# 國立交通大學

## 電信工程學系

## 碩 士 論 文

# 無線感測網路之合作式定位研究

# Cooperative Localization in Wireless Sensor Networks

研究生：王瑋萱

指導教授：謝世福　教授

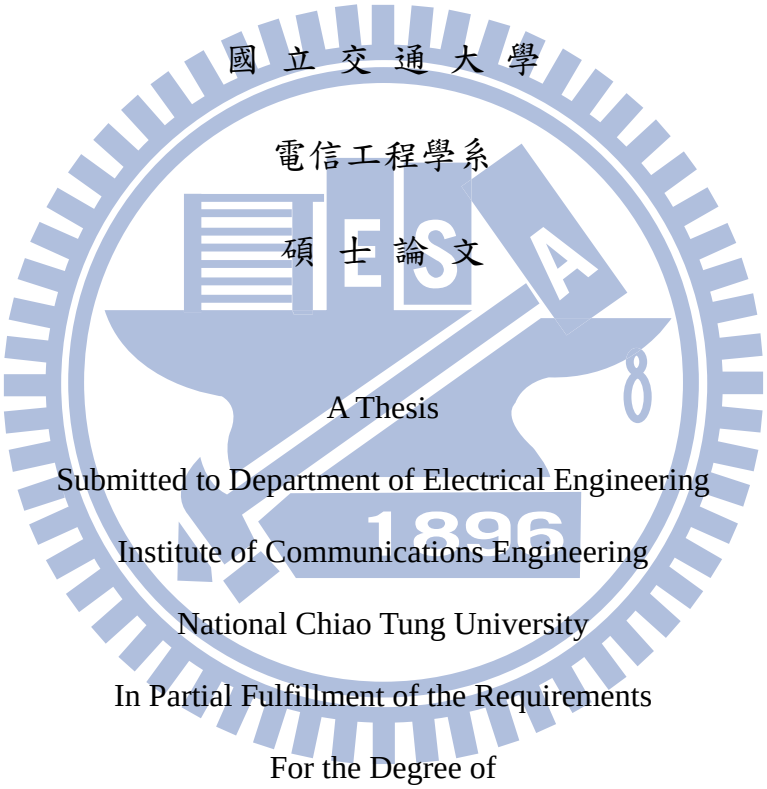
中 華 民 國 九十九　年　十一　月

# 無線感測網路之合作式定位研究

# Cooperative Localization in Wireless Sensor Networks

研 究 生：王瑋萱　　Student：W.S. Wang

指導教授：謝世福　　Advisor：S. F. Hsieh



國 立 交 通 大 學

電信工程學系

碩 士 論 文

A Thesis

Submitted to Department of Electrical Engineering

Institute of Communications Engineering

National Chiao Tung University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

In

Communication Engineering

Hsinchu 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年十一月

# 無線感測網路之合作式定位研究

學生:王瑋萱　　指導教授:謝世福

國立交通大學電信工程研究所

## 中文摘要

定位的應用非常廣泛，如:追蹤目標、航海定位和救援…等。在定位系統中，藉由感測器對待測目標的量測資料(角度或距離)估計待測目標的位置。當網路中多個待測目標可相互通訊時，合作式定位被提出:藉由待測目標之間(合作)的量測資料可以有效的提升定位精準度。在合作式定位中，最佳的演算法為最大概似法，其方程式為非線性方程式，可使用聯合牛頓法來實現。但是當待測目標數目增加時，運算量會變的相當大。因此我們提出兩種降低運算量的方式：第一種方式是藉由過去非合作線性化的經驗，我們成功推導出聯合泰勒展開演算法。第二種方法，我們提出分割疊代演算法來降低運算複雜度，疊代可提升定位準確性。也將原本的非線性方程式做雙曲線、新增變數和泰勒展開三種線性化處理。在效能評估中，我們討論三種非合作線性演算法的定位效能，進一步在分割演算法裡，針對泰勒展開線性化方法，設定在兩個待測目標情況下，推導其收斂的 MSE 理論值。另外，合作式的 *C*ramer-*R*ao *L*ower *B*ound (CRLB) 藉由 *F*isher *I*nformation *M*atrix (FIM) 的反矩陣求得，然而矩陣的大小隨著待測目標增加而變大，導致求反矩陣的運算非常的複雜。因此我們提出遞迴方塊反矩陣的特性，推導一個運算簡單的近似等 FIM，並且利用近似結果，進一步去推導近似的合作 CRLB。最後，利用電腦模擬驗證推導的結果。

# Cooperative Localization in Wireless Sensor Networks

**Student: W. S. Wang** **Advisor：S. F. Hsieh**

**Department of Communication Engineering**

**National Chiao Tung University**

## Abstract

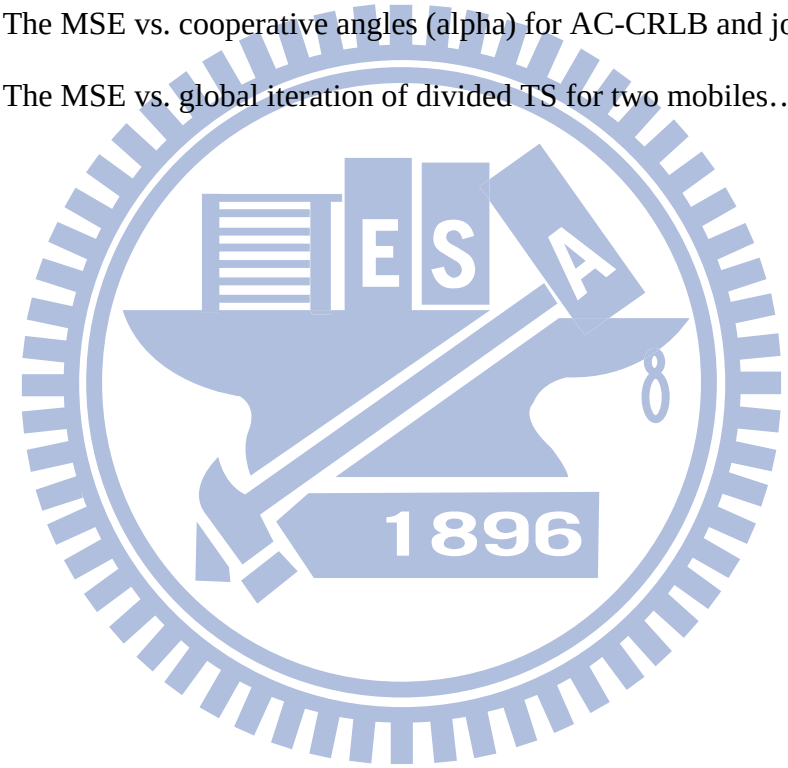
There are a lot of applications of localization including tracking, search, navigation and rescue. We can estimate position of mobile (object) by measurements (angles or distances) in sensor network. When mobiles can communicate to each other, cooperative localization has been proposed to improve the localization accuracy. In cooperative localization, the optimal ML estimator is nonlinear. It can be solved by Newton's method, but the cost of computation increases when the number of mobiles increases. Therefore, we propose two methods to reduce the computation cost, joint Taylor-series expansion algorithm and divide-and-conquer algorithm. In divide-and-conquer algorithm, we use recursive method to enhance localization accuracy and simplify the nonlinear function by three linearization methods. Next, we compare the MSE performance of three linearized algorithms and derive the theoretical converged mean-square-error for divided Taylor-series expansion algorithm. Besides, cooperative *C*ramer-*R*ao *L*ower *B*ound (CRLB) is derived by *F*isher *I*nformation *M*atrix (FIM) inverse, but the size of matrix increases when the number of mobiles increases. Then, we propose recursive block matrix inversion to derive a simple *A*pproximated *E*quivalent FIM (AEFIM) and we further utilize the result to derive the *A*pproximate *C*ooperative CRLB (AC-CRLB). Simulations are performed to support the theoretical results.

