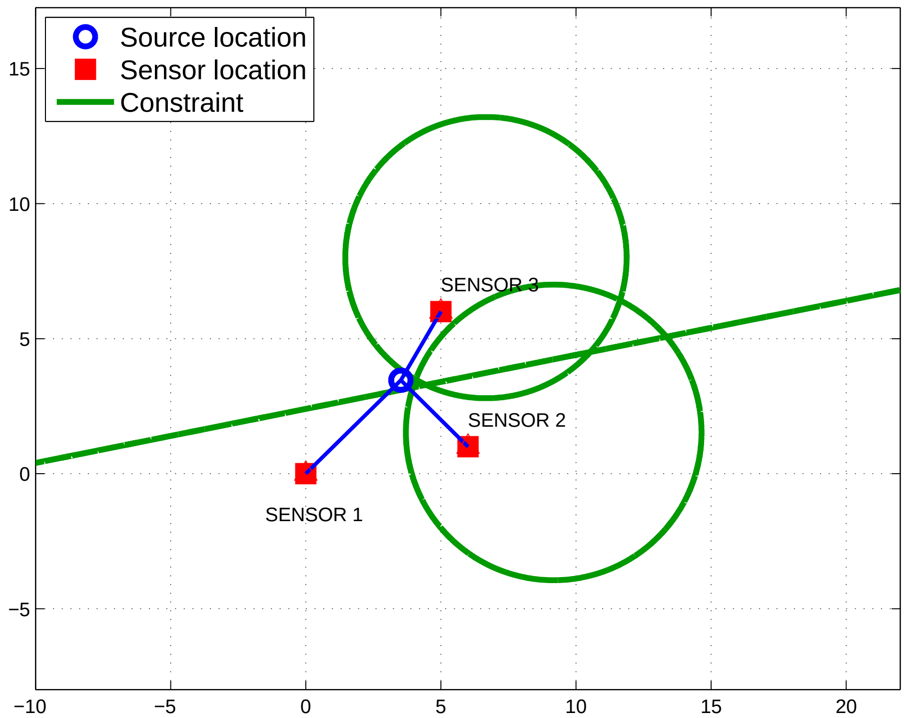


Figure 3.5: An example of constraints obtained through a set of relative distance relations in which the number of involved sensors is 3.

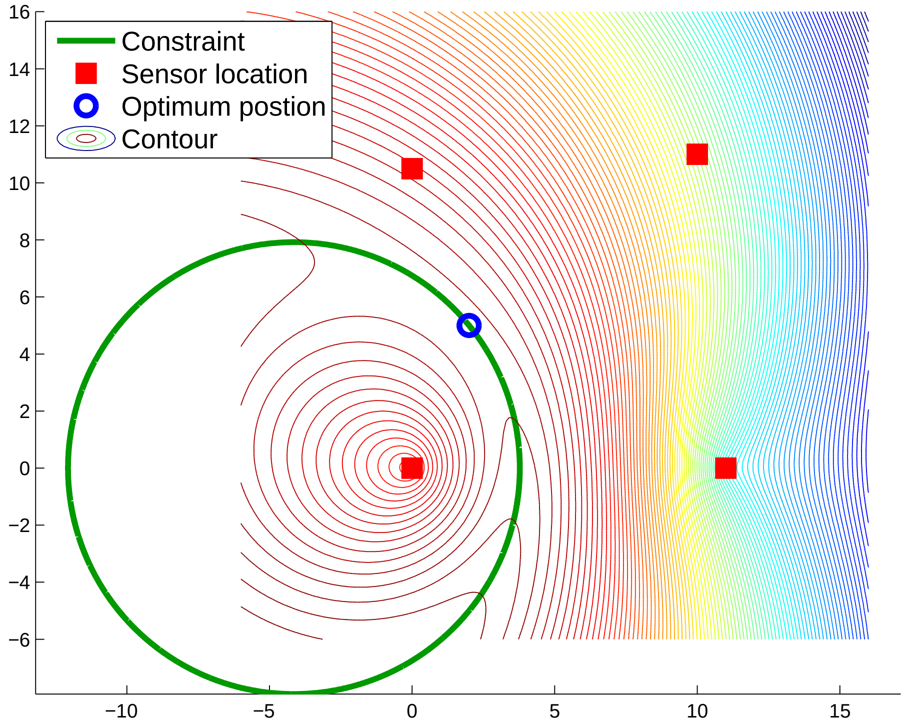


Figure 3.6: Illustration of the rigid problem, one of the reasons that make gradient-based convergence slow.

$$\mathbf{p}^{(m)} = \underset{\mathbf{p}^{(m)}}{\arg\min}\, \mathbf{F}_j. \tag{3.22}$$

In fact, the objective function in (3.21) is not the quadratic form with respect to $\mathbf{p}^{(m)}$, but each constraint gives a quadratic form with respect to the minimum distance from $\mathbf{p}^{(m)}$ to the curve that satisfies $d_{ij} = r_{ilj}d_{lj}$. As depicted in Fig.3.6, a combination of constraints may cause a valley whose bottom is slightly sloped. This situation takes the simple negative gradient method much time to converge because upon reaching the bottom of the valley, the search process oscillates back and forth on the sides with little progress to the optimum position. Based on the analysis in [100] (page 149) and the results in [91], we choose the Fletcher-Reeves conjugate-gradient method for this optimization problem. The line search is performed using the Fibonacci segment search [100] and the first order derivative used in the optimization is:

$$\frac{d\mathbf{F}_j}{d\mathbf{p}^{(m)}} = \begin{bmatrix} \sum\limits_{i}^{Q}\sum\limits_{l,l\neq i}^{Q-1} \begin{pmatrix} 2\left(\frac{(p_{jx}-m_{ix})}{d_{ij}} - \frac{r_{ilj}(p_{jx}-m_{lx})}{d_{lj}}\right)\times \\ \times\left(d_{ij} - r_{ilj}d_{lj}\right) \end{pmatrix} \\ \sum\limits_{i}^{Q}\sum\limits_{l,l\neq i}^{Q-1} \begin{pmatrix} 2\left(\frac{(p_{jy}-m_{iy})}{d_{ij}} - \frac{r_{ilj}(p_{jy}-m_{ly})}{d_{lj}}\right)\times \\ \times\left(d_{ij} - r_{ilj}d_{lj}\right) \end{pmatrix} \end{bmatrix} \tag{3.23}$$

One remarkable note is that since our method is derived from the mutual information from every pair of active sensors, when the sensor number is greater than 3, the constraint number becomes much higher than the sensor number. For instance, if the sensor number is 5, then the constraint number is 10. The more constraints are there, the more accurate will be the results due to the averaging mechanism.

Source Positioning

Data from the "F-component Positioning" stage includes the positions of the f-components. The source to which a specific f-component belongs must next determined, and the answer can be found based on the estimated positions of these f-components. This stage clusters f-component positions and computes the source coordinates as the mean values of f-component groups. The nearest-neighbor clustering, or d-min clustering technique [21] is used in this "Source Positioning" process with some modification compared to the typical version. The reason for this modification is that the resulted positions of different f-components have different levels of error depending on their energy to noise ratios. The higher is the energy, the more reliable is the resulted position. Therefore, each f-component position should be assigned with a weight. Then, instead of being randomly chosen as an input of d-min clustering, the positions are considered in the order of their weights. Moreover, the weights are also used to calculate the final source locations so that the computed f-component positions with higher energy play more important roles (see Fig.3.7). The

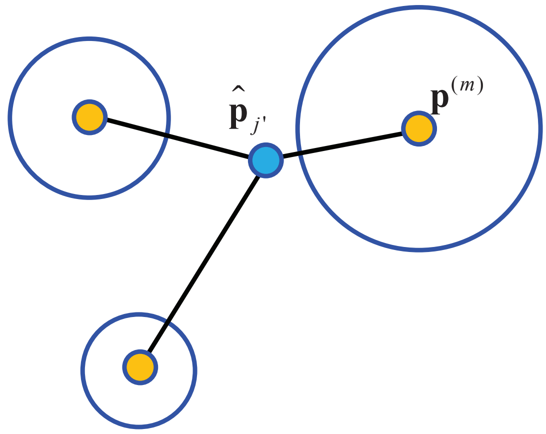


Figure 3.7: Illustration of weighted average mechanism, the final position is closer to the higher weight f-component position.

weight, $w^{(m)}$, is the mean values of the f-component's energies at the sensors. Equations (3.24) and (3.25) estimate the position of a source that includes the f-components whose positions are in the group j', denoted by $D_{j'}$:

$$\widehat{\mathbf{p}}_{j'} = \frac{\sum\limits_{\mathbf{p}^{(m)} \in D_{j'}} w^{(m)} \mathbf{p}^{(\mathbf{m})}}{\sum\limits_{\mathbf{p}^{(\mathbf{m})} \in \mathbf{D}_{\mathbf{j}'}} w^{(m)}}. \tag{3.24}$$

where

$$w^{(m)} = \frac{1}{Q} \sum_{i}^{Q} E_i^{(m)}. \tag{3.25}$$

$E_i^{(m)}$ is the energy of the f-component m received at sensor i. The distance from an f-component's position to a current group j' is the distance from it to the current estimated $\widehat{\mathbf{p}}_{j'}$.

3.3.3 Algorithm summary of the proposed method

**** SELF LOCALIZATION:***

+ De-biasing function $g(d) = d$;

if *measurement error model is log-normal* **then**
De-biasing function $g(d) = log_{10}(d)$;

end

+ $Phase3\_Flag$ =0;

if *measurement error model is proportional to distance* **then**
$Phase3\_Flag$ = 1;

end

+ Perform PPE's phase 1 with $g(d)$;

+ Perform PPE's phase 2 with $g(d)$;

if $Phase3\_Flag$ == *0* **then**
+ Perform PPE's phase 3 with $g(d)$;

end

+ Send location to base station.

**** DATA COLLECTION FOR OBJECT POSITIONING:***

while $AcousticDetectFlag == 1$ **do**
+ Sample and do time segmentation

+ Get $x_i(t_n)$, $n = 1..L$

+ Multiply Hamming window

$$w_{Hamming}(k) = 0.54 - 0.64cos\left(\frac{2\pi(k-1)}{L-1}\right)$$

$$\mathbf{x} = \mathbf{x}^T\mathbf{w}_{Hamming}$$

+ DFT amplitude calculation

$$X_i(\omega_k) = \sum_{n=1}^{L} x_i\left(t_n\right)e^{-2\pi j t_n}$$

+ Filtering

$$\widetilde{X}_i\left(\omega_k\right) = \begin{cases} X_i\left(\omega_k\right) & if\ \left|X_i\left(\omega_k\right)\right| \geq \max\left(\left|HighFrequencyNoise\right|\right) \\ 0 & otherwise. \end{cases}$$

+ Compress $\left|\widetilde{X}_i\left(\omega_k\right)\right|$ and send it to base station.

end

Algorithm 4: Sensor i work flow

+ Decompress data

+ Frequency Segmentation base on Sliding window in 3.3.2

+ Calculate relative distance information

$$r_{ilz}^{(m)} = \left|\frac{a_{iz}}{a_{lz}}\right| = \frac{d_{lz}}{d_{iz}} = \sqrt{\frac{\left|\tilde{X}_i(\omega_k^{(m)})\right|^T \left|\tilde{X}_i(\omega_k^{(m)})\right|}{\left|\tilde{X}_l(\omega_k^{(m)})\right|^T \left|\tilde{X}_l(\omega_k^{(m)})\right|}},\ i \neq l \text{ (see (3.16))}$$

+ Position frequency component

$$\mathbf{F}_j = \sum_{i}^{Q} \sum_{l,l \neq i}^{Q-1} (d_{ij} - r_{ilj} d_{lj})^2,\ 0 < r_{ilj} < \infty$$

$$\mathbf{p}^{(m)} = \underset{\mathbf{p}^{(m)}}{\arg\min}\, \mathbf{F}_j. \text{ (see Appendix B and first derivative of } \mathbf{F}_j \text{ in (3.22)).}$$

+ Cluster for final source position

$$w^{(m)} = \frac{1}{Q} \sum_{i}^{Q} E_i^{(m)}.$$

$$\widehat{\mathbf{p}}_{j'} = \frac{\sum\limits_{\mathbf{p}^{(m)} \in D_{j'}} w^{(m)} \mathbf{p}^{(\mathbf{m})}}{\sum\limits_{\mathbf{p}^{(\mathbf{m})} \in \mathbf{D}_{\mathbf{j}'}} w^{(m)}}.$$

Algorithm 5: Base station i work flow

Chapter 4

Experiment results and discussions

4.1 Experiment results and Discussions on PPE

We conduct four main experiments. The first experiment focuses on PPE's convergence confirmation; the second tests the accuracy enhancement of the refinement phase compared with that of the pre-refinement phase. The third experiment is about PPE's ability to reduce communication cost, while the last one analyzes the performance comparison of PPE with dwMDS and MLE, both with simulated data and real data. We mainly examine a group of nodes to highlight PPE's characteristic instead of simulating the whole network with a very large number of nodes. Each normal node has four reference anchors and has other normal nodes as reference neighbors. Table.4.4 defines the common parameters that we use in all four experiments. For convenience of demonstration, the values $\alpha^{(2)}$ and $\alpha^{(3)}$ are equal to $\alpha^{(1)}$. In this section, the normal nodes are updated in a random order to show the distributed ability of PPE.

4.1.1 Verifying the convergence of PPE

This simulation set is aimed at verifying the algorithm's robustness in both pre-refinement and refinement phases. We choose a normal node as the tested node and examine its movement trend

Table 4.1: Common parameters used in simulations.

Deployed area	[0m, 28m] x [0m, 28m]
Total number of nodes N	196
Number of anchors	4
Distribution over area	uniform

under the force effect of the other normal nodes in the group after they undergo their raw estimations. The chosen nodes are tested in two situations: at the corner and in the middle of the deployed area of the group. The $noise_{ij}$ in this subsection is Gaussian with $\sigma_n = 0.4$, the largest value in the chosen range.

Fig.4.1a is the result of the first phase where each normal node uses anchors to produce its raw-estimation ($k_{max}^{(1)} = 10$). The actual locations of all nodes are denoted as small dots while the errors are presented as solid lines. Fig.4.1b illustrates the convergence image (or the representation of the movement tendency) of the corner tested node. Every position on the [-7m,35m]x[-7m,35m] grid with a grid step of 2m (the deployed field is [0m,28m]x[0m,28m]) is examined as if it is the current estimated position of the tested node. From each position, the tested node performs two successive position updates which are presented by two arrows to show its movement tendency. The figure shows that the tested node, at any current estimation and with its reference nodes suffering from a high degree of error, always moves to the unique balanced point which is close to its actual position.

Likewise, Fig.4.1c is the estimation result of all normal nodes when phase 2 is finished with $k_{max}^{(2)} = 10$ and Fig.4.1d is the convergence image of the corner tested node with respect to the estimation result in Fig.4.1c. The errors of all normal nodes are reduced compared to the result of the raw estimation in Fig.4.1a. Comparing the two convergence images (Fig.4.1b and Fig.4.1d), the balanced positions of the corner tested node are almost the same. This confirms the fact that a node's updating is quite independent of those of others and this is the basis for the reduction in

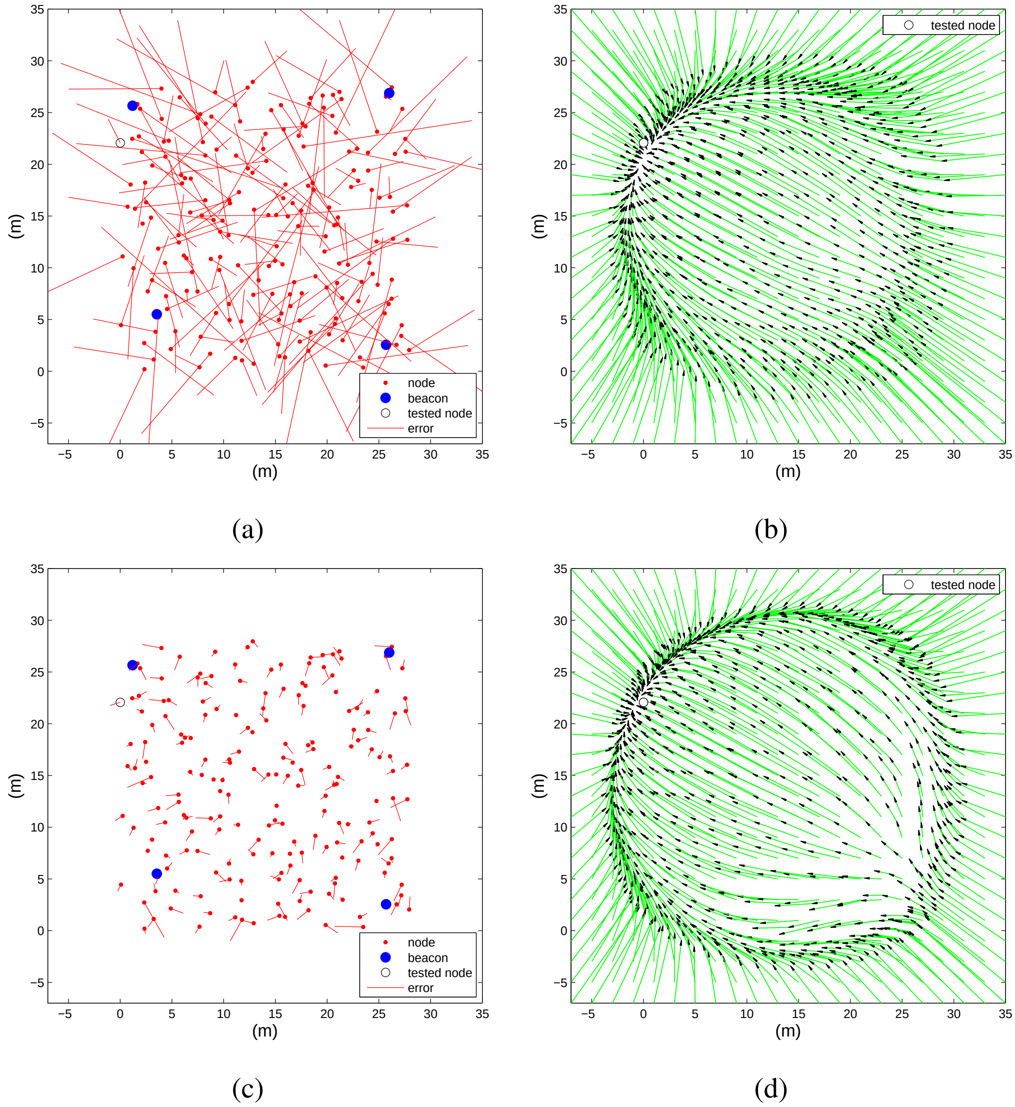
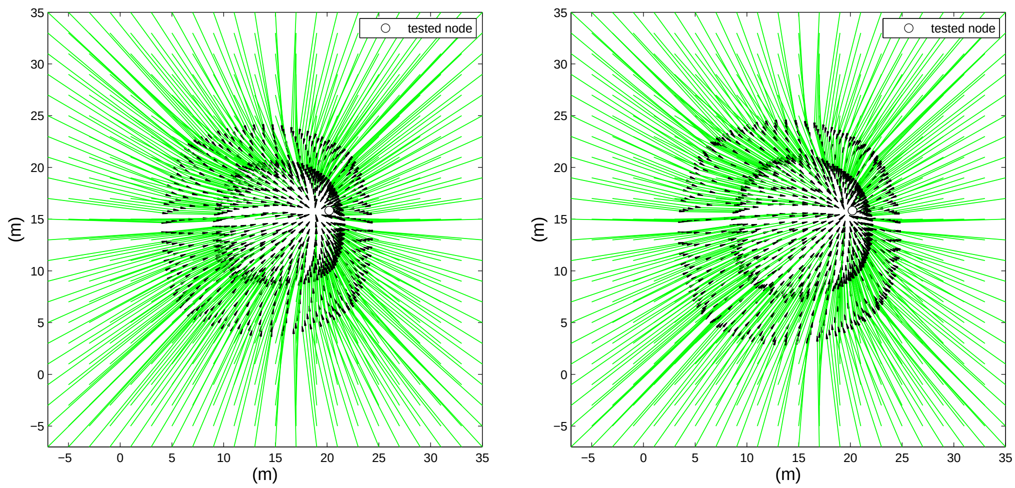
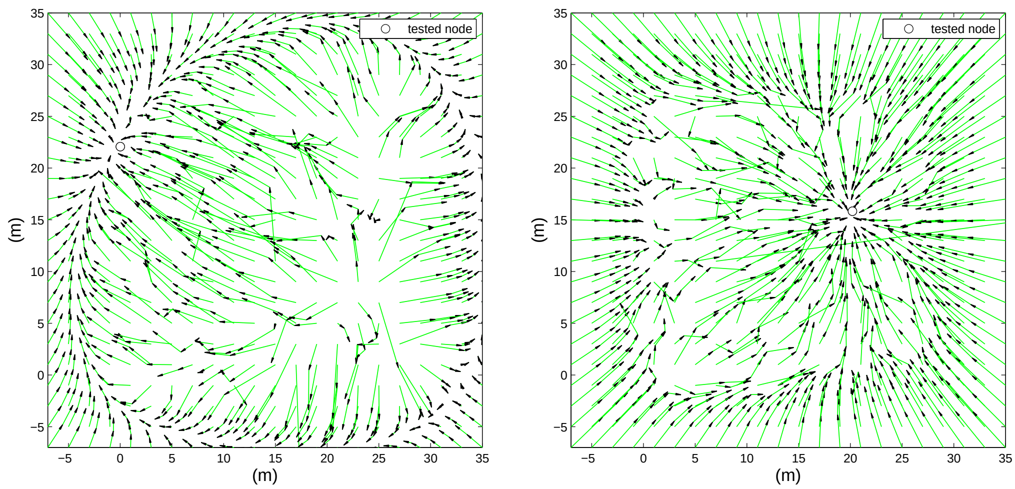


Figure 4.1: (a) and (c) are the estimation results at the beginning and ending of phase 2, alternately, the errors of the estimated positions are denoted with solid lines. (b) and (d) are the movement tendency images of the corner tested node with respect to the estimation results in (a) and (c), alternately when equation (2.7) is used.



(a) (b)

Figure 4.2: (a) and (b) are the movement tendency images of the middle tested node with respect to the estimation results in Fig.4.1a and Fig.4.1c, alternately when equation (2.7) is used.



(a) (b)

Figure 4.3: (a) and (b) are the movement tendency images of the corner tested node and the middle tested node, alternately at the start of phase 3 in which equation (2.8) is used.

communication cost of PPE, as we mentioned in Section 2.1.4. This is also true when the tested node is in the middle of the deployed area, as depicted in Fig.4.2. Fig.4.2a and Fig.4.2b are the convergence images of the middle tested node when phase 1 is complete and when phase 2 is complete, respectively. The movement tendency of a middle node is straighter and faster than that of a corner node. The balanced point is still unique and quite independent of the current estimation result of the reference neighbors.

The movement curve in the convergence images can be explained by the force effects of the small sub-areas of the reference nodes. Each of these areas attempts to keep away the tested node at a distance approximate to the distance from itself to the balanced point. Therefore, on the way to the balanced position, the tested node tends to move away from the side on which there are more reference nodes and creats the movement curve. Fig.4.1 and Fig.4.2 show that if the current estimated position is in the middle of the group, then the convergence is fast and the movement is straight. This is why we use the average beacon position as the initial input position in phase 1.

Phase 3's convergence verification is demonstrated in Fig.4.3, after phase 2 is complete. Fig.4.3a is for the corner tested node, and Fig.4.3b illustrates the middle tested node. It can be seen that if the current estimated position of the tested node has a large error, phase 3 still guarantees a correct convergence. The balanced position is unique and close to the actual position. Although the number of steps for the tested node to get to the balanced position is higher, the result of estimation is better. Obviously, all three phases of PPE guarantee network localization convergence even when the updates are ubiquitously performed on nodes. In other words, PPE is a robust distributed algorithm.

4.1.2 Necessity of the refinement phase when the distance error is proportional to the real range

In these simulations, we vary the standard deviation of $noise_{ij}$, σ_n, and use different distributions for $noise_{ij}$ to show the improvement in the refinement phase over that in the pre-refinement phase.

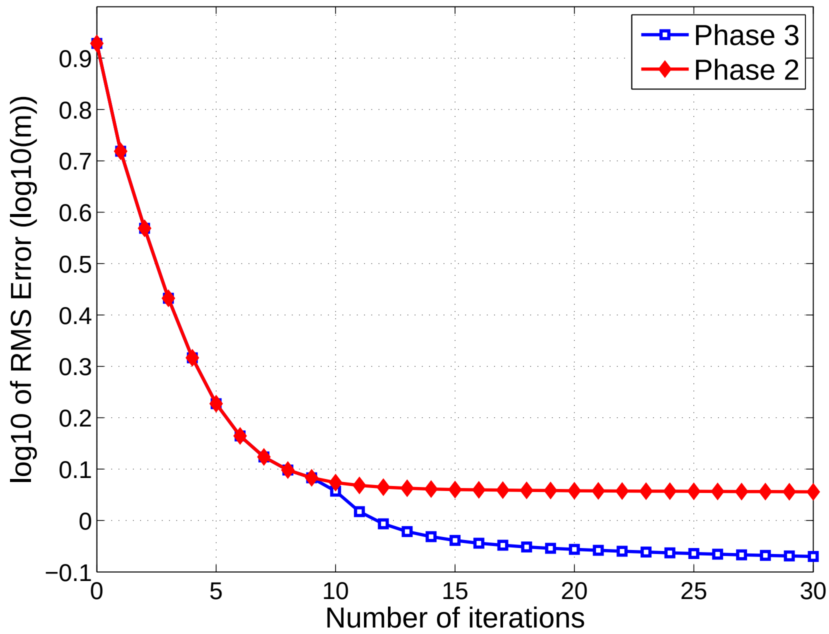
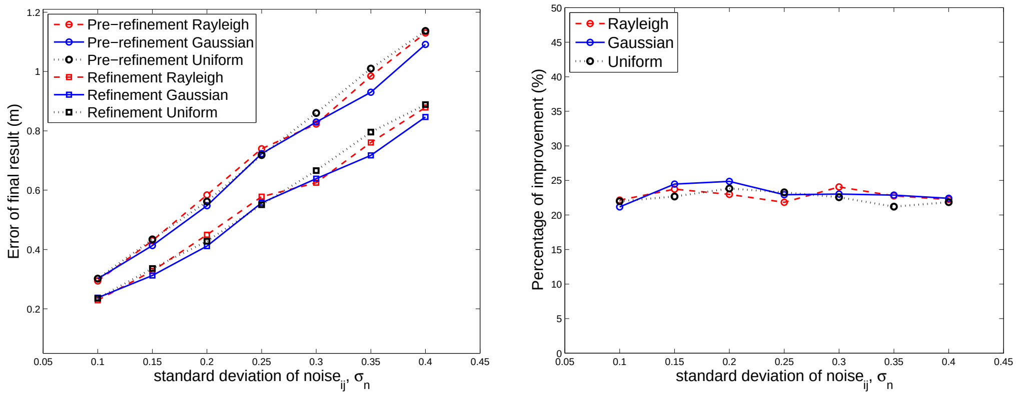


Figure 4.4: RMS error of the estimated position over 30 iterations, the curve with diamonds represents the result of using the pre-refinement phase only while the curve with squares represents the result of refinement phase after ten iterations. Error is in log10 of meters.

After the first ten iterations of phase 2, the comparison between phase 2 and phase 3 occurs in the next 20 iterations. Apparently, the analysis in Section 2.1.3 is verified with the lower RMS error (see (2.12)) for phase 3 as shown in Fig.4.4. Higher accuracy is gained with a very simple but efficient step which eliminates the distance factor from the force model.

Fig.4.5a displays the average error result of 20 trials when phase 3 is used and not used after the 10^{th} iteration. Gaussian, Rayleigh, and Uniform distributions for $noise_{ij}$ are used, and 30 total iterations take place after the completion of phase 1. Although Uniform distributions for $noise_{ij}$ are not appropriate in reality, we use it to emphasize that PPE does not depend on any specific distribution. Meanwhile, Fig.4.5b plots the percentage of the improvement of phase 3 over that of phase 2. Generally, the error reduction is around 26%, a significant improvement. Fig.4.5a



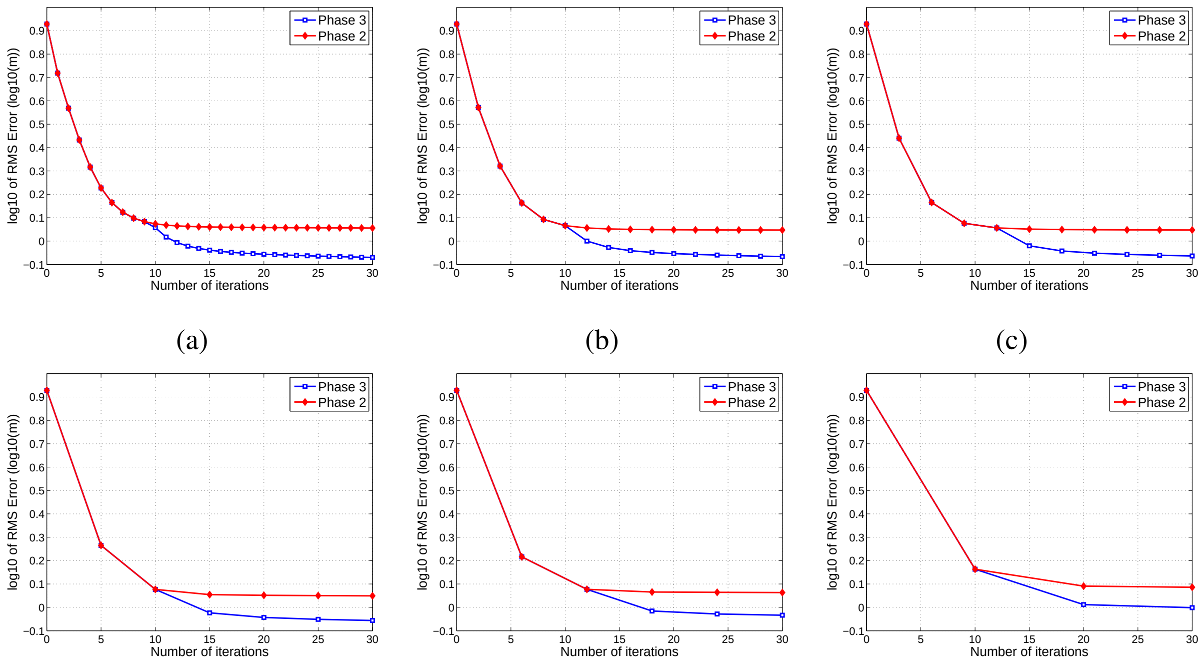
(a) (b)

Figure 4.5: Error and accuracy improvement over σ_n, (a) Result error with and without refinement phase. (b) Percentage of improvement when the refinement phase is used.

does not only demonstrate the linear relation between the standard deviation σ_n of $noise_{ij}$ and the RMS error of the solution, but also the independence of the result on the distribution of $noise_{ij}$. These consequences are consistent with (2.34), (2.38), and (2.39), in which the variance of the mean-force is linearly proportional to the variance of the individual force and does not depend on the distribution of the individual force. This independence is an important advantage point of PPE whereas other successive refinement methods like MLE [47][84] or MDS variations [8][15][44] depend strongly on the noise model. For these approaches, the formula of the measurement error distribution must be known in order to derive the formula of the result error and then to find the calculation to force this error to a minimum.

4.1.3 Communication reduction

This experiment is performed to examine the ability of PPE to reduce communication cost. Each sub-figure in Fig.4.6 is the average result of 20 trials, with the total iteration number of each trial being 30 (iterations of phase 1 are not included). The sub-figures correspond to the different cycle request values R: 1, 2, 3, 5, 6, and 10. Fig.4.6 again shows that the convergence of PPE



(a) (b) (c)

(d) (e) (f)

Figure 4.6: Illustration of communication cost reduction when the cycle request number R = 1, 2, 3, 5, 6, and 10, alternately.