# Acknowledgment

在碩士班的兩年期間學習做研究，起初對我而言既陌生而且艱難。但是在指導教授 謝世福老師耐心的指引與教導，讓我在面對問題的時，能以正確的邏輯觀念與嚴密的思考模式，嚴謹的推敲出解決方法。學習的過程雖然辛苦，但是我很感激老師的栽培；也很感謝實驗室的學長、學姐、同學的幫忙與鼓勵。我很珍惜並感謝在這些日子裡接受到的磨練，讓我能勇敢並樂觀的接受未來的種種挑戰。

# Contents

# List of Figures



# Chapter 1
# Introduction

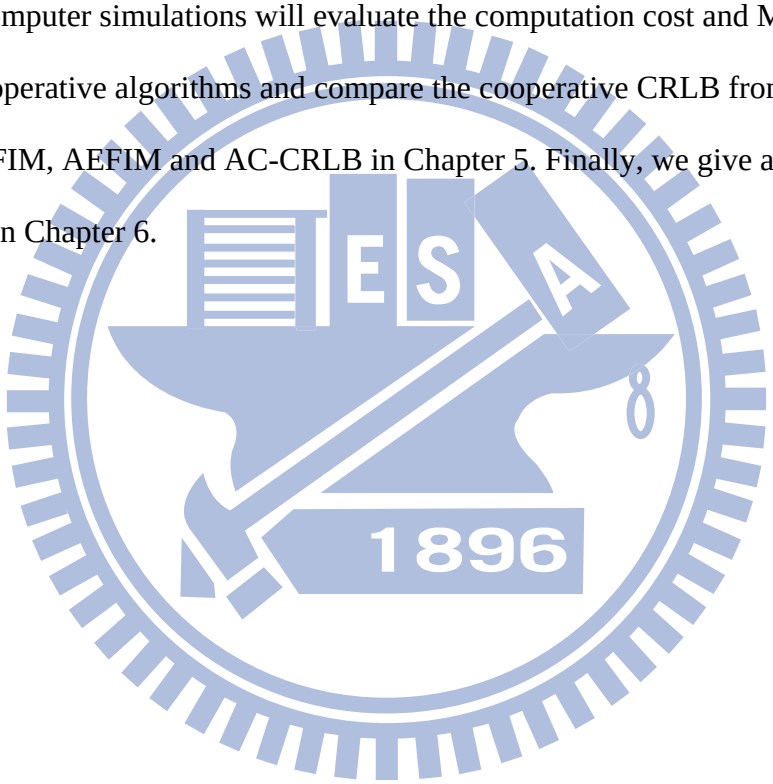
In recent years there has been interest in wireless sensor networks for variety of applications [2, 3 ,4]. Among those are health, commercial, environmental, public safety, home applications. In literature there are many methods to provide the localization estimation for wireless sensor network. Classify these localization methods as the deterministic [1, 12, 27] and probabilistic approached [22, 23]. Typical positioning parameters include time-of-arrival (TOA) [29, 45], time-difference-of arrival (TDOA) [5, 6], angle-of-arrival (AOA) [7] and received signal strength (RSS) [8, 9], and hybrid TDOA/AOA of mixture method [10, 11]. In this thesis, we only consider the TOA localization algorithms. However, in cooperative localization system, cooperative connection or Ad Hoc short-range communication among the terminals will be supported [41, 42]. Because they consider small-scale information between unknown positions can improve the localization accuracy. Therefore they focus on the data fusion of large-scale and small-scale. Cooperative Localization with Optimum Quality of Estimate (CLOQ) is proposed in [26] which takes advantage of the behavior of the channel to provide accurate indoor positioning. But, the estimated positions are fixed when they are estimated. Therefore, [15] devised an error propagation aware algorithm to update the unknown positions. Note that cooperative localization is not a well solved problem because the distance measurements between any pairs of unknown positions are utilized to aid in the location estimation. Then, [12] devised three novel subspace methods to solve that problem. Assuming the range measurements error are Gaussian distributed, the ML estimator [42] is another localization method and it is a nonlinear least squares problem [30]. It can be solved

by joint Newton's algorithm, but the computation cost becomes more complex when the numbers of unknown positions increases. The algorithm of [15] tracks the extent of the uncertain virtual position error, but this algorithm has trade-off between computation cost and localization accuracy. Then, we propose two methods to reduce the computation cost, joint Taylor-series expansion (TS) algorithm and divided-and-conquer method. However divided Newton's algorithm is a nonlinear function. Based on previous research of uncooperative linearized algorithms, we further use three algorithms, Taylor-series expansion (TS) algorithm [35] , distance-augmented (DA) algorithm [46] and hyperbolic-canceled (HC) algorithm [22] to perform divided-and-conquer method. One of the most important problems is the source of errors, including non-line-of-sight (NLOS) [16, 17] and multipath propagation [29, 37]. Then, tracking [20, 21] a moving unknown position is another important issue in sensor network localization. While the mobile is moving, the main concern is to estimate its trajectory. Kalman filter[18, 19] has been widely applied in trajectory estimation of a moving object.

However, we know that the variance of the estimate is bounded by the *C*ramer-*R*ao *L*ower *B*ound (CRLB) [25]. It reveals the full *F*isher *I*nformation *M*atrix (FIM). In cooperative localization, the full cooperative FIM is too complex to see the benefits of cooperation. Therefore, [31] proposed an eigenvalue view of *E*quivalent FIM (EFIM) to provide some insight in cooperative localization information for only two unknown positions. If there are more than two unknown positions, it still can not see the effect of cooperative. Therefore, we propose a recursive block matrix inversion based on eigenvalue view to derive the *A*pproximation of EFIM (AEFIM). We can see that the more cooperative positions, the better localization accuracy. Then, we further utilize the result to derive the Approximation of Cooperative CRLB (AC-CRLB) and we find that some parameters, variance of measurement error,

numbers of known positions, numbers of unknown positions and cooperative angles can influence the localization accuracy.

This thesis is organized as follows. We introduce basic localization system in Chapter 2 and proposed two low-cost cooperative localization methods, joint Taylor-series expansion algorithm (joint TS) and divided-and-conquer method (divided algorithms) based on TOA in Chapter 3. In Chapter 4, we derive AC-CRLB and theoretical converged Mean-Square-Error (MSE) of divided TS for two unknown positions. Computer simulations will evaluate the computation cost and MSE for proposed cooperative algorithms and compare the cooperative CRLB from full cooperative FIM, AEFIM and AC-CRLB in Chapter 5. Finally, we give a conclusion of our work in Chapter 6.



# Chapter 2
# Basic Localization System

Figure 1 shows a basic localization system. There are *N* known positions of sensors and *M* unknown positions of mobiles. Mobiles transmit information to all sensors by wireless networks. $(x_i, y_i)$ and $(\bar{x}_j, \bar{y}_j)$ are coordinates of mobile *i* and sensor *j*, respectively.

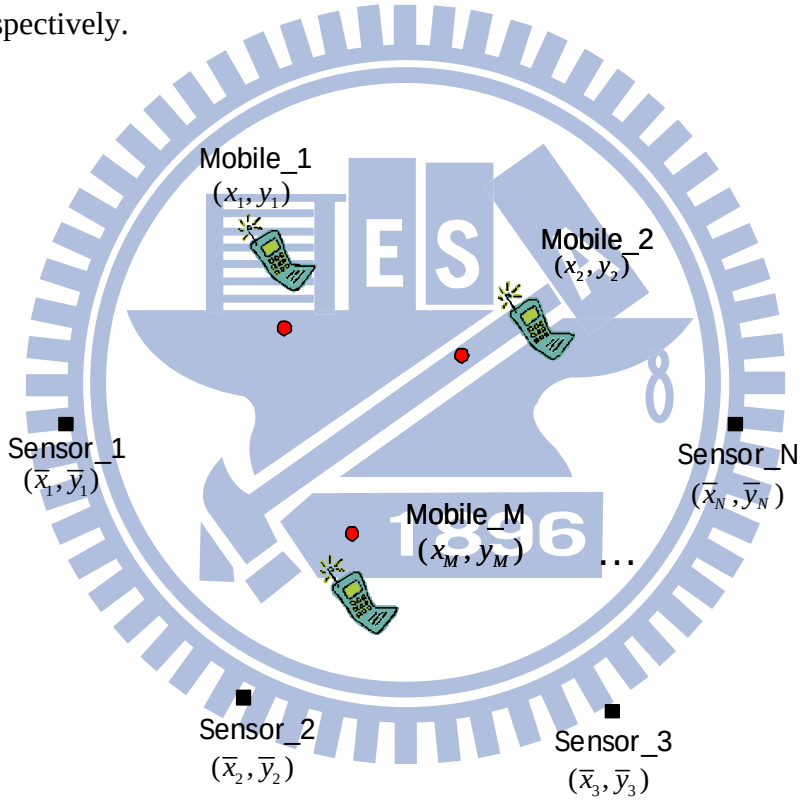



Figure 2.1 A basic localization system in wireless sensor network.