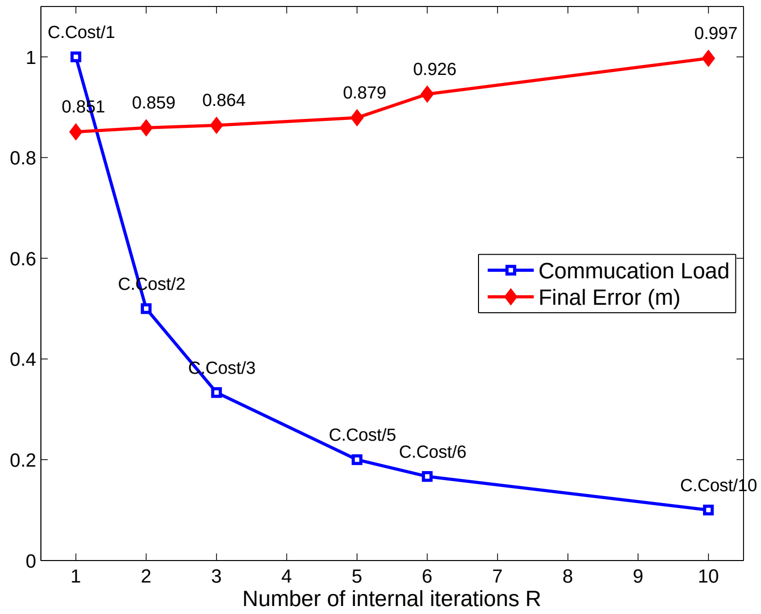


Figure 4.7: Communication cost and final RMS error over the number of iterations. "C.Cost" is the maximum communication cost when the cycle request number R is 1.

is quite independent of the current position at the reference nodes. Apparently, achieving high accuracy incurs a trade-off of the number of iterations and the communication cost. Nonetheless, it is worth reducing the communication cost, the most critical cost, especially when the trade-off error increases insignificantly as in Fig.4.7. Increasing the cycle request number R is very efficient especially when this number is not very high (2, 3 or 5). The RMS error then increases insignificantly (0.94$\%$, 1.52$\%$ or 3.2$\%$) while the communication cost decreases remarkably (50$\%$, 75$\%$ or 80$\%$) compared to the error when a data exchange occurs with every node position update ($R = 1$). Even when the number of internal iterations between two successive data exchanges is 9 ($R = 10$), the error would increase only 17.1$\%$. Hence, based on the specific requirement of designed networks in terms of communication cost and accuracy, an appropriate R can be chosen.

4.1.4 Comparison to MLE and dwMDS via simulations

The contemporary popular MLE and dwMDS methods aim to minimize the stress function [15][47]

$$S = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} w_{ij}(\tilde{d}_{ij} - \delta_{ij})^2, \tag{4.1}$$

where w_{ij} is the weight value.

For the dwDMS (distributed weighted Multi Dimensional Scaling) simulation, we rely mainly on Costa's paper [15] and a part of the Matlab code available on Patwari's website [60] to solve our problems and compare the results with our PPE's. For MLE (Maximum Likelihood Estimation), the updating method is the nonlinear conjugate gradient. dwDMS is actually a two-step method in which the first step is raw estimation, and the second is refined estimation. However, the convergence time of this algorithm is large; the result of the first step suffers from negative estimation (the estimated positions make the deployed area look shrunk); and the accuracy is poor especially when the measurement error has the log-normal model. Meanwhile, MLE's estimation would hardly converge with random initial positions of the normal nodes. Therefore, we take the output of PPE's phase 1 as the raw estimation inputs for both MLE and dwDMS although this unbiased estimation is one of the important advantages of PPE over other algorithms. Especially for the case of log-normal distance error, we use the actual positions of all nodes as the input for the second step of dwDMS so that the simulation time is smaller and the accuracy is higher. Even so, the simulation results illustrate that the performance of PPE is superior to those of the other methods.

Measurement error proportional to the real distance case

The measurements are generated according to the noise model in (2.2) where $noise_{ij}$ is Gaussian for comparison purpose. σ_n varies from 0.1 to 0.4 in intervals of 0.05, as in Fig.4.8. We use the same set of actual node positions, and with each value of σ_n, we create 20 sets of measurement data based on the error distribution. For dwDMS and MLE, the algorithms stop if the maximum

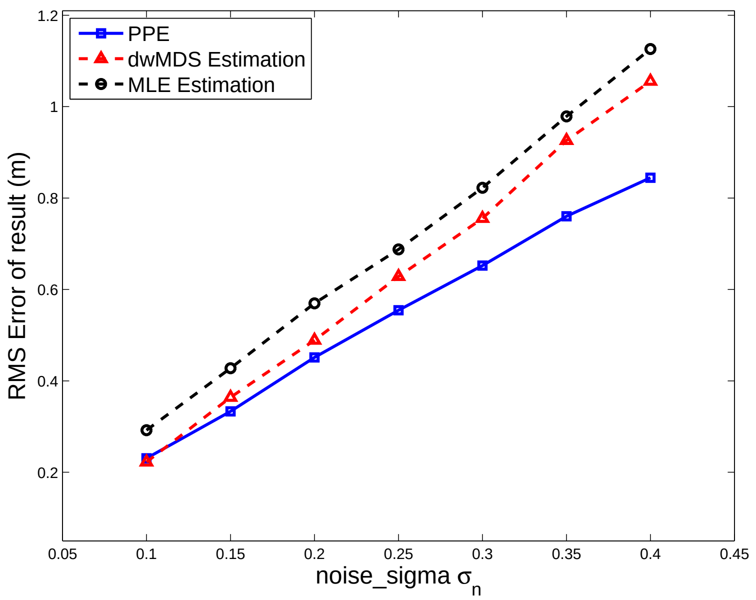


Figure 4.8: RMS errors of different methods over the standard deviation σ_n of $noise_{ij}$, for the error model in equation (2.2)

iteration number reaches 80 or if the average updating change of the stress function is less than 10^{-4}. The stop conditions for PPE are $k^{(1)}_{max} = 10$, $k^{(2)}_{max} = 10$, $k^{(3)}_{max} = 30$, or the movement distance is less than 10^{-4}m. Fig.4.8 shows that PPE gives the lowest RMS error at every value of σ_n. This means that PPE has a better performance even when dwDMS and MLE use the unbiased raw estimation of PPE's phase 1. Moreover, PPE also attempts to eliminate d_{ij} from the error, turning the variance of $d_{ij}.noise_{ij}$ into the smaller variance of $\overline{d_{ij}}.noise_{ij}$ (see (2.40)). As a consequence, the accuracy of PPE in the range-based scheme whose error of measurement is modeled in (2.2) is better.

Table.4.5 shows the approximated computation costs of each normal node for three methods in the simulation. The values in the second column are the approximated numbers of multiplications in one iteration. We consider a square-root operation as a multiplication and ignore several extra multiplications outside the loops. The values in the third column are the average numbers of

Table 4.2: Computation cost.

	Multiplications per iteration	Iteration number
dwDMS	$3(M_i^{(3)})^2 + 13M_i^{(3)}$	80
MLE	$7M_i^{(3)} + 3M_i^{(3)} * Iter_0$	52
PPE	$5M_i^{(3)}$	40

iterations necessary to produce the final result when the input data is the raw estimation. $Iter_0$ is the extra number of iterations used in MLE for computing the line search optimization. The average of this number or $mean(Iter_0)$ is 5.6. Table.4.5 also shows that PPE requires fewer iterations than do dwDMS and MLE to reach a better result. In other words, PPE needs less of both communication and computation costs even when the cycle request number in this simulation is 1.

Log-normal distribution for the measurement error case

For this part, sets of measurements modeled in (2.49) are used instead of those in (2.2). σ_p is the standard deviation of $Pnoise$ and it is varied from 1 to 4 in increment of 0.5. We repeat that with log-normal error model, only two first phases of PPE with the de-biasing function $log_{10}(.)$ are used. MLE has the same stop condition as in the previous experiment while PPE stops at $k_{max}^{(1)} = 20$, $k_{max}^{(2)} = 40$, or when the movement distance is less than 10^{-4}m. It should be noted that the input data for PPE does not include the information of normal nodes' locations, while dwMDS uses the real node positions as the raw input and MLE uses PPE's raw estimation provided by phase 1. Therefore, for dwMDS, we set the maximum number of iterations to be 20 in this case because a small value of this number leads to small final error. However, it can be seen in Fig.4.9 that dwMDS gives the worst error compared to those of PPE and MLE which have equivalently good result at small σ_p. Although MLE has the best result when $\sigma_p = 1$, MLE's accuracy decreases faster than that of PPE while PPE's error maintains a linear relation with σ_p

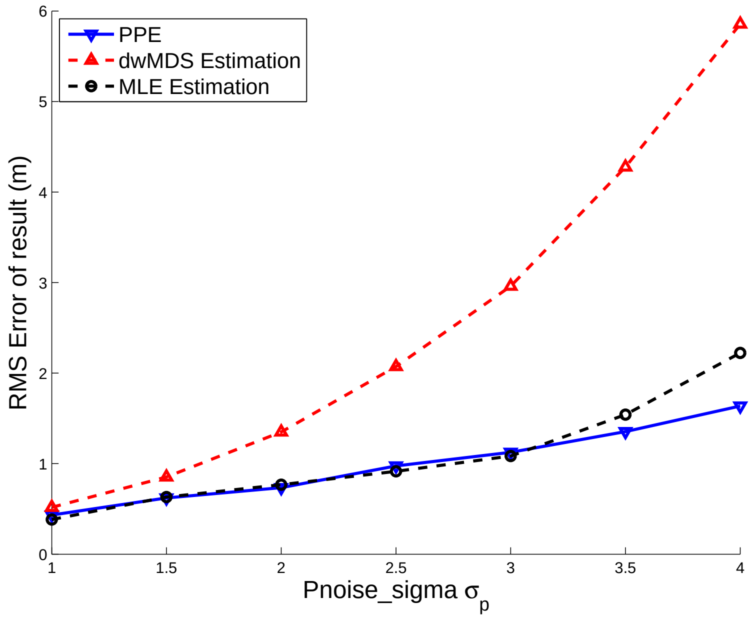


Figure 4.9: RMS errors of different methods over the standard deviation σ_p of $Pnoise$, for the error model in equation (2.49).

when σ_p is large ($\sigma_p > 3$). Obviously, MLE is more sensitive to the big log-normal noise. Even with the same raw estimation, and when the stop conditions are tougher for a higher accuracy, MLE is not as good as PPE when σ_p increases. It is known that large values of σ_p are the case of the practical implementation, especially for indoor settings.

4.1.5 Comparison to MLE and dwMDS with real data

In this part, we perform the comparisons among PPE, dwMDS and MLE with the real data which is available on Patwari's website [60]. The data sets are the RF power measurements and the distance measurements under TOA scheme. Both of data sets are obtained from the test-bed in which there are 4 beacons and 40 normal nodes in the area of [-4m,10m]x[0m,12m]. The beacons are around the corners of the deployed area and are denoted with the red diamond nodes in Fig.4.10 and Fig.4.11. To visualize the accuracy of the proposed algorithm, we draw lines connecting the

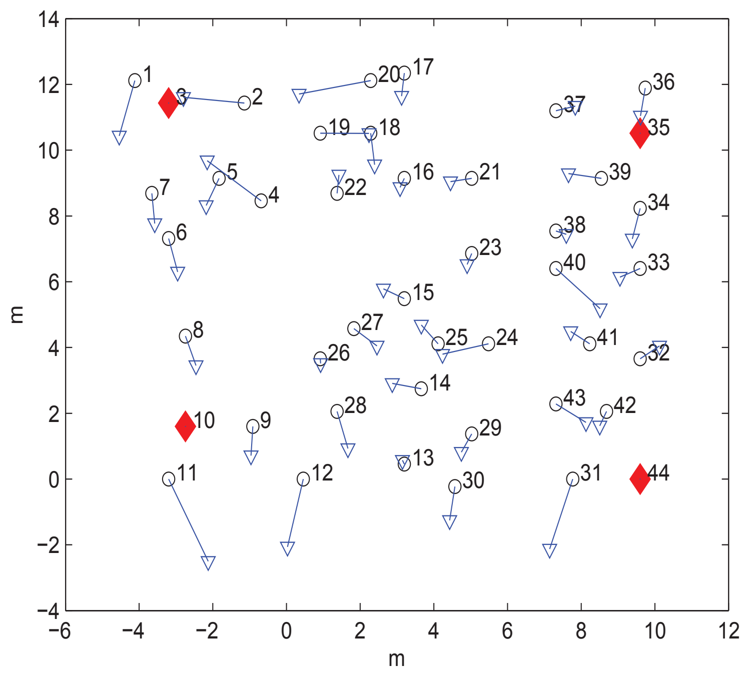


Figure 4.10: PPE's result based on TOA real data [60] for 4 beacons and 40 normal nodes, RMSE = 1.1028m.

estimated and actual coordinates of sensors respectively. The real locations of the normal nodes are denoted with round dots while their estimated positions are denoted with triangle dots.

Fig.4.10 displays the final estimation result of the normal nodes' locations when the input data is the distance set obtained from TOA data. Meanwhile Fig.4.11 is the final result of sensor location estimations where the distance is estimated from the RF power measurements. The de-biasing function in (2.48) is used to improve our previous work [45]. The statistical results of the two figures are in Table.4.3. They are also for the comparison purpose with other popular works such as Standard MDS, dwMDS and MLE. For RSS data, MLE's result is also comparatively as good as PPE's result, but for TOA data set, its result is not as good as dwMDS. Consistent with the analysis and simulation results above, it can be seen in Table.4.3 that PPE gives the best results for both RSS and TOA data sets.

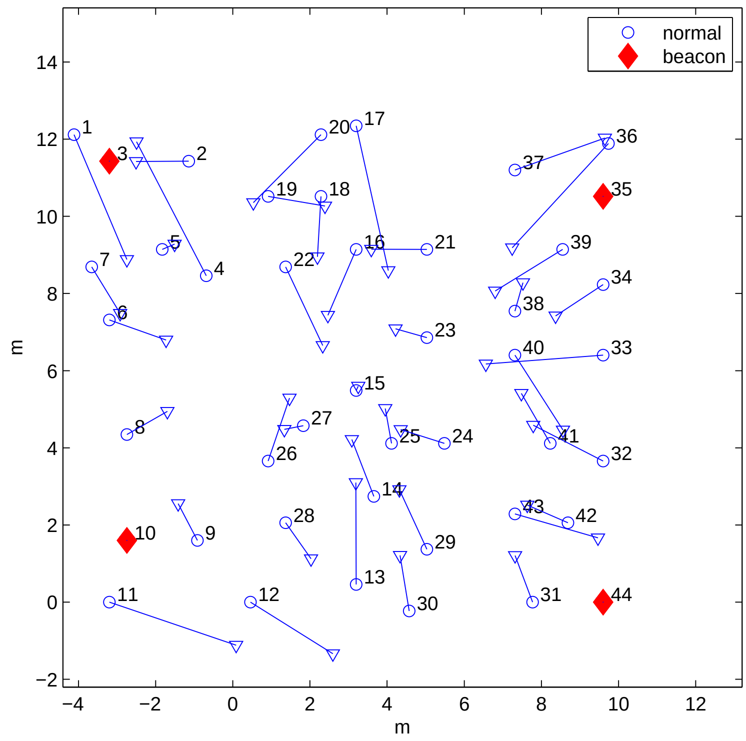


Figure 4.11: PPE's result based on RSS real data [60] for 4 beacons and 40 normal nodes, RMSE = 2.1688m. Triangle dots are estimated position

Table 4.3: The RMSE result comparison among PPE with other popular methods of localization estimation in WSNs.

	Clasical MDS	MLE	dwMDS	PPE
RSS	4.26m	2.18m	2.48m	2.17m
TOA	1.85m	1.23m	1.12m	1.10m

4.2 Experiments and discussions on Acoustic Based Multi-Object Positioning

For the system working demonstration and the system evaluation, two main experiment sets are conducted via simulations in this section. They will be presented in details after we introduce the experiment setup and modeling in the next subsection. At the end of this section, system working is demonstrated with a real data set. Moreover, a comparison table is presented to highlight the proposed method's superiorities.

4.2.1 Experiment setup and modeling

The monitored area is within [0m,12m]x[0m,12m]. We generate five simulated sources ($M = 5$), four of which imitate the sounds of different vehicles and motors while the rest mimics the sound of a siren (see Fig.4.12). They are parametrically determined so that the received signals affected by Doppler effect can be generated properly using equation (3.1). These continuous sources have equivalent power levels, none have overlapped dominant f-components, and each is at least 5 meters from the others. It is reasonable to consider a group of close sources as one sound source; thus in order to illustrate how the system works, it is necessary for the sources to be distinct from one another. If the system functions properly when the sources are separated, then the source characteristics can be determined, memorized and then used for future localization even when they are close to one another. The independence condition that requires no overlapped f-components among the sources is an elastic requirement. If the interference of other f-components is small, then it is considered as noise. In order to illustrate this point, we let one source, the second one in Fig.4.12, have a wide spectrum which interferes into other frequency segments. The wave files of received signals are available on the website [16] as examples where the source speed is 40km/h in 3 seconds at the highest noise level used in these simulation sets. Four sensors ($N = 4 < M$) are deployed around the corners of the deployed area in which the sources are set randomly (see

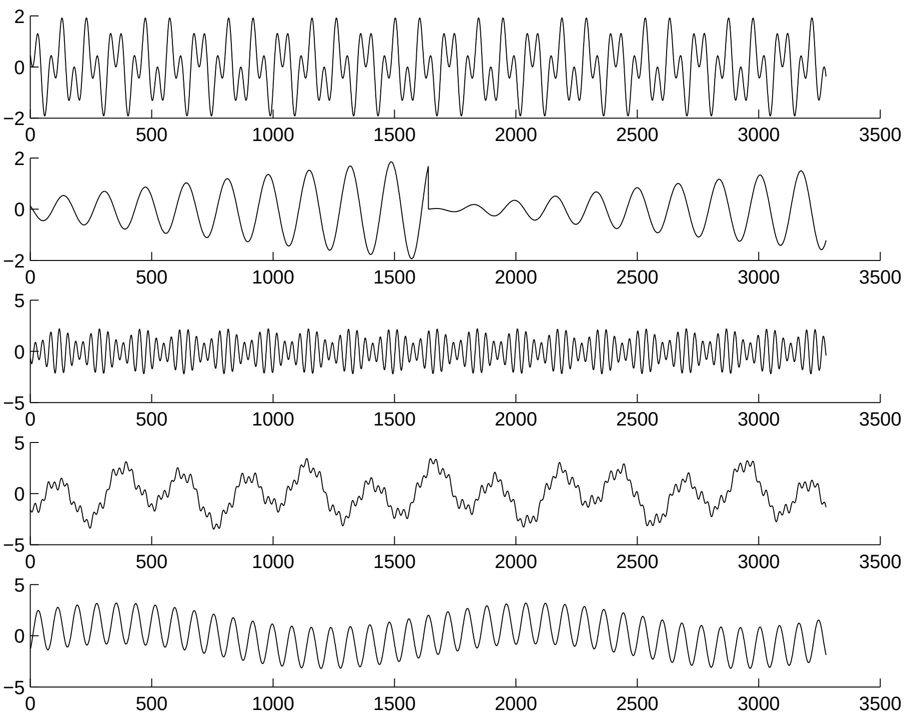


Figure 4.12: Signals of sources which are parametrically determined to imitate sounds of vehicles, motors and a siren.

Figure 4.14). The locations of sensors are not based on previous estimation but are indicated exactly. A source can move into and out of the monitored area during the monitoring time. The energies of the line-of-sight signals propagating to the sensors decrease according to the inverse square law at the sound speed of $c = 343m/s$. The sampling frequency is $F_s = 16.384KHz$ and the time segment length is not less than 0.2 seconds. Because the sound signals of vehicles and motors are mainly composed by low frequency components, F_s and time segment T_f are set to be high so that different f-components can be separated as much as possible. The background noise is considered to be produced by the surrounding environment and the microphones, thus the noise is chosen to be Gaussian and its level is the same at all sensors.

In reality, received acoustic data always includes the effects of shadowing and fading due to multiple path reflections besides the received line-of-sight signals and Doppler effect. Therefore, we examine the situation under the existence of a Rayleigh fading channel and consider Rayleigh multi-path signals as another kind of noise, instead of taking the reverberation model which is often used for indoor settings [23][26][36][82][83]. Generating multiple paths for each source takes much computing time especially when Doppler effect is present, due to: (a) for each source, a large number of random paths are needed and for each path a new signal with respect to the direction of the path has to be generated, and (b) for each sensor, we have to generate different sets of simulated data as in (a). To reduce the simulation time for generating input data, the Young model [39], which generates a Rayleigh channel with two arrays of Gaussian random variables and the inverse-DFT (IDFT) technique, is applied. The model in [39] is for single f-component signals whereas the signals in this simulation set are multi f-component signals. Moreover, different f-components have different Doppler shift ranges and the Rayleigh energy in a range must be statistically proportional to the energy of the corresponding f-component. Therefore, in these simulations, after the ranges of Doppler shifts are determined, for each frequency bin within the range, a complex value is generated. The value's real and imaginary parts are Gaussian variables with the same standard deviation. If the ranges overlap, then the number of generated complex values is the number of overlapped ranges, and the Rayleigh noise at the bin is the sum of these values. In order for the result of IDFT to be real, the array representing the Rayleigh fading effect on the frequency domain $R(k)$ must satisfy

$$\begin{cases} R\left(0\right) = 0 \\ R\left(k\right) = R^{*}\left(Q-k\right), \quad k = 1,..,Q-1 \end{cases}, \tag{4.2}$$

where the asterisk denotes the conjugate of the complex number. This makes the magnitude of the frequency frame $|R(k)|$ symmetrical over the dash line at $k = Q/2$ (see Fig.4.13). Fig.4.13 illustrates the generated Rayleigh fading signal on the frequency domain. The higher is the frequency of the f-component, the wider is the Doppler shift range and the smaller are the frequency magnitudes within the spread range.

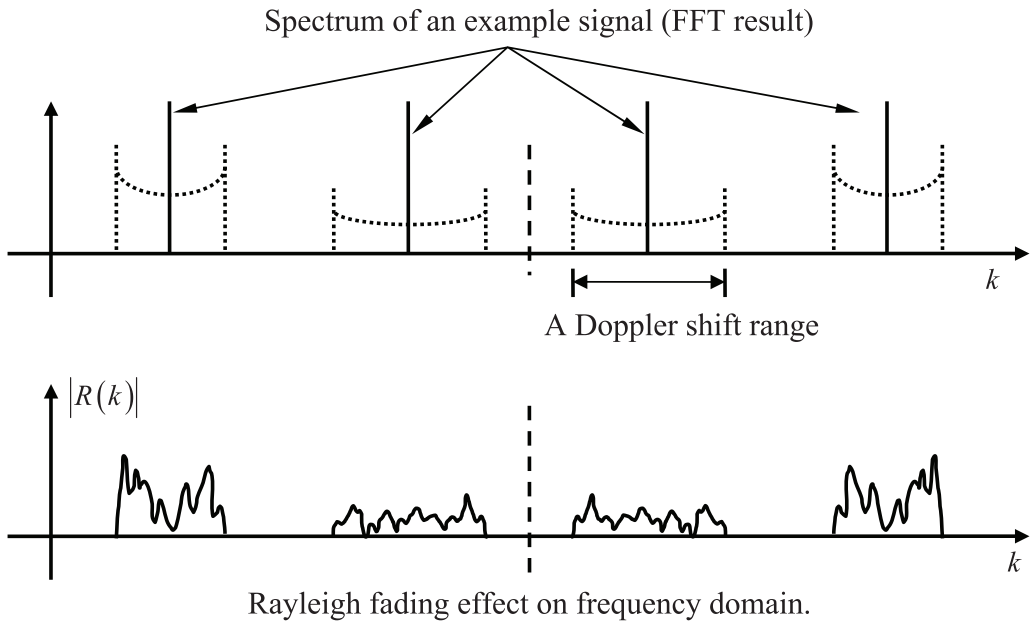


Figure 4.13: Each bin within a Doppler shift range is generated with two Gaussian random variables to produce Rayleigh fading effect.

We increase the level of the total noise and choose the parameter Signal to Noise Ratio SNR for evaluating noise influence.

$$\begin{aligned} SNR &= \frac{E_{mean}}{E_{noise}} \\ &= \frac{E_{mean}}{\alpha_{RL} E_{Rayleigh} + (1-\alpha_{RL}) E_{Background}}. \end{aligned} \tag{4.3}$$

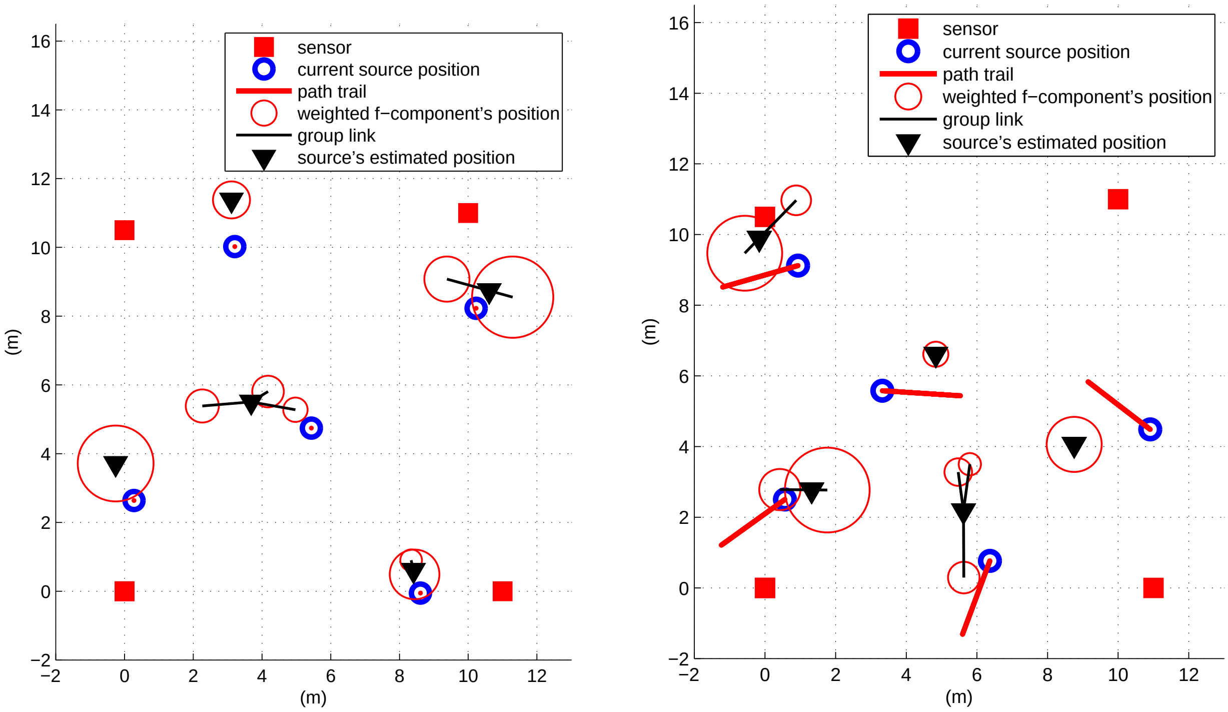
where E_{mean} is the mean value of the average signal energies received at the four sensors, E_{noise} is the sum noise energy of the background noise $E_{Background}$ and the Rayleigh noise $E_{Rayleigh}$, while $\alpha_{RL} \in (0,1)$ represents the percentage of Rayleigh noise energy in the total noise energy.

4.2.2 System working demonstration

Four sensors in the deployed area are used to record data transmitted from fives sources, as described in Subsection 4.2.1. The processing steps of the system are basically consistent with those previously mentioned. The recorded data is sampled and segmented into overlapped frames. In this thesis, we do not concentrate on continuously monitoring and refining the position estimations but only on positioning. Therefore, overlapped frames are not meaningful in these simulations. Each frame is multiplied by a Hamming window whose length is equal to the frame length (see (3.18)). DFT transformation is applied to convert data to the frequency domain in which the Gaussian noise filtering occurs. The filtering task can be conducted by detecting the maximum value of the frequency magnitude in the high frequency region having no f-components from the sources. Another simple method to detect Gaussian noise level is to determine the maximum magnitude of f-component on the frequency domain in advance when no sources are monitored. After the "F-component Positioning" stage, each f-component's position is determined and plotted with a circle. The radius of the circle is proportional to the f-component's weight or to the mean energy that the sensors receive from that f-component. Those f-components whose estimated positions are close to one another are grouped together as described in 3.3.2, in which $d_{min} = 3.5m$ (new cluster is generated if an f-component's location is more than d_{min} from all others).

Table 4.4: Percentages (%) of resulting clusters when $T_f = 0.2$s and $\alpha_{RL} = 0.2$.

SNR \ Speed (km/h)	0	8	16	24	32	40
	Percentages of resulting into 5 clusters					
∞, (no noise)	91.7	85.9	86.8	84.3	84.1	86.8
40.21	89.5	83.2	80.6	83.5	85.1	81.6
10.05	86.5	79.3	77.7	74.4	79.0	76.8
4.46	72.6	71.5	67.1	67.5	67.8	69.4
2.51	66.3	64.8	63.7	62.6	63.2	64.8
	Percentages of resulting into 4 clusters					
∞,(no noise)	6.4	9.2	10.2	11.9	11.7	8.9
40.21	8.4	12.2	15.3	12.4	10.5	13.1
10.05	10.0	14.4	16.4	16.7	14.1	14.0
4.46	17.8	18.5	18.3	19.7	19.1	17.6
2.51	19.4	20.9	21.0	19.4	19.34	19.6
	Percentages of resulting into 6 clusters					
∞,(no noise)	1.9	4.9	2.7	3.6	4.1	4.2
40.21	2.1	4.6	3.9	3.8	4.1	4.8
10.05	2.9	5.0	4.8	7.2	6.4	8.2
4.46	9.1	8.8	13.5	11.4	11.0	11.5
2.51	12.6	13.0	13.7	16.3	15.1	14.2



(a) (b)

Figure 4.14: (a) and (b) are the estimation results of two examples using the highest level of noise in the simulation set and $\alpha_{RL} = 0.2$. One (a) locates five sources with source speeds of zero and the other (b) positions five sources with source speeds of 40km/h.

Fig.4.14 demonstrates the results of source positioning when the system tries to localize five independent sources in the time segment T_f of 0.2 seconds and the Rayleigh fading noise energy contributes 20% of the total noise energy. The path trails that the sources leaves within the time segment of 0.2 seconds are denoted with the bold lines. The estimated positions of the sources are calculated based on the groups' f-component positions and their weights according to equations (3.24) and (3.25). Fig.4.14a shows the position estimation results when the speeds of the sources are zero, so the path trails are the small dots. Meanwhile, Fig.4.14b presents the position estimation result when the speeds of the sources are all $40km/h$. It can be seen that the system is able to locate the sources even when the source number is greater than the sensor number. The Root Mean Square Errors (RMSEs) for the successful clustering in both Fig.4.14a and Fig.4.14b are less than 1.6 meters, an acceptable level especially when the speeds of the sources are high (in 0.2s, the trail lengths are around 2.2m). Obviously, positioning based on f-component localization is a good method to deal with multiple acoustic source positioning in situations affected by high Gaussian noise, multi-path fading and Doppler effect. The SNR here is 2.51, the highest level of all simulations.

To demonstrate the results of clustering, Table.4.4 shows the percentages of the group number resulted after f-component positions are clustered in the same condition as in the above example ($T_f = 0.2s$, $\alpha_{RL} = 0.2$). This task is repeated for 1000 trials for each pair of total noise level and source speed. Since we focus on statistically evaluating the performance of the system during positioning, we do not try to monitor the sources in sequences of frames or to refine the location estimations over time. Instead, for each trial, we randomly set the positions and the moving directions of the sources and let the system perform positioning with only one frame. As one can expect, the percentages of exactly grouping into 5 sources decreases with the rise of noise and source speed, while the percentages of resulting in 4 and 6 groups after clustering increase. However, in worse case, the percentage of clustering into 5 groups is still high, not less than 62%.