In order to estimate positions of mobiles, we have to use the measurements between the sensors and mobiles. However, the positioning accuracy is degraded in the NLOS environment when large NLOS error is imposed on the TOA, TDOA, RSS or AOA measurement. There are some algorithms to estimate the positions of mobiles

based on these measurement. Mathematically, assuming that the range of measurement errors are Gaussian distributed, the maximum likelihood (ML) methods for localization correspond to the nonlinear least squares problem [30], but the ML approach cannot guarantee global convergence. In order to ensure a global solution, semi-definite programming (SDP) [1, 32] and classical multidimensional scaling (MDS) [33, 34] have been proposed. These unknown positions of mobiles will be estimated by four common measurements. Then, these measurements are introduced in Section 2.1. Then, the basic localization algorithm based on TOA measurement is discussed in Section 2.2.

## 2.1 Measurement Characterization

In a localization system, we utilize the measurements between sensors and mobile to estimate the position of mobile. The major measurements are time of arrival (TOA), time different of arrival (TDOA), angle of arrival (AOA) and received signal strength (RSS). In the following, we recommend these four type measurements and model it in Figure 2.1.

1. TOA [45, 46]: Measuring propagation time from mobile to sensors, the delay time $t_{i\bar{j}}$ between transmission at mobile *i* and sensor *j*. Thus, the distance $d_{i\bar{j}}$ between mobile *i* and sensor *j* can be calculated by multiplying the propagation time of the signal propagation speed. The cornerstone of time-based techniques is the receiver's ability to accurately estimate the arrival time of the lone-of-sight (LOS) signal. This estimation is likely to suffer both additive noise and multipath signals. The model of TOA is

$$t_{i\bar{j}} = \frac{r_{i\bar{j}}}{v_c} + w_{i\bar{j}} \qquad (2.1)$$

where $r_{i\bar{j}}$ is real distance between mobile *i* and sensor *j* and $w_{i\bar{j}}$ is measurement noise modeled as AWGN (additive white Gaussian noise), denoted as $N\left(0, \delta_{i\bar{j}}^2\right)$. Then we have the measurement distance by multiplying the propagation speed $v_c$,

$$d_{i\bar{j}} = r_{i\bar{j}} + n_{i\bar{j}} \qquad (2.2)$$

where $n_{i\bar{j}} \sim N\left(0, \sigma_{i\bar{j}}^2\right)$ is AWGN as well. The position of mobile *i* is hided in real distance because of $r_{i\bar{j}} = \sqrt{\left(x_i - \bar{x}_j\right)^2 + \left(y_i - \bar{y}_j\right)^2}$. In this thesis, we only consider the TOA measurement.

2. TDOA [5, 6] : Measuring propagation time difference from different sensors, then we can calculate the measurement distance difference between different sensors to the same mobile. With the cross-correlation of different sensors, the unknown time can be differentiated. The model is given by

$$\left(t_{i\bar{j}} - t_{i\bar{k}}\right) = \frac{\left(r_{i\bar{j}} - r_{i\bar{k}}\right)}{v_c} + \left(w_{i\bar{j}} - w_{i\bar{k}}\right) \qquad (2.3)$$

The measurement difference distance is

$$\left(d_{i\bar{j}} - d_{i\bar{k}}\right) = \left(r_{i\bar{j}} - r_{i\bar{k}}\right) + \left(n_{i\bar{j}} - n_{i\bar{k}}\right) \qquad (2.4)$$

The position of mobile is hided on real distance $r_{i\bar{j}}$ and $r_{i\bar{k}}$ for mobile *i*.

3. RSS [8, 9] : The power on transmitter (sensors or mobile) are known on the system. Measuring the power difference between sensors and mobile to estimate the distance between them. It is can easy to perform by cheap equipment. A model that solely depends on the relative is the so-called Okumura-Hata model [14],

$$P_{loss} = K - 10_{\alpha} \log\left(r_{i\bar{j}}\right) + w_{i\bar{j}} \qquad (2.5)$$

But in the literature, it shows that RSS is not accurate enough because multi-path, noise, humidity, temperature can affect the RSS measurement [36]. As TOA measurement, the position of mobile is hided on $r_{i\bar{j}}$.

4. AOA [7] : The use of directionally sensitive and complex antenna array to estimate the angle of arrival from mobile to sensors. But AOA is disturbed by many factors. For instance, multipath [40], NLOS and so on. The model is

$$\alpha_{i\bar{j}} = \beta_{i\bar{j}} + w_{i\bar{j}} \qquad (2.6)$$

where $\alpha_{i\bar{j}}$ is measurement angle from mobile *i* to sensor *j*, $\beta_{i\bar{j}}$ is real angle and $w_{i\bar{j}}$ is AWGN as well. Every sensors extend the angle to form a intersection which is $(x_i, y_i)$.

TOA is a good candidate in terms of accuracy, and then we utilize TOA to discuss the localization algorithm in following section.

## 2.2 TOA Localization Algorithm

Figure 2.2 shows an uncooperative localization system with TOA measurements. We can estimate mobiles *i* and *j* locations by individual TOA measurements from all sensors. Without loss of generality, we focus on the position of mobile *i*. $r_{i\bar{j}}$ and $d_{i\bar{j}}$ are real distance and measurement distance between mobile *i* and sensor *j*.

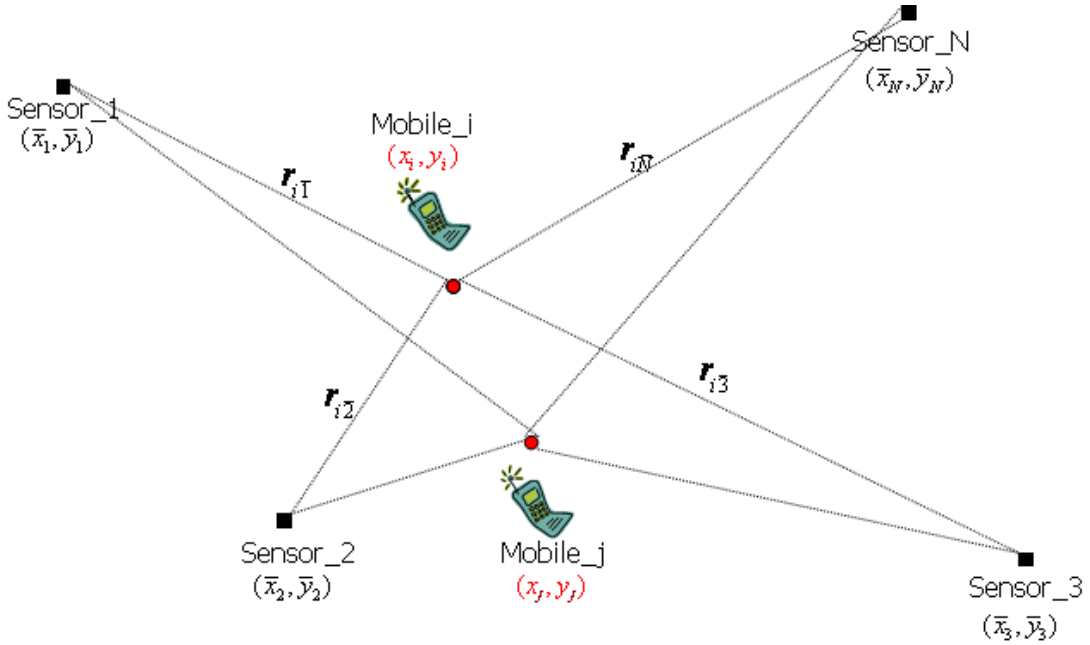



Figure 2.2 A basic localization scenario with TOA measurement.

The TOA from sensors *j* to mobile *i* can be modeled as follows,

$$d_{i\bar{j}} = \left\| \mathbf{x}_i - \bar{\mathbf{x}}_j \right\| + n_{i\bar{j}} = r_{i\bar{j}} + n_{i\bar{j}}, \quad j = 1, 2, \ldots M \qquad (2.7)$$

where $\mathbf{x}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$ is the coordinate vector of mobile *i*, $\bar{\mathbf{x}}_j = \begin{bmatrix} \bar{x}_j & \bar{y}_j \end{bmatrix}^T$ is coordinate vector of sensor *j* and $n_{i\bar{j}}$ is Gaussian noise $N\left(0, \sigma_{i\bar{j}}^2\right)$.

We want to utilize these measurements to estimate the position of mobile *i*. In previous section, there are different algorithms can be solve the problem. We recommend a nonlinear *M*aximum *L*ikelihood (ML) estimator in Section 2.2.1 and other linear estimators are discussed in Section 2.2.2.

### 2.2.1 ML Estimator

According to the model of range error, the probability density function (PDF) of the measurement distance of mobile *i* is

$$p\left(d_{i\bar{j}} \mid \mathbf{x}_i\right) = \frac{1}{\sqrt{2\pi\sigma_{i\bar{j}}^2}} \exp\left( -\frac{\left(d_{i\bar{j}} - \left\|\mathbf{x}_i - \bar{\mathbf{x}}_{\bar{j}}\right\|\right)^2}{2\sigma_{i\bar{j}}^2} \right) \qquad (2.8)$$

Assume the range errors are independent between sensors. The uncooperative likelihood function [25] can be denoted as

$$p\left(\mathbf{d}_i \mid \mathbf{x}_i\right) = \prod_{j=1}^{N} p\left(d_{i\bar{j}} \mid \mathbf{x}_i\right) = \frac{1}{\sqrt{2\pi}^N \prod_{j=1}^{N} \sigma_{i\bar{j}}} \exp\left( -\sum_{j=1}^{N} \frac{\left(d_i - \left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\|\right)^2}{2\sigma_{i\bar{j}}^2} \right) \qquad (2.9)$$

where $\mathbf{d}_i = \left[ d_{i\bar{1}}, d_{i\bar{2}} \cdots d_{i\bar{N}} \right]$ is the measurement set from mobile *i*.

The ML criterion searches a $\hat{\mathbf{x}}_i$ which maximizes the likelihood function (2.9),

$$\max_{\mathbf{x}_i} \left\{ \frac{1}{\sqrt{2\pi}^N \prod_{j=1}^{N} \sigma_{i\bar{j}}} \exp\left( -\sum_{j=1}^{N} \frac{\left(d_{i\bar{j}} - \left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\|\right)^2}{2\sigma_{i\bar{j}}} \right) \right\} \qquad (2.10)$$

In (2.10), it is equivalent to minimization of the summation term. The solution can be rewritten as follows,

$$\min_{\mathbf{x}_i} \left\{ \sum_{j=1}^{N} \frac{\left(d_{i\bar{j}} - \left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\|\right)^2}{2\sigma_{i\bar{j}}^2} \right\} \qquad (2.11)$$

(2.11) is an optimal solution of ML estimator [25], it is also a weighted least-squares (WLS) solution [38]. Ignoring the weight of variance $\sigma_{i\bar{j}}^2$ on (2.11), it can be written as follows

$$\min_{\mathbf{x}_i} \left\{ \sum_{j=1}^{N} \left(d_{i\bar{j}} - \left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\|\right)^2 \right\} \qquad (2.12)$$

In (2.12), it is simpler because we ignore the statistical characteristic of range error; it is least-squares estimation (LSE) [39].

The above algorithm we mention is a nonlinear functions, it can be solved by iterated Nonlinear Least Square Solution [40], but it has to afford high computation cost, then, three linearization algorithms will be introduced in Section 2.2.2.

### 2.2.2 Linearization of Least-Squares Estimator

There are three common linearization methods, Taylor-series expansion algorithm (TS), distance-augmented algorithm (DA) and hyperbolic-canceled algorithm (HC). We will introduce these algorithms in Section 2.2.2a, Section 2.2.2b and Section 2.2.2c respectively. In Section 2.2.2d, we compare the localization accuracy of three linearization methods.

### 2.2.2a Taylor-series Expansion

Our aim is to linearize the nonlinear term, real distance function in (2.7),

$$f_{\bar{j}}(x_i, y_i) = \left\| \mathbf{x}_i - \bar{\mathbf{x}}_j \right\| = \sqrt{\left(x_i - \bar{x}_j\right)^2 + \left(y_i - \bar{y}_j\right)^2}. \quad (2.13)$$

Applying Taylor-Series expansion [35] to (2.13) gives

$$f_{\bar{j}}(x_i, y_i) = f_{\bar{j}}(x_{i0}, y_{i0}) + \left[\nabla^T f_{\bar{j}}(x_{i0}, y_{i0})\right]\Delta + n_{T\_i\bar{j}} \quad (2.14)$$

where $(x_{i0}, y_{i0})$ is the reference point of mobile *i* and the gradient vector

$$\nabla^T f_{\bar{j}}(x_{i0}, y_{i0}) = \left[\frac{\partial f_{\bar{j}}(x_{i0}, y_{i0})}{\partial x_i} \quad \frac{\partial f_{\bar{j}}(x_{i0}, y_{i0})}{\partial y_i}\right] = \left[\frac{x_{i0} - \bar{x}_j}{f_{\bar{j}}(x_{i0}, y_{i0})} \quad \frac{y_{i0} - \bar{y}_j}{f_{\bar{j}}(x_{i0}, y_{i0})}\right],$$

$$\Delta = \begin{bmatrix} x_i - x_{i0} \\ y_i - y_{i0} \end{bmatrix}$$

and $n_{T\_i\bar{j}}$ denotes the higher order truncation error of the Taylor approximation for the distance $r_{i\bar{j}}$.

Rewriting (2.14) as

$$f_{\bar{j}}(x_i, y_i) = r_{i0,\bar{j}} + \frac{x_{i0} - \bar{x}_j}{r_{i0,\bar{j}}}(x_i - x_{i0}) + \frac{y_{i0} - \bar{y}_j}{r_{i0,\bar{j}}}(y_i - y_{i0}) + n_{T\_i\bar{j}} \quad (2.15)$$

where $r_{i0,\bar{j}} = f_{\bar{j}}(x_{i0}, y_{i0})$ is the distance between reference point *i* and sensor *j*.

The measurement model in (2.7) can be written as follows

$$d_{i\bar{j}} = r_{i0,\bar{j}} + \frac{x_{i0} - \bar{x}_j}{r_{i0,\bar{j}}}(x_i - x_{i0}) + \frac{y_{i0} - \bar{y}_j}{r_{i0,\bar{j}}}(y_i - y_{i0}) + n_{T\_i\bar{j}} + n_{i\bar{j}} \qquad (2.16)$$

which is a linear function and can be made into a matrix form as

$$\mathbf{H}_{i\_TS}\mathbf{x}_i = \mathbf{b}_{i\_TS} + \mathbf{n}_{i\_TS} \qquad (2.17)$$

where

$$\mathbf{H}_{i\_TS} = \begin{bmatrix} \frac{x_{i0} - \bar{x}_1}{r_{i0,\bar{1}}} & \frac{y_{i0} - \bar{y}_1}{r_{i0,\bar{1}}} \\ \frac{x_{i0} - \bar{x}_2}{r_{i0,\bar{2}}} & \frac{y_{i0} - \bar{y}_2}{r_{i0,\bar{2}}} \\ \vdots & \vdots \\ \frac{x_{i0} - \bar{x}_N}{r_{i0,\bar{N}}} & \frac{y_{i0} - \bar{y}_N}{r_{i0,\bar{N}}} \end{bmatrix} = \begin{bmatrix} \cos\theta_{i0,\bar{1}} & \sin\theta_{i0,\bar{1}} \\ \cos\theta_{i0,\bar{2}} & \sin\theta_{i0,\bar{2}} \\ \vdots & \vdots \\ \cos\theta_{i0,\bar{N}} & \sin\theta_{i0,\bar{N}} \end{bmatrix}, \qquad (2.18)$$

$$\mathbf{b}_{i\_TS} = \begin{bmatrix} d_{i\bar{1}} - \tilde{r}_{i0,\bar{1}} \\ d_{i\bar{2}} - \tilde{r}_{i0,\bar{2}} \\ \vdots \\ d_{i\bar{N}} - \tilde{r}_{i0,\bar{N}} \end{bmatrix}, \qquad (2.19)$$

$$\mathbf{n}_{i\_TS} = \begin{bmatrix} n_{T\_i\bar{1}} + n_{i\bar{1}} \\ n_{T\_i\bar{2}} + n_{i\bar{2}} \\ \vdots \\ n_{T\_i\bar{N}} + n_{i\bar{N}} \end{bmatrix} \qquad (2.20)$$

and

$$\tilde{r}_{i0,\bar{j}} = r_{i0,\bar{j}} - \left(\frac{x_{i0} - \bar{x}_j}{r_{i0,\bar{j}}}\right)x_{i0} - \left(\frac{y_{i0} - \bar{y}_j}{r_{i0,\bar{j}}}\right)y_{i0}$$

$\mathbf{H}_{i\_TS}$ is an angle matrix and $\mathbf{n}_{i\_TS}$ is a TS error vector which includes measurement error and higher-order error term due to inaccurate reference position.