4.2.3 System performance evaluation

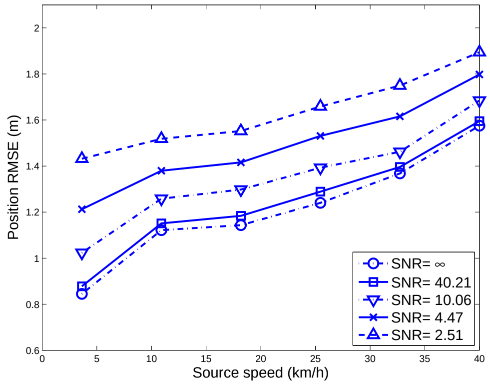
This simulation aims at examining the influences of time segment T_f, signal-noise ratio SNR and speed v_j on the results of positioning. The chosen T_f is either 0.2s or 0.25s, while the speed value varies from 0 to 40km/h in increments of 8km/h. The signal-noise ratios are generated based on the linear increment of the standard deviation of Gaussian noise. The set of SNR values are maintained statistically the same for the purpose of comparison, even when T_f and α_{RL} vary.

Fig.4.15 shows the results of distance errors under influences of noise level SNR, the percentage of Rayleigh multi-path fading noise α_{RL} and the speed of sources v_j when the time segment is 0.2 seconds. Meanwhile, Fig.4.16 illustrates the results when the time segment is 0.25 seconds. Each plotted error is the average value of RMSEs of 1000 trials. At each trial, the RMSE is computed from the shortest distances between the estimated source positions and the actual source positions. When the number of estimated positions is different from that of actual positions, the number of distance pairs used to compute RMSE is the smaller number.

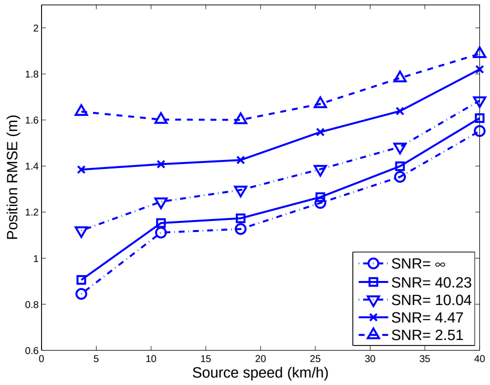
Table.4.5 shows the percentage results for clustering into 5 groups, 4 groups and 6 groups at different conditions of noise levels and different source speeds. It is constructed when the $T_f =$ 0.2s and $\alpha_{RL} = 0.8$, corresponding to Fig.4.15d. Meanwhile, Table.4.6 shows the percentage results of clustering with the same format as in Table.4.4 and Table.4.5 when the $T_f = 0.25$s and $\alpha_{RL} = 0.8$, corresponding to Fig.4.16d.

From the results, it can be seen that at any speed value, higher noise and higher velocity lead to higher source positioning error. The higher noise level results in more error in the f-component positions due to the increased error in the constraint ratios, especially if the f-components have low magnitude. Meanwhile, higher speeds lengthen the pathtrails, increasing the uncertainty of positions. Consequently, the error in the f-component positions affects the f-component clustering results, causing error in the final source position estimations.

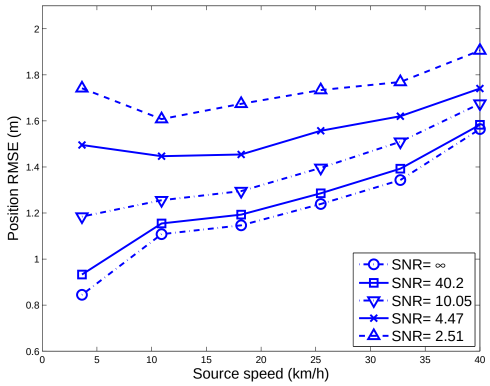
The relationships between v_j and RMSE can be considered to be positive linear when v_j is high (see Fig.4.15 and Fig.4.16). It is obvious since higher speed sources leave longer path



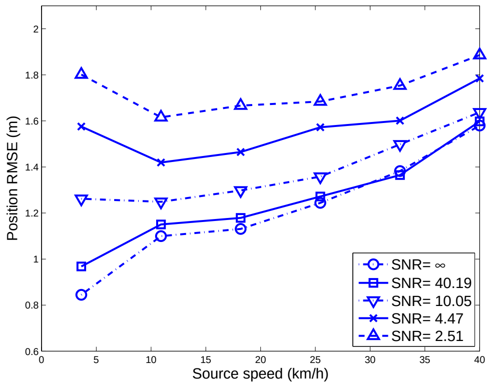
(a) $\alpha_{RL} = 0.2$



(b) $\alpha_{RL} = 0.4$



(c) $\alpha_{RL} = 0.6$



(d) $\alpha_{RL} = 0.8$

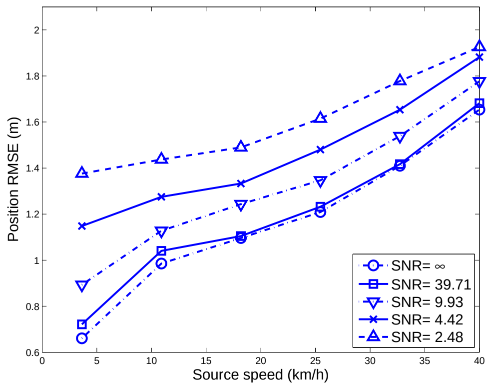
Figure 4.15: RMSE error results of source locations under influences of speeds of sources, Gaussian noise and Rayleigh multi-path fading when the time segment is 0.2s.

Table 4.5: Percentages (%) of resulting clusters when $T_f = 0.2$s and $\alpha_{RL} = 0.8$.

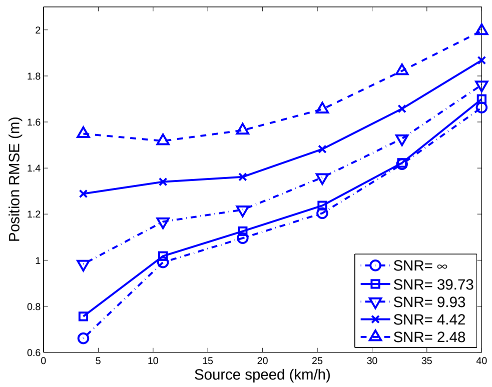
SNR \ Speed (km/h)	0	8	16	24	32	40
	Percentages of resulting into 5 clusters					
∞, (no noise)	91.7	86.3	85.7	85.0	84.9	86.1
40.19	86.8	84.1	81.3	82.1	81.7	83.2
10.05	73.4	75.4	74.0	73.8	74.0	76.6
4.47	58.1	65.1	61.9	66.3	67.0	65.2
2.51	52.5	55.3	58.5	60.3	61.1	58.4
	Percentages of resulting into 4 clusters					
∞,(no noise)	6.4	9.2	10.8	10.6	10.9	10.0
40.19	11.0	11.9	14.3	13.3	13.8	12.4
10.05	19.5	18.4	18.3	19.3	19.0	17.9
4.47	25.0	23.4	27.4	22.0	21.4	22.8
2.51	24.9	27.0	25.1	23.3	24.9	25.7
	Percentages of resulting into 6 clusters					
∞,(no noise)	1.9	4.3	3.3	4.2	3.7	3.6
40.19	2.0	3.6	3.7	4.5	4.3	4.2
10.05	5.9	5.1	6.7	5.7	6.4	4.5
4.47	13.1	9.5	8.2	9.7	8.9	9.7
2.51	16.7	13.6	13.0	12.9	10.6	11.7

Table 4.6: Percentages (%) of resulting clusters when $T_f = 0.25$s and $\alpha_{RL} = 0.8$.

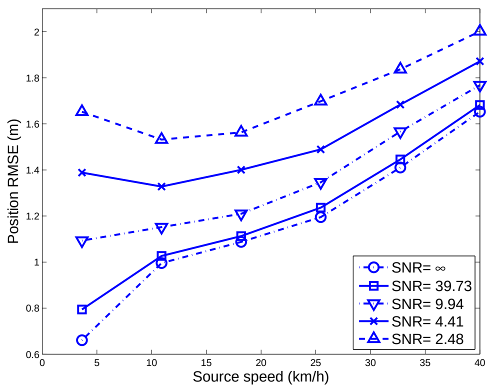
SNR \ Speed (km/h)	0	8	16	24	32	40
	Percentages of resulting into 5 clusters					
∞, (no noise)	94.3	85.3	86.7	87.1	88.4	89.7
39.71	90.2	82.3	86.0	84.8	86.1	87.7
9.93	70.3	75.8	77.5	76.5	78.6	77.7
4.42	60.4	67.3	67.2	68.6	69.2	68.2
2.48	50.1	57.5	59.4	61.0	63.4	65.0
	Percentages of resulting into 4 clusters					
∞,(no noise)	5.6	13.3	11.8	11	9.1	8.8
39.71	8.8	15.3	12.1	13.2	11.0	9.9
9.93	19.5	18.2	16.1	17.1	14.9	16.3
4.42	21.3	19.8	20.3	21.3	18.4	21.5
2.48	23.2	21.3	22.8	20.5	21.3	20.6
	Percentages of resulting into 6 clusters					
∞,(no noise)	0.1	1.2	1.3	1.7	2.1	1.4
39.71	1	1.9	1.7	1.4	2.8	2.3
9.93	8.2	5.6	5.6	5.5	5.2	5.7
4.42	14.5	10.6	10.8	8.7	10.4	9.3
2.48	22.6	17.7	14.9	15.7	12.1	10.9



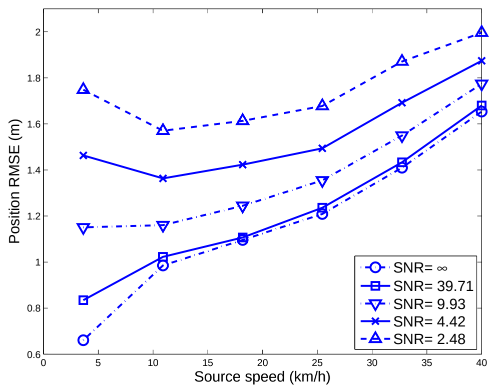
(a) $\alpha_{RL} = 0.2$



(b) $\alpha_{RL} = 0.4$



(c) $\alpha_{RL} = 0.6$



(d) $\alpha_{RL} = 0.8$

Figure 4.16: RMSE error results of source locations under influences of speeds of sources, Gaussian noise and Rayleigh multi-path fading when the time segment is 0.25s.

trails. In addition, Rayleigh multi-path fading caused by high v_j affects the accuracy less than that caused by low v_j (compare the sub-figures). The accuracy of estimated f-component positions decreases when the source velocities increase, leading to the increased error in the source location estimations. On the other hand, when the SNR is high and v_j is low, the accuracy decreases rapidly because of Rayleigh multi-path fading. In the case where $SNR = 2.5$ and $\alpha_{RL} = 0.8$, the RMSE error is up to around 1.8m (see Fig.4.15d and Fig.4.16d). An f-component at a low v_j produces noise in a narrow and condensed Doppler shift range on the frequency domain. As a result, at the same level of Rayleigh noise, the received energy of an f-component through the line-of-sight path is corrupted more by a narrow shift range than by a wide shift range, causing poor accuracy at low v_j when there exists Rayleigh fading. Meanwhile, when v_j is high, the energy of Rayleigh fading noise is spread wider and thinner on the frequency domain and less affects the line-of-sight f-component. The fading noise can also be partly filtered by the Gaussian filtering stage. Therefore, Rayleigh fading noise in high v_j cases does not affect the accuracy as much as it does when v_j is low. Generally, the results in Fig.4.15 and Fig.4.16 show that the more Rayleigh fading noise contributes to the total noise level, the worse result the system achieves especially if the source speeds are low.

One can easily notice that the system could not give the ideal result when there is no noise ($SNR = \infty$). The f-component location errors are caused by a limited time segment of a frame which produces unavoidable spectral leakage. In addition, there exists the influence of the second original source, whose f-components appear and contribute interference all over the frequency domain, to other sources' f-components. As a consequence, the errors in final estimations are unavoidable. Nevertheless, when the noise level increases quickly, the system can tolerate the noise well with little error increment. The estimation error increases an amount of 0.6m in average when the SNR decreases 16 times from around 40 to 2.5. Comparing sub-figures by pair from Fig.4.15 and Fig.4.16, we can see that a bigger time segment T_f improves the accuracy when v_j is low (including the ideal case). However, it decreases the accuracy if v_j is high. Obviously, although bigger T_f reduces the spectral leakage which leads to better estimations, this improvement cannot compensate the increased error when the path trails are longer, producing high error results when v_j is high ($>$ 30km/h).

Table.4.5 and Table.4.6 show that f-component clustering relied on f-component locations is closely associated with the estimation accuracy. An incorrect number of clusters is caused by missing f-component locations due to noise filtering or by using too poor the f-component position estimations. Higher percentage of grouping into exactly 5 clusters increases the estimation accuracy. Contrariwise, this percentage decreases at high levels of noise and high source speeds.

Comparing Table.4.4 and Table.4.5 both of which result from the same T_f of 0.2s, one can see that the higher is the contribution of Rayleigh fading noise to the same level of total noise, the worse is the f-component clustering result. This is consistent with the above analysis of Rayleigh fading influence on the RMSE. Meanwhile, comparing Table.4.5 and Table.4.6 shows that in most of the cases, longer monitoring time reveals higher clustering accuracy and the percentage of clustering into 5 groups is higher. This is because most f-components of vehicle and motor sources appear in the low frequency range, so a longer time segment helps separate the f-components more clearly and decrease the spectral leakage. However, increased T_f makes the final estimation results at high source speeds not as good as those at low source speeds (see Fig.4.15 and Fig.4.16) since the path trails at high v_j are longer. This reflects the uncertainty law between time and frequency: low frequency needs a long observation time to be indicated. Obviously, there is a trade-off between f-component separation and the path trail length when T_f varies.

Now we consider the harsh conditions, when data collected from high speed sources under serious noise and when low speed sources move in the setting where Rayleigh fading noise accounts for most of the total noise. Although the percentage of clustering into 5 groups is quite low (in the worse case, only 50.1%), the percentage of clustering results for all 4, 5 and 6 groups are very high. Table.4.7 displays the high values of total percentage of clustering for 4, 5 and 6 groups in three critical conditions in this simulation set (the least is 94.1%, see Table.4.7), indicating that

Table 4.7: Total percentages (%) of clustering 4 groups, 5 groups and 6 groups

SNR \ Speed (km/h)	0	8	16	24	32	40
	$T_f = 0.2s$; $\alpha_{RL} = 0.2$					
∞, (no noise)	100.0	100.0	99.7	99.8	99.9	99.9
39.71	100.0	100.0	99.8	99.7	99.7	99.5
9.93	99.4	98.7	98.9	99.4	99.5	99.0
4.42	99.5	98.8	98.9	98.6	97.9	98.5
2.48	98.3	98.7	98.4	98.3	97.6	98.6
	$T_f = 0.2s$; $\alpha_{RL} = 0.8$					
∞,(no noise)	100.0	99.8	99.8	99.8	99.5	99.7
39.71	99.8	99.6	99.3	99.9	99.8	99.8
9.93	98.8	98.9	99.0	98.8	99.4	99.0
4.42	96.2	98.0	97.5	98.0	97.3	97.7
2.48	94.1	95.9	96.6	96.5	96.6	95.8
	$T_f = 0.25s$; $\alpha_{RL} = 0.8$					
∞,(no noise)	100.0	99.8	99.8	99.8	99.6	99.9
39.71	100.0	99.5	99.8	99.4	99.9	99.9
9.93	98.0	99.6	99.2	99.1	98.7	99.7
4.42	96.2	97.7	98.3	98.6	98.0	99.0
2.48	95.9	96.5	97.1	97.2	96.8	96.5

a good inference mechanism module can be used in conjunction with this positioning system to combine the information of cluster number and the relationships between f-components and clusters. The continuous combination over time can better clustering results, measure the velocities of the sources and refine the position estimations by predicting their next positions. This module, however, requires much effort in both analysis and design, and is beyond the scope of this thesis.