We can apply weighted least-squares(WLS) solution [38] to get the uncooperative TS estimator $\hat{\mathbf{x}}_{i\_WTS}$

$$\hat{\mathbf{x}}_{i\_WTS} = \left(\mathbf{H}_{i\_TS}{}^{T}\mathbf{W}_{i\_TS}\mathbf{H}_{i\_TS}\right)^{-1}\mathbf{H}_{i\_TS}{}^{T}\mathbf{W}_{i\_TS}\mathbf{b}_{i\_TS} \qquad (2.21)$$

where the uncooperative TS weighting is covariance inverse of the TS error vector

$$\mathbf{W}_{i\_TS} = \left(E\left[\mathbf{n}_{i\_TS}\mathbf{n}_{i\_TS}{}^{T}\right]\right)^{-1} \qquad (2.22)$$

The element of $E\left[\mathbf{n}_{i\_TS}\mathbf{n}_{i\_TS}{}^{T}\right]$ is a diagonal matrix

$$\text{On-diagonal: } E\left[\mathbf{n}_{i\_TS}\mathbf{n}_{i\_TS}{}^{T}\right]_{pp} = \sigma^2_{T\_i\bar{p}} + \sigma^2_{i\bar{p}} \qquad (2.23)$$

$$\text{Off-diagonal: } E\left[\mathbf{n}_{i\_TS}\mathbf{n}_{i\_TS}{}^{T}\right]_{pq} = 0 \qquad (2.24)$$

The resulting covariance matrix of uncooperative TS estimator $\mathbf{e}_{i\_WTS} = \hat{\mathbf{x}}_{i\_WTS} - \mathbf{x}$ is

$$\text{cov}\left(\mathbf{e}_{i\_WTS}\right) = \left(\mathbf{H}_{i\_TS}{}^{T}\mathbf{W}_{i\_TS}\mathbf{H}_{i\_TS}\right)^{-1} \qquad (2.25)$$

The Mean-Square-Error (MSE) of the estimator is

$$\sigma^2{}_{i\_WTS} = trace\left[\left(\mathbf{H}_{i\_TS}{}^{T}\mathbf{W}_{i\_TS}\mathbf{H}_{i\_TS}\right)^{-1}\right] \qquad (2.26)$$

As before if we ignore the statistics $\mathbf{W}_{i\_TS}$, the solution in (2.21) can be further simplified as

$$\hat{\mathbf{x}}_{i\_TS} = \left(\mathbf{H}_{i\_TS}{}^{T}\mathbf{H}_{i\_TS}\right)^{-1}\mathbf{H}_{i\_TS}{}^{T}\mathbf{b}_{i\_TS} \qquad (2.27)$$

and the MSE becomes as

$$\sigma^2{}_{i\_TS} = \text{trace}\left[\left(\mathbf{H}_{i\_TS}{}^{T}\mathbf{H}_{i\_TS}\right)^{-1}\mathbf{H}_{i\_TS}{}^{T}\mathbf{W}_{i\_TS}{}^{-1}\mathbf{H}_{i\_TS}\left(\mathbf{H}_{i\_TS}{}^{T}\mathbf{H}_{i\_TS}\right)^{-1}\right] \qquad (2.28)$$

TS algorithm can calculate a quite accurate solution with a very good reference point and avoid the high cost of nonlinear iteration. In fact, we can get the better reference point by updating the reference point in (2.21).

$$\hat{\mathbf{x}}_{i\_WTS}(k+1) = \left(\mathbf{H}_{i\_TS}{}^{T}(k)\mathbf{W}_{i\_TS}(k)\mathbf{H}_{i\_TS}(k)\right)^{-1}\mathbf{H}_{i\_TS}{}^{T}(k)\mathbf{W}_{i\_TS}(k)\mathbf{b}_{i\_TS}(k) \qquad (2.29)$$

where

$$\mathbf{H}_{i\_TS}(k)=\begin{bmatrix} \frac{x_{i0}(k)-\bar{x}_1}{r_{i0,\bar{1}}(k)} & \frac{y_{i0}(k)-\bar{y}_1}{r_{i0,\bar{1}}(k)} \\ \frac{x_{i0}(k)-\bar{x}_2}{r_{i0,\bar{2}}(k)} & \frac{y_{i0}(k)-\bar{y}_2}{r_{i0,\bar{2}}(k)} \\ \vdots & \vdots \\ \frac{x_{i0}(k)-\bar{x}_N}{r_{i0,\bar{N}}(k)} & \frac{y_{i0}(k)-\bar{y}_N}{r_{i0,\bar{N}}(k)} \end{bmatrix}=\begin{bmatrix} \cos_{i0,\bar{1}}(k) & \sin_{i0,\bar{1}}(k) \\ \cos_{i0,\bar{2}}(k) & \sin_{i0,\bar{2}}(k) \\ \vdots & \vdots \\ \cos_{i0,\bar{N}}(k) & \sin_{i0,\bar{N}}(k) \end{bmatrix}, \quad (2.30)$$

$$\mathbf{W}_{i\_TS}(k)=\begin{bmatrix} \frac{1}{\sigma^2_{T\_i\bar{1}}(k)+\sigma^2_{i\bar{1}}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma^2_{T\_i\bar{2}}(k)+\sigma^2_{i\bar{2}}} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{\sigma^2_{T\_i\bar{N}}(k)+\sigma^2_{i\bar{N}}} \end{bmatrix}, \quad (2.31)$$

$$\mathbf{b}_{i\_TS}(k)=\begin{bmatrix} d_{i\bar{1}}-\tilde{r}_{i0,\bar{1}}(k) \\ d_{i\bar{2}}-\tilde{r}_{i0,\bar{2}}(k) \\ \vdots \\ d_{i\bar{N}}-\tilde{r}_{i0,\bar{N}}(k) \end{bmatrix}, \quad (2.32)$$

and $k$ is the iteration index.

The reference point is replaced by the TS estimator at global index $k$, $\left(x_{i0}(k), y_{i0}(k)\right)=\hat{\mathbf{x}}_{i\_WTS}(k)$. In (2.26) and (2.28), we know the angle matrix $\mathbf{H}_{i\_TS}$ and uncooperative TS weighting matrix $\mathbf{W}_{i\_TS}$ can affect the localization accuracy. We will compare it with other two linearized algorithms in Section 2.2.2d.

However, we know this method has sensitive reference point. If we have a good point, the localization is quite accurate. Then, we can use some very simple method to find a not-too-bad point based on measurement distances. For example, in a 20m x 20m room, we have four known sensor locations on the four corners. According to four measurement distances, we can use proportion to find a simple reference point.

### 2.2.2b Distance-Canceled

The other linear algorithm is distance-canceled (DA) algorithm [46], we summarize it as follows. First squaring up (2.7) gives

$$d_{i\bar{j}}^{\,2} = \left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\|^2 + 2\left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\| n_{i\bar{j}} + n_{i\bar{j}}^{\,2} \qquad (2.33)$$

(2.13) is a vector form, we use a scalar form to indicate

$$d_{i\bar{j}}^{\,2} = \left(x_i - \bar{x}_j\right)^2 + \left(y_i - \bar{y}_j\right)^2 + 2r_{i\bar{j}} n_{i\bar{j}} + n_{i\bar{j}}^{\,2} \qquad (2.34)$$

Expansion of (2.34) gives

$$d_{i\bar{j}}^{\,2} = x_i^{\,2} + y_i^{\,2} + \bar{x}_j^{\,2} + \bar{y}_j^{\,2} - 2x_i\bar{x}_j - 2y_i\bar{y}_j + 2r_{i\bar{j}} n_{i\bar{j}} + n_{i\bar{j}}^{\,2} \qquad (2.35)$$

we know the nonlinear term is $x_i^{\,2} + y_i^{\,2}$. We augment a squared distance variable

$$R_i = x_i^{\,2} + y_i^{\,2} \qquad (2.36)$$

then (2.35) can be rewritten as follows

$$2x_i\bar{x}_j + 2y_i\bar{y}_j - R_i = \bar{g}_j - d_{i\bar{j}}^{\,2} + 2r_{i\bar{j}} n_{i\bar{j}} + n_{i\bar{j}}^{\,2}. \qquad (2.37)$$