4.2.4 System demonstration with real data and comparison to other works

In this section, we conduct an experiment to demonstrate the system's performance with the real data and present some related problems when dealing with real-world data. We deployed 4 microphone sensors, which are marked as $M1$, $M2$, $M3$ and $M4$, at a segment of a four-lane road as depicted in Fig.4.19. Their coordinates are set as in Fig.4.20 and the coordinate system is organized so that the third sensor M3 is at the origin of the system. Four sensors are at the points of the rectangle (0,0),(30,0),(30,17.4) and (0,17.4). Fig.4.17 is an example of a real collected data while Fig.4.18 is the Fourier properties of a segment of 1000 samples extracted from the collected data example. We try to record the sound data with four pairs of microphone and laptop at the same time and the recorded programs are performed in Matlab environment. In this experiment, we choose only one good data set for demonstrating the system's performance and the data in Fig.4.17 is actually the chosen one caused by two moving sources whose frequencies are most independent.

Fig.4.20 is an example when the system tries to find the positions of two sources (a car and a motobike) within a frame of 1000 samples, (sampling frequency $F_s = 32000Hz$). The microphone sensors are plotted with red triangles. The contour is for objective function representation in 2-D space. In this demonstration, we use simple negative gradient method in order to highlight the system's performance to find the positions that compromises all the constrains. Final estimated positions are marked with the text 't_1'.

The final result of the estimation is displayed in Fig.4.21 where the black circle dots represent

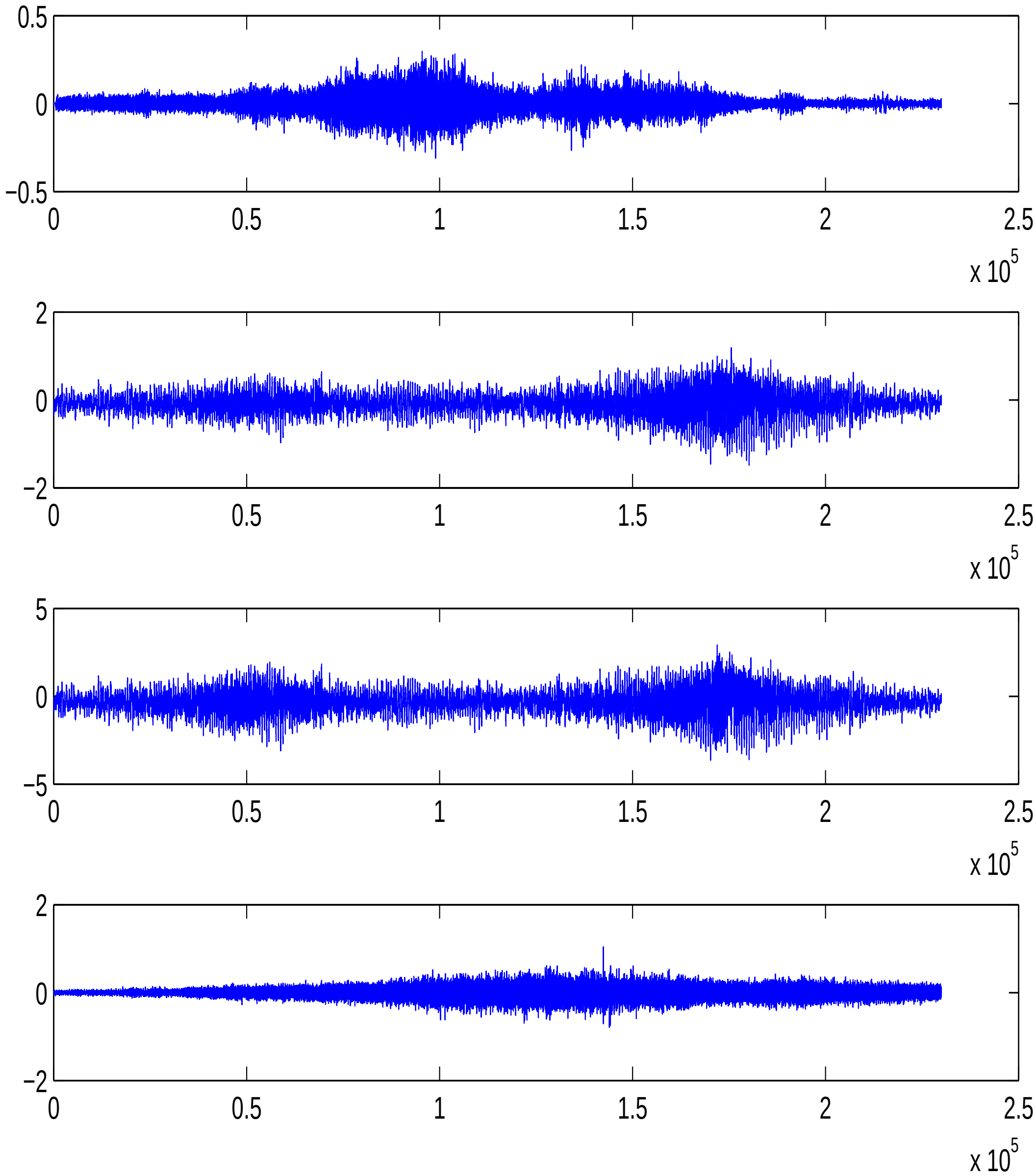


Figure 4.17: A calibrated data set example.

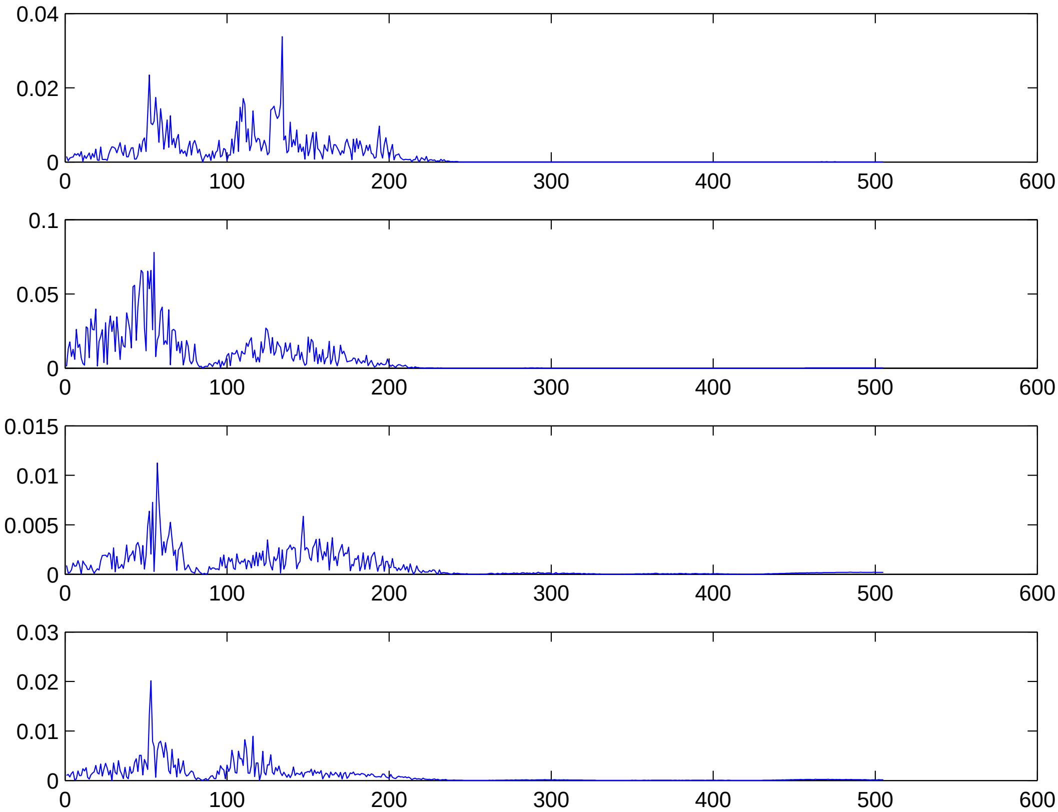


Figure 4.18: Frequency properties of 4 microphone in one frame of 0.0313 seconds.

the estimated positions and the dashed blue lines link the actual positions of a source to its time-corresponding estimated positions. It can be seen that the accuracy is low with the real world data where the RMSE of the estimated position is 10.42metres.

There are several important observations that would be useful for applying the positioning system to deal with real-world problems which makes low accuracy:

(i) Calibration data for power gain must be done because the power gain at different microphones are different, not to mention the different related hardware and softwares of the computer like sound cards and drivers.

(ii) Calibration must be considered carefully on frequency domain because each microphone has its own frequency response characteristics. That means for different frequency, the amplitude gain responses are also different. This kind of calibration is difficult to manage automatically.

(iii) Environment noise is not an ideal Gaussian noise. Noise level at high frequency is low and vice versa. That means the noise is mainly made by the moving sources themselves. It is because the sound signals emitted by the sources are at low band of frequency. As a consequence, noise removing is performed at low frequency band. If the system is designed in the way that the sensors do the DFT, then they just need to calculate the DFT for a small range of frequency based on prior statistical information. This implies that the computation load on the sensors will decrease significantly (less than 100 instead of 1000).

(iv) The dominant frequencies of a moving source is not composed of several separated f-components but usually a group of nearby f-components. Therefore, in this experiment, we do not let the system automatically decide the dominant f-components with the sliding window but let it choose the groups of nearby f-components. Moreover, the collected data is not from independent sources which violates the assumption mentioned in Chapter 3. This leads to the high error in the final result.

(v) Based on the experiment data, we also realize that same kind of moving sources can be regarded as independent sources, e.g. two motobikes at two very different speeds. It is because

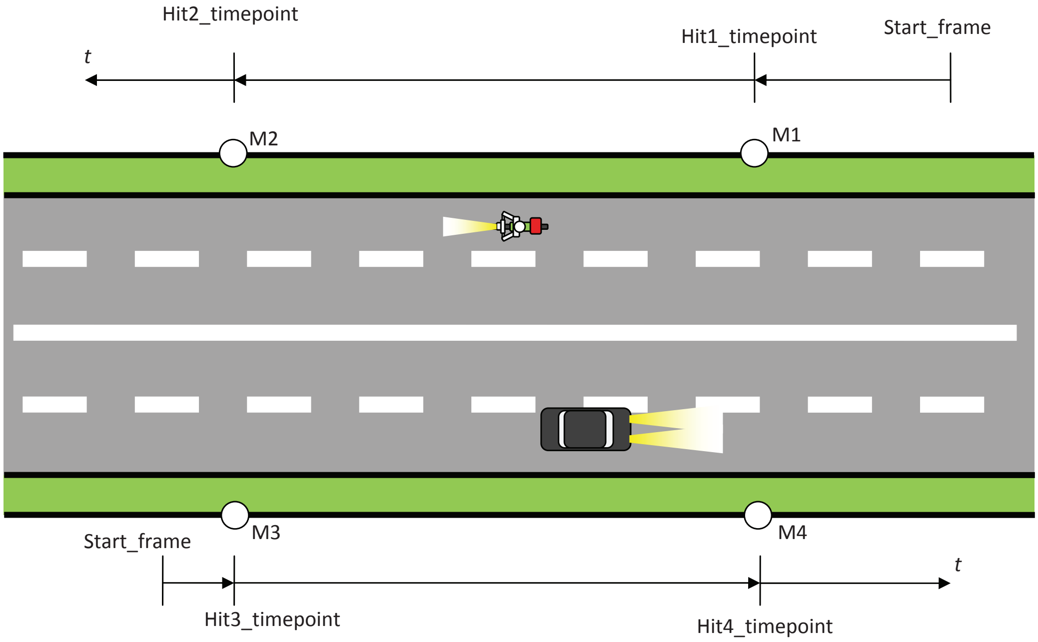


Figure 4.19: Monitoring area, microphones setting (M1, M2, M3, M4) and real position estimation with hit time points.

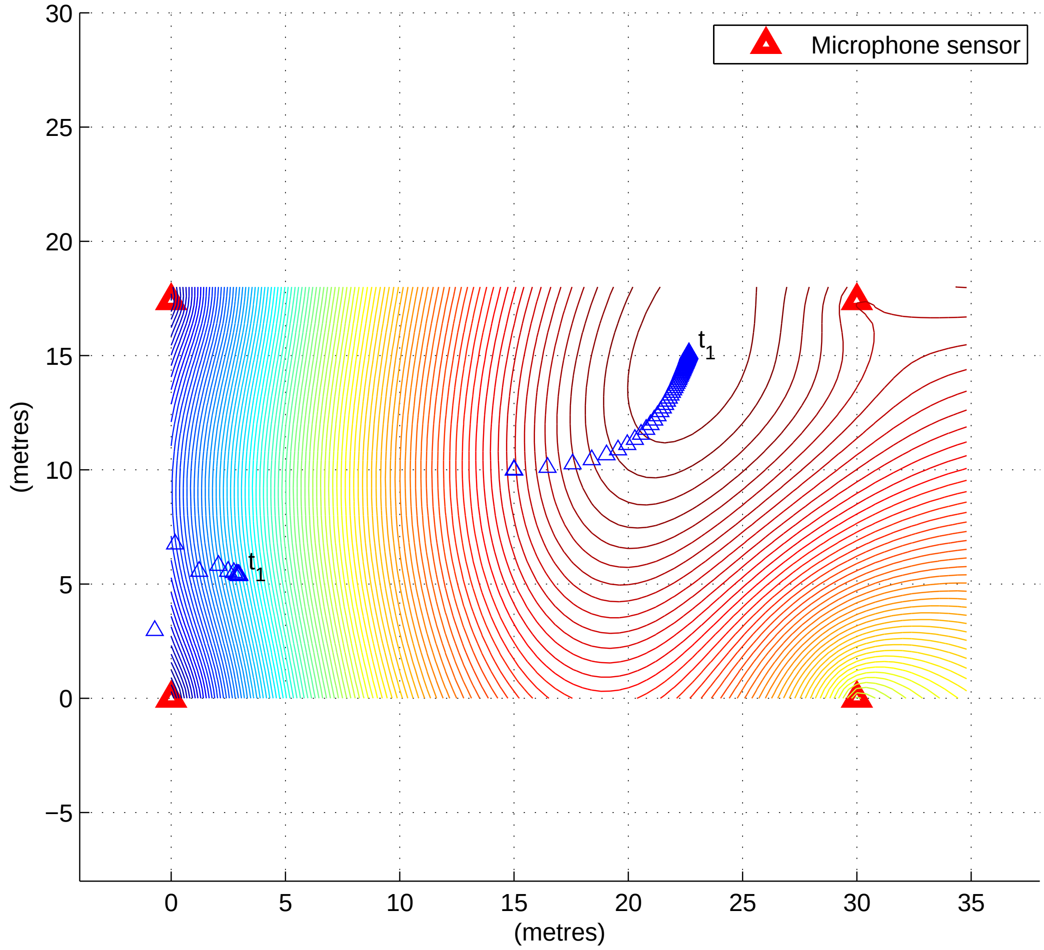


Figure 4.20: Monitoring area, microphones setting (M1, M2, M3, M4) and estimated positions of two sources at a time point. The contour is for illustrating the optimum position based on the objective function.

the motor of the motobike spinning at higher speed would emit higher frequency sound, even with the sound of the exhausted gas. Contrariwise, two different vehicles of different kinds can be high correlated, or not independent. That means the system can position several vehicles of a same kind as long as the collected data is the combinations of independent sources, of course with the time delay and Doppler effect.

(vi) For this real-world problem, we can reduce the positioning result error by forcing the position of the moving source to the nearest 'reasonable' position. In other words, if an estimated position is not in the road, then we set it to the nearest position which belongs to the roads. In this case, the mean absolute error would be reduced to 9.53 metres (RMSE decreases to 9.99 metres).

(vii) Although the accuracy is not very good, however, the tendency of moving of the sources can be still recognized. Therefore, we emphasize again that if an intelligent inference stage is used after this positioning system, then the error would be reduced considerably.