Rewrite (2.37) in a matrix form as follows

$$\mathbf{H}_{DA}\tilde{\mathbf{x}}_i = \mathbf{b}_{i\_DA} + \mathbf{n}_{i\_DA} \qquad (2.38)$$

where

$$\mathbf{H}_{DA} = \begin{bmatrix} 2\bar{x}_1 & 2\bar{y}_1 & -1 \\ 2\bar{x}_2 & 2\bar{y}_2 & -1 \\ \vdots & \vdots & \vdots \\ 2\bar{x}_N & 2\bar{y}_N & -1 \end{bmatrix}, \qquad (2.39)$$

$$\tilde{\mathbf{x}}_i = \begin{bmatrix} x_i \\ y_i \\ R_i \end{bmatrix} = \begin{bmatrix} \mathbf{x}_i^{\,T} \\ R_i \end{bmatrix}, \qquad (2.40)$$

$$\mathbf{b}_{i\_DA} = \begin{bmatrix} \bar{g}_1 - d_{i\bar{1}}^2 \\ \bar{g}_2 - d_{i\bar{2}}^2 \\ \vdots \\ \bar{g}_N - d_{i\bar{N}}^2 \end{bmatrix}, \qquad (2.41)$$

and

$$\mathbf{n}_{i\_DA} = \begin{bmatrix} 2r_{i\bar{1}}n_{i\bar{1}} + n_{i\bar{1}}{}^2 \\ 2r_{i\bar{2}}n_{i\bar{2}} + n_{i\bar{2}}{}^2 \\ \vdots \\ 2r_{i\bar{N}}n_{i\bar{N}} + n_{i\bar{N}}{}^2 \end{bmatrix}. \qquad (2.42)$$

$\mathbf{H}_{DA}$ is a coordinate matrix and $\mathbf{n}_{i\_DA}$ is a DA error vector. Here, the solution vector $\tilde{\mathbf{x}}_i$ (2.34) is different from other two linearized method's $\mathbf{x}_i$ because we augment a variable $R_i$. In (2.38) we do not know the DA error vector, the matrix function we meet

$$\mathbf{H}_{DA}\tilde{\mathbf{x}}_i \approx \mathbf{b}_{i\_DA} \qquad (2.43)$$

As before, we can apply WLS solution to get uncooperative DA estimator

$$\hat{\tilde{\mathbf{x}}}_{i\_WDA} = \left(\mathbf{H}_{DA}{}^T\mathbf{W}_{i\_DA}\mathbf{H}_{DA}\right)^{-1}\mathbf{H}_{DA}{}^T\mathbf{W}_{i\_DA}\mathbf{b}_{i\_DA} \qquad (2.44)$$

where the uncooperative DA weighting $\mathbf{W}_{i\_DA}$ is the covariance inverse of the DA error vector

$$\mathbf{W}_{i\_DA} = \left(E\left[\mathbf{n}_{i\_DA}\mathbf{n}_{i\_DA}{}^T\right]\right)^{-1} \qquad (2.45)$$

and $E\left[\mathbf{n}_{i\_DA}\mathbf{n}_{i\_DA}{}^T\right]$ is a diagonal matrix as well

$$\text{On-diagonal: } E\left[\mathbf{n}_{i\_DA}\mathbf{n}_{i\_DA}{}^T\right]_{pp} = 4r_{i\bar{p}}^2\sigma_{i\bar{p}}^2 + 3\sigma_{i\bar{p}}^2 \qquad (2.46)$$

$$\text{Off-diagonal: } E\left[\mathbf{n}_{i\_DA}\mathbf{n}_{i\_DA}{}^T\right]_{pq} = 0 \qquad (2.47)$$

and the resulting covariance matrix of uncooperative DA estimator error $\mathbf{e}_{i\_WDA} = \hat{\mathbf{x}}_{i\_WDA} - \mathbf{x}$ is

$$\mathrm{cov}\left(\mathbf{e}_{i\_WDA}\right)=\left(\mathbf{H}_{DA}{}^{T}\mathbf{W}_{i\_DA}\mathbf{H}_{DA}\right)^{-1} \qquad (2.48)$$

The size of covariance matrix is 3x3 and it is different from other two linearized methods. Then, the MSE of the estimator is

$$\sigma^{2}{}_{i\_WDA}=trace\left[\left(\mathbf{H}_{DA}{}^{T}\mathbf{W}_{i\_DA}\mathbf{H}_{DA}\right)^{-1}\right]_{2\times 2} \qquad (2.49)$$

As before, if we ignore the statistics $\mathbf{W}_{i\_DA}$, the solution in (2.44) can be further simplified as

$$\hat{\hat{\mathbf{x}}}_{i\_DA}=\left(\mathbf{H}_{DA}{}^{T}\mathbf{H}_{DA}\right)^{-1}\mathbf{H}_{DA}{}^{T}\mathbf{b}_{i\_DA} \qquad (2.50)$$

and the MSE becomes as

$$\sigma^{2}{}_{i\_DA}=\mathrm{trace}\left[\left(\mathbf{H}_{DA}{}^{T}\mathbf{H}_{DA}\right)^{-1}\mathbf{H}_{DA}{}^{T}\mathbf{W}_{i\_DA}{}^{-1}\mathbf{H}_{DA}\left(\mathbf{H}_{DA}{}^{T}\mathbf{H}_{DA}\right)^{-1}\right]_{2\times 2} \qquad (2.51)$$

From (2.49) and (2.51), we know that coordinate matrix $\mathbf{H}_{DA}$ and DA weighting matrix $\mathbf{W}_{i\_DA}$ can affect the localization accuracy. DA algorithm is very easy to operate but it also suffers from new variable $R_i$. However, $R_i$ is not independent on the variable of position mobile *i* in WLS solution. The accuracy might be the worst .

## 2.2.2c Hyperbolic-Canceled

Now, we introduce the HC algorithm [22]. In (2.35) we know that $x_i^{2}+y_i^{2}$ is nonlinear term. Then, we choice the sensor *k* as a reference sensor. The reference equation is

$$d_{i\bar{k}}{}^{2}=x_i^{2}+y_i^{2}+\bar{x}_k{}^{2}+\bar{y}_k{}^{2}-2x_i\bar{x}_k-2y_i\bar{y}_k+2r_{i\bar{k}}n_{i\bar{k}}+n_{i\bar{k}}{}^{2} \qquad (2.52)$$

In order to cancel the nonlinear term, subtraction of (2.35) from (2.52) gives

$$2x_i\left(\bar{x}_j-\bar{x}_k\right)+2y_i\left(\bar{y}_j-\bar{y}_k\right)=d_{i\bar{k}}{}^{2}-d_{i\bar{j}}{}^{2}+\bar{g}_j-\bar{g}_k+2r_{i\bar{j}}n_{i\bar{j}}-2r_{i\bar{k}}n_{i\bar{k}}+n_{i\bar{j}}{}^{2}-n_{i\bar{k}}{}^{2},\ \ j\neq k$$

(2.53)

where $\bar{g}_k = \bar{x}_k^{\,2} + \bar{y}_k^{\,2}$, $\bar{g}_j = \bar{x}_j^{\,2} + \bar{y}_j^{\,2}$

Without loss of generality, let *k=1*. (2.53) is a linear function, we can rewrite (2.53) as a matrix form

$$\mathbf{H}_{HC}\mathbf{x}_i = \mathbf{b}_{i\_HC} + \mathbf{n}_{i\_HC} \qquad (2.54)$$

where

$$\mathbf{H}_{HC} = \begin{bmatrix} (\bar{x}_2 - \bar{x}_1) & (\bar{y}_2 - \bar{y}_1) \\ (\bar{x}_3 - \bar{x}_1) & (\bar{y}_3 - \bar{y}_1) \\ \vdots & \vdots \\ (\bar{x}_N - \bar{x}_1) & (\bar{y}_N - \bar{y}_1) \end{bmatrix}, \qquad (2.55)$$

$$\mathbf{b}_{i\_HC} = \frac{1}{2}\begin{bmatrix} d_{i\bar{1}}^2 - d_{i\bar{2}}^2 + \bar{g}_2 - \bar{g}_1 \\ d_{i\bar{1}}^2 - d_{i\bar{3}}^2 + \bar{g}_3 - \bar{g}_1 \\ \vdots \\ d_{i\bar{1}}^2 - d_{i\bar{N}}^2 + \bar{g}_N - \bar{g}_1 \end{bmatrix}, \qquad (2.56)$$

and

$$\mathbf{n}_{i-HC} = \begin{bmatrix} r_{i\bar{2}}n_{i\bar{2}} - r_{i\bar{1}}n_{i\bar{1}} + \frac{1}{2}\left(n_{i\bar{2}}^2 + n_{i\bar{1}}^2\right) \\ r_{i\bar{3}}n_{i\bar{3}} - r_{i\bar{1}}n_{i\bar{1}} + \frac{1}{2}\left(n_{i\bar{3}}^2 + n_{i\bar{1}}^2\right) \\ \vdots \\ r_{i\bar{N}}n_{i\bar{N}} - r_{i\bar{1}}n_{i\bar{1}} + \frac{1}{2}\left(n_{i\bar{N}}^2 + n_{i\bar{1}}^2\right) \end{bmatrix}. \qquad (2.57)$$

$\mathbf{H}_{HC}$ is a coordinate-difference matrix and $\mathbf{n}_{i\_HC}$ is a HC error vector for estimated mobile *i*.

In fact, we do not know the HC error vector in (2.54). The matrix equation we meet

$$\mathbf{H}_{HC}\mathbf{x}_i \approx \mathbf{b}_{i\_HC} \qquad (2.58)$$

We can apply WLS solution to get the uncooperative HC estimator

$$\hat{\mathbf{x}}_{i\_WHC} = \left(\mathbf{H}_{HC}^{\,T}\mathbf{W}_{i\_HC}\mathbf{H}_{HC}\right)^{-1}\mathbf{H}_{HC}^{\,T}\mathbf{W}_{i\_HC}\mathbf{b}_{i\_HC} \qquad (2.59)$$

where the uncooperative HC weighting matrix $\mathbf{W}_{i\_HC}$ is the covariance inverse of the HC error vector

$$\mathbf{W}_{i\_HC} = \left( E\left[ \mathbf{n}_{i\_HC} \mathbf{n}_{i\_HC}{}^{T} \right] \right)^{-1} \qquad (2.60)$$

Then, the covariance of $E\left[ \mathbf{n}_{i\_HC} \mathbf{n}_{i\_HC}{}^{T} \right]$ is

$$\text{On-diagonal: } E\left[ \mathbf{n}_{i\_HC} \mathbf{n}_{i\_HC}{}^{T} \right]_{pp} = r_{i\bar{p}}^2 \sigma_{\bar{i}\bar{p}}^2 + r_{i\bar{1}}^2 \sigma_{i\bar{1}}^2 + \frac{3}{4}(\sigma_{\bar{i}\bar{p}}^2 - \sigma_{i\bar{1}}^2) \qquad (2.61)$$

$$\text{Off-diagonal: } E\left[ \mathbf{n}_{i\_HC} \mathbf{n}_{i\_HC}{}^{T} \right]_{pq} = r_{i\bar{1}}^2 \sigma_{i\bar{1}}^2 + \frac{3}{4} \sigma_{i\bar{1}}^2 \qquad (2.62)$$

It is not diagonal matrix anymore. The resulting covariance matrix of the uncooperative HC estimator error $\mathbf{e}_{i\_WHC} = \hat{\mathbf{x}}_{i\_WHC} - \mathbf{x}$ is

$$\text{cov}\left( \mathbf{e}_{i\_WHC} \right) = \left( \mathbf{H}_{HC}{}^{T} \mathbf{W}_{i\_HC} \mathbf{H}_{HC} \right)^{-1} \qquad (2.63)$$

We can further get the MSE of the estimator

$$\sigma^2{}_{i\_WHC} = trace\left[ \left( \mathbf{H}_{HC}{}^{T} \mathbf{W}_{i\_HC} \mathbf{H}_{HC} \right)^{-1} \right] \qquad (2.64)$$

If we ignore the statistic $\mathbf{W}_{i\_HC}$, the solution in (2.59) can be further simplified as

$$\hat{\mathbf{x}}_{i\_HC} = \left( \mathbf{H}_{HC}{}^{T} \mathbf{H}_{HC} \right)^{-1} \mathbf{H}_{HC}{}^{T} \mathbf{b}_{i\_HC} \qquad (2.65)$$

and the MSE becomes as

$$\sigma^2{}_{i\_HC} = \text{trace}\left[ \left( \mathbf{H}_{HC}{}^{T} \mathbf{H}_{HC} \right)^{-1} \mathbf{H}_{HC}{}^{T} \mathbf{W}_{i\_HC}{}^{-1} \mathbf{H}_{HC} \left( \mathbf{H}_{HC}{}^{T} \mathbf{H}_{HC} \right)^{-1} \right] \qquad (2.66)$$

From (2.51) and (2.53), we know that coordinate-difference matrix $\mathbf{H}_{HC}$ and uncooperative HC weighting matrix $\mathbf{W}_{i\_HC}$ can affect the localization accuracy. Then, we will compare $\mathbf{H}_{\text{algorithm}}$ and $\mathbf{W}_{\text{algorithm}}$ of three linearized algorithms in Section 2.2.2d.

### 2.2.2d Summary of Three Linearization Algorithms

We know three linearization algorithms have similar matrix equation as $\mathbf{Hx} = \mathbf{b} + \mathbf{n}$. Now, we compare **H**, **x**, and **n**. Now, we can see the **H** for TS is angle matrix in (2.18)

$$\mathbf{H}_{i\_TS} = \begin{bmatrix} \frac{x_{i0} - \bar{x}_1}{r_{i0,\bar{1}}} & \frac{y_{i0} - \bar{y}_1}{r_{i0,\bar{1}}} \\ \frac{x_{i0} - \bar{x}_2}{r_{i0,\bar{2}}} & \frac{y_{i0} - \bar{y}_2}{r_{i0,\bar{2}}} \\ \vdots & \vdots \\ \frac{x_{i0} - \bar{x}_N}{r_{i0,\bar{N}}} & \frac{y_{i0} - \bar{y}_N}{r_{i0,\bar{N}}} \end{bmatrix} = \begin{bmatrix} \cos\theta_{i0,\bar{1}} & \sin\theta_{i0,\bar{1}} \\ \cos\theta_{i0,\bar{2}} & \sin\theta_{i0,\bar{2}} \\ \vdots & \vdots \\ \cos\theta_{i0,\bar{N}} & \sin\theta_{i0,\bar{N}} \end{bmatrix} \quad (2.18)$$

which consists of angles from sensors. Then, **H** for DA is coordinate matrix in (2.39)

$$\mathbf{H}_{DA} = \begin{bmatrix} 2\bar{x}_1 & 2\bar{y}_1 & -1 \\ 2\bar{x}_2 & 2\bar{y}_2 & -1 \\ \vdots & \vdots & \vdots \\ 2\bar{x}_N & 2\bar{y}_N & -1 \end{bmatrix} \quad (2.39)$$

It is made up of coordinates for all sensors. The last **H** of HC is coordinate-difference matrix in (2.55)

$$\mathbf{H}_{HC} = \begin{bmatrix} (\bar{x}_2 - \bar{x}_1) & (\bar{y}_2 - \bar{y}_1) \\ (\bar{x}_3 - \bar{x}_1) & (\bar{y}_3 - \bar{y}_1) \\ \vdots & \vdots \\ (\bar{x}_N - \bar{x}_1) & (\bar{y}_N - \bar{y}_1) \end{bmatrix}, \quad (2.55)$$

It is made up of differences of coordinates based on reference mobile *1*. However, only the variable $\mathbf{x} = [x \quad y \quad R]$ of DA is different from other linearized methods' $\mathbf{x} = [x \quad y]$. Finally, we discuss **n** from three parts, noise source, algorithm weighting and covariance matrix of estimator error as follows.

(1) Noise Source:

Noise source can affect the theoretical MSE. In HC algorithm, after linearization

operation, the HC error term is $r_{i\bar{j}}n_{i\bar{j}} - r_{i\bar{1}}n_{i\bar{1}} + \frac{1}{2}\left(n_{i\bar{j}}{}^2 - n_{i\bar{1}}{}^2\right)$ which is involved in the effect of real distance $r_{i\bar{j}}$ and $r_{i\bar{1}}$. The effect amplifies the measurement error and destroys the theoretical MSE. However, in TS algorithm, after linearization operation, the TS error term is $n_{T\_i\bar{j}} + n_{i\bar{j}}$. If the reference position is perfectly ideal, TS error term only involves the measurement error $n_{i\bar{j}}$. Then, in DA algorithm, after linearization operation, the DA error term is $2r_{i\bar{j}}n_{i\bar{j}} + n_{i\bar{j}}{}^2$ which is also involved in real distance effect. In addition, it has a new variable $R_i$. In fact, the new variable is not independent on position of mobile in WLS solution (2.44). Therefore, it has the worst MSE performance. After the comparison of noise source, we will compare the weighting matrix for these linearized algorithms.