Based on the real-world data and the related result. We construct a table which is for comparison among our proposed method and some other works. In this table, the conditions, the common parameters and the accuracy are presented. This table shows that our proposed method has good performance and many advantages with both simulated data and real data. It works in a distributed manner and is able to deal with moving source with Doppler effect while the others could not. The number of detected objects can be very high if the sources are independent (5 is the number in the simulation). Meanwhile, the number of microphone sensors is low, the least necessary for the positioning task to work is 3 microphones (each sensor has 1 isotropic microphone). For both simulation and real data experiment, we used only 4 isotropic sensors. Moreover, for other methods, the sensors must be carefully deployed and the direction of the sensors must be set up and known in advance because the detected angle at a sensor is just the relative difference of the direction of arrival and the direction of the sensor. Therefore, without sensor direction, the detected angle is meaningless. The longest range of detection belongs to towed array techniques while all the rest are short, around less than 30m. It is because towed array is used under water where the signal

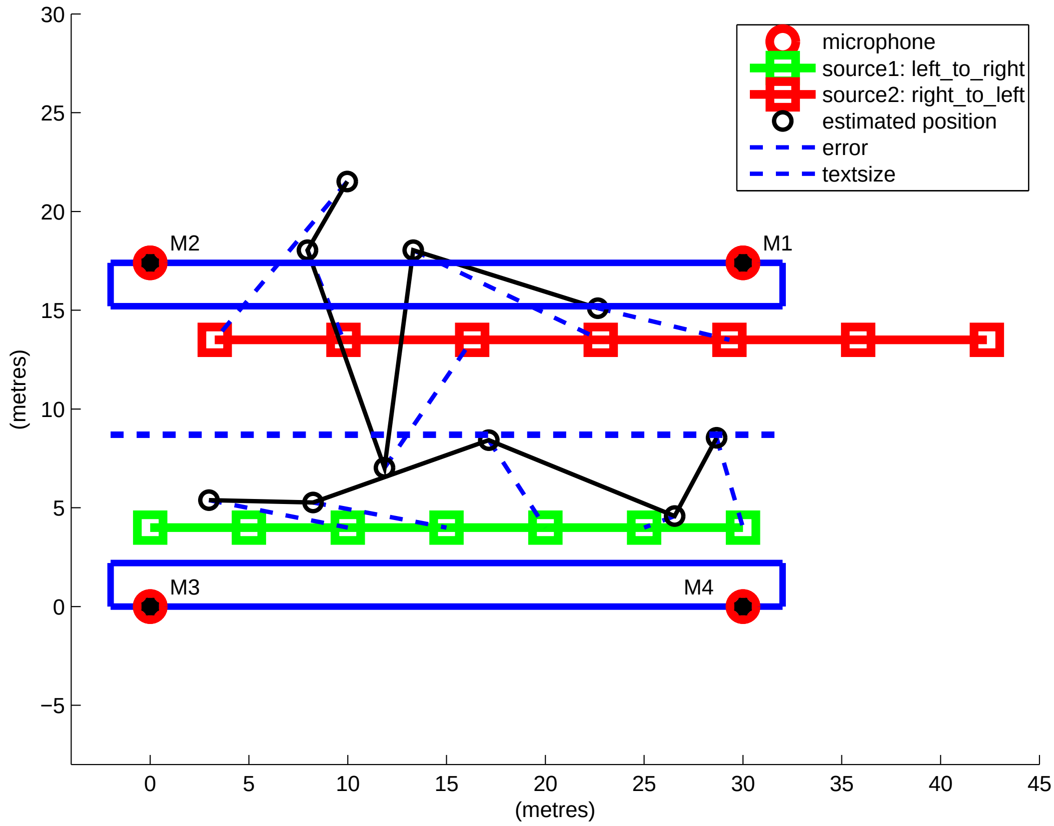


Figure 4.21: Monitoring area, microphones setting (M1, M2, M3, M4) and final result of positioning task at 5 consecutive time points. Dashed lines represent the 2-dimension errors.

Table 4.8: Summary table of comparison to other works

	Proposed method	Binaural [23][68]	Towed array [29][40][48]	Array of [105] sensor for DoA
Independent requirement	Yes	Yes	-	-
Distributed	Yes	No	No	No
Moving source /w Doppler effect	Possible	impossible	impossible[40][48] possible[29]	impossible
Environment	Air/Water	Air	Water	Air/Water
Number of objects	depend on ratio in (3.20)	2 in [23] 3 in [68]	1	1
Sensor kind	isotropic mono microphone	array of microphones	array of ($\geq$ 20)x1 microphones	array of ($\geq$ 2)x($\geq$ 2) microphones
Necessary number of microphones	≥ 3	≥ 4	≥ 20	≥ 8
Methodology	f-component energy ratio	DoA (angle detection)	DoA (angle detection)	DoA (angle detection)
Flexibility of deploying	position: Yes angle: Yes	position: No angle: No	position: No angle: No	position: No angle: No
Range of detection	Short $< 35m$	Short $< 30m$	Long $< 1km$	Short $< 30m$
Accuracy for real data (Error)	9.54m/(30mx18m) $22.8^0/360^0$ (SourceNum=2)	- -	$3 \sim 15^0/360$ [29] (dual active tow) (SourceNum=1)	$10^0/180$ (SourceNum=1)
Accuracy for simulated data (Error)	1.8m/(12mx12m) $\simeq 12.2^0/360^0$ (SNR=3.97dB) (Mic-Num=4) (SourceNum=5)	$13^0/180^0$ [68] (SNR=40dB) (Mic-Num=4) (SourceNum=3)	$23^0/180^0$ [48] (SNR=11.86dB) (Mic-Num=20) (SourceNum=1)	-

can propagate faster and less being absorbed than in the air. Moreover, the number of sensors in towed array is very large which helps the system to record data from far. In order to compare on accuracy (see the last two lines of Table.4.8), we interpret the error from meter unit to degree unit which appear all most of the other works since DoA scheme is used. The results show that for the real data, the method in [105] which tries to localize the sound source in a conference hall (one at a time) gives very good result (worst error is $10^0/360^0$) while the proposed method give the error of $22.8^0/360^0$. However, it should be noted that AMOP works with moving source where Doppler effect exists and the microphone sensors are isotropic. For simulated data, the interpreted result is $12.2^0/360^0$ while the binaural methods gives $13^0/180^0$ and towed array gives $23^0/180^0$. That means the direction of arriving angle is determined by our method while the others still have the ambiguity of arriving directions. Moreover, the SNR in our simulation is lowest when considering all worst cases of all approaches. The number of sources is also more in our simulation than in the others. Obviously, our method outperforms the others which try to locate the acoustic sources.

Chapter 5

Conclusions

Object localization is one of the most important applications of WSNs. This work thoroughly considers the problem in two sub-problems: sensor localization and object localization. Sensor localization is actually the first stage that provides prerequisite information about sensor locations since only given this information, object localization could be performed. Sensor localization problems in WSNs include numerous challenges for network designers in terms of accuracy, low-cost computation, low-cost communication, distributed ability, scalability, etc. In this thesis, we introduced a new method, the Push-pull Estimation, which has many advantages that localization solvers aim at. With the range measurements to its reference nodes and the current estimated distances, each normal node models the differences to forces and then moves under the effect of these forces. The algorithm relies mainly on geometry and is not based on available mathematical methods, such as MLE, MDS, Particle Filter, etc. As a result, we have to pay more attention to analyzing and proving most of PPE's characteristics as well as its related problems. Simulation experiments do not only confirm the above analysis, but also show that PPE is a robust distributed algorithm with very simple calculations and a low communication cost. This cost can even be reduced considerably with an insignificant accuracy trade-off by increasing the cycle request number. Furthermore, the method has a flexibility to deal with different unbiased distributions of measurement error, especially when the error depends on the actual distance or when the error has a log-normal distribution. If the measurement error is unbiased and is proportional to the actual range, PPE's result is better than that of the popular successive refinement representatives MLE and dwMDS. With the log-normal distribution of the measurement interfered by noise, PPE is far better than dwDMS in terms of computational cost and accuracy. It is also more robust than MLE, particularly if the power error has a large variance, even when MLE uses its raw estimation as the input. Moreover, PPE needs neither a complex calculation for these initial data nor final corrections like the rescaling Procrustes algorithm to improve result accuracy, thus the calculation and communication are reduced and simpler. Therefore, PPE is a very good localization candidate to WSN designers. For the purpose of object localization, this work introduces a distributed system for independent acoustic source positioning in parallelism of *data-and-process decomposition*. The information for separation is the ratios of f-component energy values received at the sensors. In order to obtain these ratios, the received signals are first segmented into frames and transformed to frequency frames with DFT transformation at the sensors. The Gaussian noise filtering and compressing also take place at the sensors before the preprocessed data is sent to the base computer. At the base, after decompressed, the data is used for f-component clustering, or determining the segments that contain the different f-components. All of the relative distance ratios are computed in order to establish an objective function for each f-component. Minimizing these objective functions reveals the location estimations of the f-components. A d-min clustering approach is used to group close f-component positions and to compute the final source locations. The simulations in this work have been made as realistic as possible with the coexistence of Doppler effect and Rayleigh multipath fading to illustrate how the system solves the problem of multiple moving source positioning. The results show that the method gives high accuracy and requires a very low communication cost for a large data set. The proposed system can be considered as the acoustic source localization design for the future generation of WSNs since it needs sensors with high computing ability in order to perform DFT on a long segment of data. However, powerful computing ability is not crucial because using the feedback from the base, the sensors only perform the full DFT once and then focus on calculating DFT at bins in several frequency segments containing major f-components. The system is actually more useful than just positioning multiple independent acoustic sources. It can also provide characteristics of the acoustic sources which can be utilized for further position estimation refinement and for recognition or classification since most acoustic features are related to the frequency domain. As the task of continuously positioning and refining the estimations of source locations requires much effort in both analysis and simulation, we let it be our next work in future.

Appendix A: Main parameters

Main parameters for Push-pull Estimation analyses

i, j, k : nodes' names, also used as the index related to the nodes.

i_0, k_0 : actual position of node i, node k.

N, n, m : total number of nodes, number of normal nodes and number of anchors respectively, $N = n + m$.

x_i : true position of node i, $x_i \in \mathbf{R}^2$.

$\tilde{x}_i$: current estimated position of node i, $\tilde{x}_i \in \mathbf{R}^2$.

d_{ij} : actual distance from x_i to x_j.

$\tilde{d}_{ij}$: current calculated distance from $\tilde{x}_i$ to $\tilde{x}_j$.

δ_{ij} : measured distance, or measurement from node i to node j which is estimated by the network and serves as the input for PPE algorithm.

p : index of PPE's phases; $p = 1, 2$ or 3.

$\overrightarrow{f_{ij}}^{(p)}$: individual force caused by j on i. $\overrightarrow{f_{ij}}^{(1)}$, $\overrightarrow{f_{ij}}^{(2)}$, and $\overrightarrow{f_{ij}}^{(3)}$ are defined in equations (2.6), (2.7), and (2.8) respectively.

$\overrightarrow{e_{ij}}$: unit vector pointing the direction from $\tilde{x}_i$ to $\tilde{x}_j$. $\overrightarrow{e_{ij}} = \frac{1}{\tilde{d}_{ij}}(\tilde{x}_j - \tilde{x}_i)$.

$\overrightarrow{F_i}^{(p)}$: the mean-force, defined in (2.4).

${M_i}^{(p)}$: number of node i's related nodes. ${M_i}^{(1)}$ is number of i's related beacons, ${M_i}^{(2)}$ is number of both i's related beacons and i's neighbors, and ${M_i}^{(3)} = {M_i}^{(2)}$.

$\alpha^{(p)}$: movement rate.

L_{ij} : variable defined in (2.10) to unify the form of $\overrightarrow{f_{ij}}^{(p)}$ for statement *iii)*'s proof.

J, K : center points of sub-areas.

(u_i, v_i), (u_j, v_j): positions of node i and node j in the coordinate system uOv.

$(\Delta F_u, \Delta F_v)$: coordinate of $\overrightarrow{\Delta F_i}$ in the coordinate system uOv.

$\overrightarrow{F_i}$: variable replacing $\overrightarrow{F_i}^{(p)}$ in statement *i)*'s proof.

P_{ij}, P_{noise} : received power (dB) and error power (dB) at i respectively when j transmits.

$g(.)$: de-biasing function.

$k^{(p)}$, $k^{(p)}_{max}$: iteration counting variable and it maximum value at phase p.

Main parameters for Multiple Moving Object Positioning analyses

Q : number of location-known sensors

$s_i(t)$: signals of source j

M : number of the sources to be located

$x_i(t)$: received signal at sensor i

$x_i(t_n)$: discrete form of $x_i(t)$ on the time domain resulted from sampling.

a_{ij} : amplitude gain of the signal from source j measured at sensor i

$\tau_{ij}(t)$: immediate propagation time of signal from source j to sensor i.

v_c, v_j : velocity of acoustic propagation and velocity of source j.

$d_{ij}(t)$: immediate distance from sensor i to source j.

f_i, f_{ij} : a specific f-component of source j and its shifted version at sensor i.

$X_i(\omega), S_i(\omega)$: STFT forms of $x_i(t)$ and $s_i(t)$.

$X_i(\omega_k), S_i(\omega_k)$: discrete Fourier transformation (DFT) forms of $x_i(t_n)$ and $s_i(t_n)$.

r_{ilz} : distance constrain, the ratio of distance from source z to sensor l and from source z to sensor i.

$X_i(\omega_k^{(m)})$: a segment of $X_i(\omega)$, particularly in (ω_a, ω_b) on Fourier domain.

$\mathbf{F}_j$: objective function that needs to be minimized so that the position of a dominant f-component belonging to source j can be obtained.

$\mathbf{p}^{(m)}$: position of a dominant f-component, $\mathbf{p}^{(m)} \in \mathbf{R}^2$

$w^{(m)}$: the weight represented by the mean value of the f-component's energies at the sensors.

Appendix B: Conjugate Gradient Optimization Method

Conjugate Gradient Optimization is used to find the optimal points of a function in stead of the simple negative gradient descent method. Negative gradient descent is one of first-order optimization approaches. It makes a jump in the searching space along the direction that gives the steepest change of the function. Meanwhile, conjugate gradient optimization is a second-order optimization. It regards the surface of the function around the current considered point to be quadratic and make the jumps to the optimal point. The speed of convergence is faster than the negative conjugate and as the trade-off, the second order of derivative must be computed.

Assume that we want to find the solution of minimizing a function f. The local area of the considering point x has the approximation of a quadratic function (just like Quasi-Newton method):

$$\frac{1}{2}x^T Hx + b^T x \tag{B.1}$$

where H is the Hessen matrix representing the differential relations, H is symmetric positive definite. The unique solution of this minimization is the unique solution of

$$Hx + b = 0 \tag{B.2}$$

Then the Conjugate Gradient Algorithm is in details as following [91]:

1. $g_0 \leftarrow Hx_0 + b$
2. $d_0 \leftarrow -g_0$
3. for $k = 0, ..., n-1$ do

   $\alpha_k \leftarrow -\frac{g_k^T d_k}{d_k^T H d_k}$

   $x_{k+1} \leftarrow x_k + \alpha_k d_k$

   $g_{k+1} \leftarrow Hx_{k+1} + b$

   $\beta_k \leftarrow \frac{g_{k+1}^T H d_k}{d_k^T H d_k}$

   $d_{k+1} \leftarrow -g_{k+1} + \beta_k d_k$
4. Return x_n

Algorithm 6: Conjugate gradient algorithm

In order to overcome the sophistication of calculating the second-order derivative of function $f(x)$, another method extended from conjugate gradient has been introduced by Fletcher and Reeves. This approach only need the first-order derivative combining with any simple line search method. Its details are presented below [91]:

1. Start at some x_0
2. $d_0 \leftarrow -\nabla f(x_0)$
3. for $k = 0, ..., n-1$ do

   $\alpha_k = \arg\min_{\alpha_k} g(\alpha) = \arg\min_{\alpha_k} f(x_k + \alpha d_k)$
   $x_{k+1} \leftarrow x_k + \alpha_k d_k$
   $\beta_k \leftarrow \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$
   $d_{k+1} \leftarrow -\nabla f(x_{k+1}) + \beta_k d_k$

4. $x_0 \leftarrow x_n$
5. go back to step 2 until satisfied with the results.

Algorithm 7: Fletcher-Reeves

When considering the convergent speed, the number of loops for calculating x is equal to second-order Newton optimization. However, it does not require the Hessen H. Therefore, this is regarded to be an intermediate method between negative gradient descent and Newton's method.

References

[1] P. Aarabi and S. Mavandadi. Multi-source time delays of arrival estimation using conditional time-frequency histograms. *Information Fusion*, 4:111–122, 2003.

[2] P. S. Alessio Brutti, Maurizio Omologo. Comparison between different sound source localization techniques based on a real data collection, 2008.

[3] F. Antonacci, D. Riva, D. Saiu, A. Sarti, M. Tagliasacchi, and S. Tubaro. Tracking multiple acoustic sources using particle filtering. In *European Signal Processing Conference, EUSIPCO-2006*, 2006.

[4] H. R. Arabnia. A parallel algorithm for the arbitrary rotation of digitized images using process-and-data-decomposition approach. *Journal of Parallel and Distributed Computing*, 10:188–192, 1990.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. 50:174188, 2008.

[6] D. Barry, D. Fitzgerald, E. Coyle, and B. Lawlor. Single channel source separation using short-time independent component analysis. In *119th Convention, Audio Engineering Society*, Oct. 2005.

[7] J. Benesty. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. *Journal of the Acoustical Society of America*, 107:384–391, 2000.

[8] S. Biaz and Y. Ji. Precise distributed localization algorithms forwireless networks. In *WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)*, 2005.

[9] M. S. Brandstein, J. E. Adcock, and H. F. Silverman. A closed-form location estimator for use with room environment microphone arrays. *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, 5:45–50, 1997.

[10] A. Brutti, M. Omologo, and P. Svaizer. Multiple source localization based on acoustic map de-emphas. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010:82–90, 2010.

[11] G. Brys, M. Hubert, and P. Rousseeuw. A robustification of independent component analysis, 2005.

[12] V. Cevher, A. C. Sankaranarayanan, and R. Chellappa. Factorized variational approximations for acoustic multi source localization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*, pages 2409–2412. IEEE, 2008.

[13] J. Chen, K. Yao, and R. E. Hudson. Source localization and beamforming, 2002.

[14] A. Ciaramella and R. Tagliaferri. Separation of convolved mixtures in frequency domain ica. *International Mathematical Forum*, 16:769–795, 2006.

[15] J. Costa, N. Patwari, and A. H. III. Distributed weighted multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks*, 2:16–23, 2008.

[16] V.-H. Dang. Generated examples of multi-object sound data. In *http://uclab.khu.ac.kr/ext/dvhung/*.

[17] V.-H. Dang, V.-D. Le, Y.-K. Lee, and S. Lee. Distributed pushpull estimation for node localization in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 71:471–484, 2011.

[18] V.-H. Dang, T. Le-Tien, Y.-K. Lee, and S. Lee. Acoustic multiple object positioning system. In *International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN-2010*. ACM, 2010.

[19] V.-H. Dang, S. Lee, and Y.-K. Lee. A distributed design for multiple moving source positioning. *Journal of Supercomputing*, Accepted in 2011.

[20] E. R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices,. *Journal of Computing and Physics*, 17:87–94, 1975.

[21] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Son, 2nd edition, 2001.

[22] A. V. D. Enden and N. Verhoeckx. *Discrete-time Signal Processing An Introduction*. Prentice Hall, first edition, 1989.

[23] C. Faller and J. Merimaa. Source localization in complex listening situations: Selection of binaural cues based on interaural coherence. *Journal of the Acoustical Society of America*, 116:3075–3089, 2004.

[24] R. Fan, H. Jang, S. Wu, and Z. Zhang. Ranging error tolerable localization in wsn with inaccurately positioned anchor nodes. *Wireless communications and Mobile Computing*, 9:705–717, 2009.

[25] A. G. C. Fotios Talantzis and L. C. Polymenakos. Estimation of direction of arrival using information theory. *IEEE SIGNAL PROCESSING LETTERS*, 12:561–564, 2005.

[26] R. H. Gilkey and E. T. R. Anderson. *Binaural and Spatial Hearing in Real and Virtual Environments*. Mahwah, NJ: Lawrence Erlbaum Associates, 1997.

[27] A. Gisolf and D. Simons. *Adaptive Motion Compensation in Sonar Array Processing*. Netherlands organization for applied scientific research (TNO), 1st edition, 2006.

[28] B. Gold and N. Morgan. *Speech and Audio Signal Processing*. Wiley, 2000.

[29] J. GROEN. *Adaptive motion compensation in sonar array processing*. Netherlands organization for applied scientific research (TNO), first edition, 2006.

[30] I. Guvenc, C. T. Abdallah, R. Jordan, and O. Dedeoglu. Enhancements to rss based indoor tracking systems using kalman filters. In *Global Signal Processing Expo and International Signal Processing Conference*, 2003.

[31] R. A. Haddad and T. W. Parsons. *Digital Signal Processing Theory, Applications and Hardware*. Computer Science Press, 1991.

[32] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom'03)*, Sep. 2003.

[33] Y. Huang and J. Benesty. A class of frequency-domain adaptive approaches to blind multichannel identification. *IEEE Transaction on Signal Processing*, 51:11–24, 2003.

[34] A. Hyvarinen and E. Oja. Independent component analysis: a tutorial. *Neural Networks*, 13:4–5, 2000.

[35] A. Hyvarinen, P. Ramkumar, L. Parkkonen, and R. Hari. Independent component analysis of short-time fourier transforms for spontaneous eeg/meg analysis. *NeuroImage*, 49:257–271, 2010.

[36] J. O. S. III and N. Lee. Pitch glide analysis and synthesis from recorded tones. In *CCRMA*, pages 1–8, 2009.

[37] E. S. M. Jesus, O. C. Rodrguez, A. Alcocer, P. Oliveira, and A. Pascoal. Underwater acoustic positioning systems based on buoys with gps. In *Underwater Acoustics, 8th ECUA*, 2006.

[38] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *IEEE INFORCOM*, 2004.

[39] D. J.Young and N. C. Beaulieu. The generation of correlated rayleigh random viriates by inverses discrete fourier transform. *IEEE Transactions on Communications*, 48:1114–1127, 2000.

[40] K. Kaouri. Left-right ambiguity resolution of a towed array sonar. In *Ph.D dissertation*. MSc in Mathematical Modelling and Scientific Computing, 2000.

[41] Y. H. Kim and A. Ortega. Quantizer design for source localization in sensor networks. *IEEE Transactions on Signal Processing*, 59:5577 – 5588, 2011.

[42] R. H. Lambert. Multichannel blind devonvolution: Fir matrix algebra and separation of multipath mixtures. In *Ph.D dissertation*. University of Southern California, EE dept., 1996.

[43] G. Latsoudas and N. Sidiropoulos. Ranging error tolerable localization in wsn with inaccurately positioned anchor nodes. *IEEE Transactions on Signal Processing*, 55:5121–5127, 2007.

[44] G. Latsoudas and N. D. Sidiropoulos. A fast and effective multidimensional scaling approach for node localization in wireless sensor networks. *IEEE Transactions on Signal Processing*, 55:5121–5127, 2007.

[45] V.-D. Le, V.-H. Dang, S. Lee, and S.-H. Lee. Distributed localization in wireless sensor networks based on force-vectors. In *ISSNIP 2008: Intelligent Sensors, Sensor Networks and Information Processing.*, Dec 2008.

[46] D. Li and Y. H. Hu. Energy-based collaborative source localization using acoustic microsensor array. *EURASIP Journal on Applied Signal Processing*, page 321337, 2003.

[47] X. Li. Collaborative localization with received-signal strength in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 56:3018–3022, 2007.

[48] F. Lu, E. Milios, S. Stergiopoulos, and A. Dhanantwari. New towed-array shape-estimation scheme for real-time sonar systems. 28:552–563, 2003.

[49] A. Mahajana and M. Walworth. 3-d position sensing using the differences in the time-of-flights from a wave source to various receivers. *IEEE Transaction on Robotics and Automation*, 17:91–94, 2001.

[50] R. A. Malaney. Nuisance parameters and location accuracy in log-normal fading models. *IEEE Transactions on Wireless Communications*, 6:937–947, Mar. 2007.

[51] J. F. Michel and M. Vossiek. Wireless sensor network approach for robust localization of mobile nodes with minimal complexity. *e & i Elektrotechnik und Informationstechnik*, 125:341–346, Oct. 2008.

[52] N. Mitianoudis and M. E. Davies. Audio source separation of convolutive mixtures. *IEEE Transaction on Speech and Audo Processing*, 11:489–497, 2003.

[53] C. Morelli, M. Nicoli, V. Rampa, and U. Spagnolini. Hidden markov models for radio localization in mixed los/nlos conditions. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 55:1525–1542, 2007.

[54] O. L. Moses, D. Krishnamurthy, and R. Patterson. A self-localization method for wireless sensor networks. *EURASIP Journal on Applied Signal Processing*, 4:348–358, 2003.

[55] R. Mutihac and M. M. V. Hulle. Comparison of principal component analysis and independent component analysis for blind source separation. *IEEE Transactions on Audio, Speech, and Processing*, 56:20–32, 2004.

[56] M.Wu, D. L.Wang, and G. J. Brown. A multipitch tracking algorithm for noisy speech. *IEEE Transaction on Speech and Audio Processing*, 11:229–241, 2003.

[57] S. Namik, U. J. Ferner, and K. W. Sowerby. Localization in harsh propagation environments. In *Communications Theory Workshop, 2008. AusCTW 2008. Australian*, pages 161–166, Feb. 2008.

[58] D. Niculescu and B. Nath. Dv based positioning in ad hoc networks. *Journal of Telecommunication Systems*, 22:267–280, 2003.

[59] M. Omologo and P. Svaizer. Acoustic event localization using a crosspower-spectrum phase based technique. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 273–276, 1994.

[60] N. Patwari. Wireless sensor network localization measurement repository. In *http://www.eecs.umich.edu/hero/localize/*.

[61] N. Patwari, J. Ash, S. Kyperountas, A. H. III, R. Moses, and N. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22:54–69, 2005.

[62] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. Odea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51:2137–2148, 2003.

[63] N. Patwari and A. H. III. Adaptive neighborhoods for manifold learningbased sensor localization. In *2005 IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, pages 1098–1102, 2005.

[64] Y. Peled and B. Rafaely. Study of speech intelligibility in noisy enclosures using optimal spherical beamforming. In *IEEE 25th Convention of Electrical and Electronics Engineers*. IEEE, 2008.

[65] H. Qasem and L. Reindl. Precise wireless indoor localization with trilateration based on microwave backscatter. In *Wireless and Microwave Technology Conference, 2006. WAMICON '06. IEEE Annual*, pages 265–270, Dec. 2006.

[66] T. F. Quatieri. *Discrete-Time Speech Signal Processing: Principles and Practice*. PHI, 2002.

[67] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall Inc., 1996.

[68] N. Roman and D. Wang. Binaural tracking of multiple moving sources. *IEEE Transactions on Audio, Speech, and Processing*, 16:728–139, 2008.

[69] Y. Sasaki, S. Kagami, and H. Mizoguchi. Multiple sound source mapping for a mobile robot by self-motion triangulation. In *International Conference on Intelligent Robots and Systems*, pages 380–385. IEEE, Oct 2006.

[70] A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 112–121. ACM Press, 2002.

[71] H. Sawada, R. Mukai, S. Araki, and S. Malcino. Maximum likelihood sound source localization for multiple directional microphones. In *Antennas and Propagation Society International Symposium*, 2005.

[72] R. Serway and R. Beichner. *Physics for Scientists and Engineers*. Brooks/Cole Publishing Company, fifth edition, 1999.

[73] Y. Shang, W. Ruml, Y. Zhang, and M. P. J.Fromherz. Localization from mere connectivity. In *Mobihoc 03*, pages 201–212, 2003.

[74] Y. Shang, W. Ruml, Y. Zhang, and M. P. J.Fromherz. Localization from connectivity in sensor networks. *IEEE Transaction of Parallel Distributed System*, 15:961–974, 2004.

[75] X. Sheng and Y.-H. Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transaction on Signal Processing*, 53:44–53, 2005.

[76] J.-P. Sheu, P.-C. Chen, and C.-S. Hsu. A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement. *IEEE Transactions on Mobile Computing*, 7:1110–1123, 2008.

[77] J. Shlens. Notes on independent component analysis, 2002.

[78] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *Network, IEEE*, 18:45–50, Aug. 2004.

[79] P. Smaragdis. Blind separation of convolved mixtures in the frequency domain. *Neurocomputing*, pages 22:21–34, 1998.

[80] M. Stanacevic. Micropower gradient flow acoustic localizer. *IEEE Transations on Circuits and systems*, 52:2148–2157, 2005.

[81] L. D. Stone. A bayesian approach to multiple-target tracking. In *Handbook of Multisensor Fusion*, 2001.

[82] B. Supper, T. Brookes, and F. Rumsey. An auditory onset detection algorithm for improved automatic source localization. *IEEE Transaction Speech Audio Process*, pages 1008–1017, 2006.

[83] P. Svaizer, A. Brutti, and M. Omologo. Use of reflectedwavefronts for acoustic source localization with a line array. In *in Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pages 165 – 169, 2011.

[84] J. Tabrikian, J. Krolik, and H. Messer. Robust maximum likelihood source localization in an uncertain shallow water waveguide. *Journal of the Acoustical Society of America*, 101:241–249, 1997.

[85] C. Takenga, T. Peng, and K. Kyamakya. Post-processing of fingerprint localization using kalman filter and map-matching techniques. In *ICACT2007*, 2007.

[86] F. Talantzis. An acoustic source localization and tracking framework using particle filtering and information theory. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 18:1806–1817, 2010.

[87] P. Tarrio, A. M. Bernardos, and J. R. Casar. An rss localization method based on parametric channel models. In *International Conference on Sensor Technologies and Applications*, pages 265–270, Oct. 2007.

[88] S. T.Birchfield and D. K. Gilmor. A unifying framework for acoustic localization. In *IEEE International Conference*. IEEE Computer Society, 2001.

[89] S. T.Birchfield and D. K. Gilmor. Fast bayesian acoustic localization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2002.

[90] S. Tian, S. Zhang, X. Wang, P. Sun, and H. Zhang. A selective anchor node localization algorithm for wireless sensor networks. In *ICCIT'07: International Conference on Convergence Information Technology*, pages 358–362. IEEE, 2007.

[91] J. V. Trier. Some nonlinear optimization methods. In *http://sepwww.stanford.edu/public/docs/sep51/*.

[92] J.-M. Valin, F. Michaud, J. Rouat, and D. Letourneau. Robust sound source localization using a microphone array on a mobile robot. In *Proc. ICASSP98*, pages 1228–1233, 2003.

[93] B.-N. Vo, S. Singh, and A. Doucet. Sequential monte carlo methods for bayesian multi-target filtering with random finite sets. 41:1224–1245, 2005.

[94] C. Wang and L. Xiao. Sensor localization under limited measurement capabilities. *IEEE Network*, pages 16–23, 2008.

[95] D. B. Ward, E. A. Lehmann, and R. C. Williamson. Particle filtering algorithms for tracking an acoustic source in a reverberant environment. 11:826–836, 2003.

[96] D. B. Ward and R. C. Williamson. Particle filter beamforming for acoustic source localization in a reverberant environment. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2002.

[97] Website. Futurlec. In *http://www.futurlec.com/GPS.shtml*.

[98] Website. Gps tracklog. In *http://gpstracklog.com/buyers-guides/handheld-gps-buyers-guide*.

[99] H.-W. Wei, Q. Wan, Z.-X. Chen, and S.-F. Ye. A novel weighted multidimensional scaling analysis for time-of-arrival-based mobile location. *IEEE Transactions on Signal Processing*, 56:3807–3817, 2008.

[100] D. A. Wismer and R. Chattergy. *Introduction to Nonlinear Optimization*. North-Holland, 1979.

[101] X. Xie and R. J. Evans. Multiple target tracking and multiple frequency line tracking using hidden markov models. 39:2659–2676, 1991.

[102] J. Yang and Y. Chen. A theoretical analysis of wireless localization using rf-based fingerprint matching. In *Proceedings of the 22nd IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, Apr. 2008.

[103] J. Yim, S. Jeong, J. Joo, and C. Park. Utilizing map information for wlanbased kalman filter indoor tracking. In *Future Generation Communication and Networking Symposia*. FGCNS'08, 2008.

[104] S. Zainalie. A clustering algorithm for localization in wireless sensor networks. In *2008 Internatioal Symposium on Telecommunications*, pages 435–439, 2008.

[105] C. Zhang, D. Florncio, S. Member, D. E. Ba, and Z. Zhang. Maximum likelihood sound source localization for multiple directional microphones. In *In ICASSP*, 2007.