(2) Weighting Matrix:

First, We assume variances of measurement errors are the same as $\sigma^2$. We know the weighting matrix is gotten from noise source covariance. Then, the uncooperative HC weighting matrix is

$$\mathbf{W}_{i\_Hyper} = \frac{1}{\sigma^2}\begin{bmatrix} r_{i\bar{2}}^2 + r_{i\bar{1}}^2 + \frac{3}{2} & r_{i\bar{1}}^2 + \frac{3}{4} & \cdots & r_{i\bar{1}}^2 + \frac{3}{4} \\ r_{i\bar{1}}^2 + \frac{3}{4} & r_{i\bar{3}}^2 + r_{i\bar{1}}^2 + \frac{3}{2} & r_{i\bar{1}}^2 + \frac{3}{4} & \vdots \\ \vdots & r_{i\bar{1}}^2 + \frac{3}{4} & \ddots & r_{i\bar{1}}^2 + \frac{3}{4} \\ r_{i\bar{1}}^2 + \frac{3}{4} & \cdots & r_{i\bar{1}}^2 + \frac{3}{4} & r_{i\bar{N}}^2 + r_{i\bar{1}}^2 + \frac{3}{2} \end{bmatrix}^{-1} \quad (2.67)$$

In (2.53), uncooperative HC weighting matrix compensates the HC error term with real distance effect. In fact, it is related with the reference point mobile *i*.

Next, in TS algorithm, we assume the reference position is very good $n_{T\_i\bar{j}} \approx 0$.

Therefore, the uncooperative TS weighting matrix is

$$\mathbf{W}_{i\_TS} \approx \frac{1}{\sigma^2}\mathbf{I}_{N\times N} \qquad (2.68)$$

In (2.70), TS weighting compensates Taylor error term $n_{i\bar{1}}$ which is also measurement error. Finally, in DA algorithm, the uncooperative DA weighting can be rewritten as follow as

$$\mathbf{W}_{i\_DA} = \frac{1}{\sigma^2}\begin{bmatrix} \frac{1}{4r_{i\bar{1}}^2+3} & 0 & \cdots & 0 \\ 0 & \frac{1}{4r_{i\bar{2}}^2+3} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{4r_{i\bar{N}}^2+3} \end{bmatrix} \qquad (2.69)$$

In (2.71), DA weighting compensates the error term with real distance effect. However, before the compensation of weighting matrix, the best MSE performance is uncooperative TS estimator and the best is the DA estimator. After the compensation of weighted, their have similar theoretical MSE. The computer simulation result for theoretical MSE of three linearized algorithms will show in Figures 5.2 and 5.3.

(3) Covariance Matrix of Estimator Error

The covariance matrix includes matrix $\mathbf{H}_{\text{algorithm}}$ and weighting matrix $\mathbf{W}_{\text{algorithm}}$ because of $\text{cov}\left(\mathbf{e}_{\text{algorithm}}\mathbf{e}_{\text{algorithm}}^T\right) = \left(\mathbf{H}_{\text{algorithm}}{}^T\mathbf{W}_{\text{algorithm}}\mathbf{H}_{\text{algorithm}}\right)^{-1}$. Then, the HC algorithm's is difficult to calculate because it's covariance of HC error vector is not a diagonal matrix in (2.61) and (2.62). Therefore, we discuss TS algorithm and DA algorithm. However, in TS algorithm, the covariance matrix of estimator error in (2.26) included angle matrix $\mathbf{H}_{i\_TS}$ and its weighting matrix $\mathbf{W}_{i\_TS}$ is given by

$$\operatorname{cov}\left(\mathbf{e}_{i\_WTS}\right)=\left\{\frac{1}{\sigma^2}\sum_{j=1}^{N}\begin{bmatrix}\dfrac{\left(x_{i0}-\bar{x}_j\right)^2}{r_{i0,\bar{j}}^2} & \dfrac{\left(x_{i0}-\bar{x}_j\right)\left(y_{i0}-\bar{y}_j\right)}{r_{i0,\bar{j}}^2}\\ \dfrac{\left(x_{i0}-\bar{x}_j\right)\left(y_{i0}-\bar{y}_j\right)}{r_{i0,\bar{j}}^2} & \dfrac{\left(y_{i0}-\bar{y}_j\right)^2}{r_{i0,\bar{j}}^2}\end{bmatrix}\right\}^{-1} \tag{2.70}$$

If the reference point $\left(x_{i0},y_{i0}\right)$ is very close to true point $\left(x_i,y_i\right)$, (2.70) can be approximated as follows

$$\operatorname{cov}\left(\mathbf{e}_{i\_WTS}\right)\approx\sigma^2\left\{\sum_{j=1}^{N}\begin{bmatrix}\dfrac{\left(x_i-\bar{x}_j\right)^2}{r_{i\bar{j}}^2} & \dfrac{\left(x_i-\bar{x}_j\right)\left(y_i-\bar{y}_j\right)}{r_{i\bar{j}}^2}\\ \dfrac{\left(x_i-\bar{x}_j\right)\left(y_i-\bar{y}_j\right)}{r_{i\bar{j}}^2} & \dfrac{\left(y_i-\bar{y}_j\right)^2}{r_{i\bar{j}}^2}\end{bmatrix}\right\}^{-1} \tag{2.71}$$

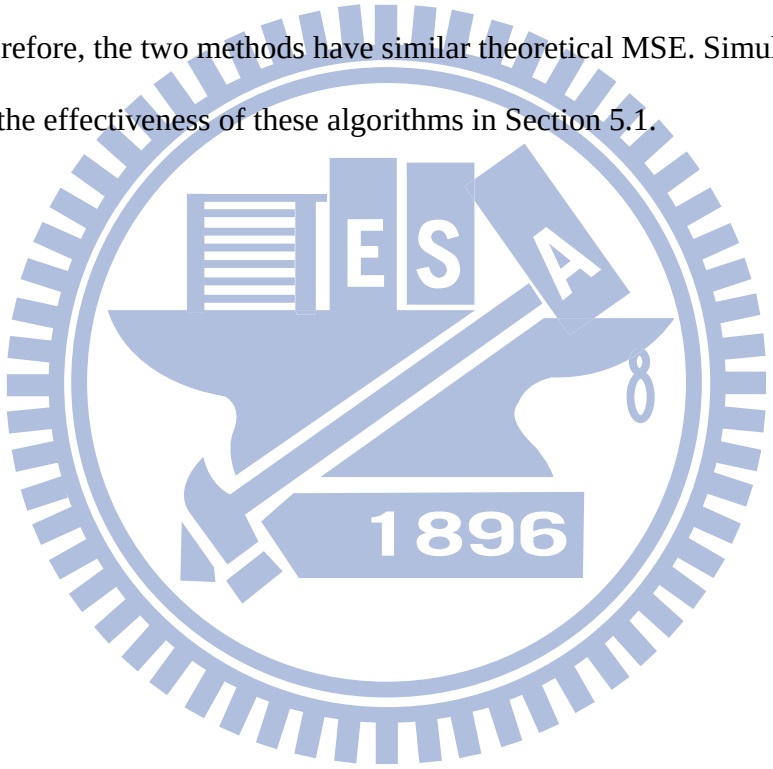
In fact, the right hand side of (2.71) is the full uncooperative Fisher Information Matrix (FIM) (4.4) inverse and is proportion to the noise variance $\sigma^2$. Therefore, the theoretical MSE of TS estimator is very close to uncooperative CRLB. As before, the covariance of estimator error for DA algorithm can be formed as

$$\operatorname{cov}\left(\mathbf{e}_{i\_DA}\right)=\sigma^2\left\{\sum_{j=1}^{N}\begin{bmatrix}\dfrac{4\bar{x}_j^2}{4r_{i\bar{j}}^2+3} & \dfrac{4\bar{x}_j\bar{y}_j}{4r_{i\bar{j}}^2+3} & -\dfrac{2\bar{x}_j}{4r_{i\bar{j}}^2+3}\\ \dfrac{4\bar{x}_j\bar{y}_j}{4r_{i\bar{j}}^2+3} & \dfrac{4\bar{y}_j^2}{4r_{i\bar{j}}^2+3} & -\dfrac{2\bar{y}_j}{4r_{i\bar{j}}^2+3}\\ -\dfrac{2\bar{x}_j}{4r_{i\bar{j}}^2+3} & -\dfrac{2\bar{y}_j}{4r_{i\bar{j}}^2+3} & \dfrac{1}{4r_{i\bar{j}}^2+3}\end{bmatrix}\right\}^{-1} \tag{2.72}$$

The covariance matrix is proportion to the noise variance $\sigma^2$ as well. If $r_{j\bar{i}}^2\gg 3$, (2.53) can be simplified as follows

$$\operatorname{cov}\left(\mathbf{e}_{i\_WDA}\right)=\sigma^2\left\{\sum_{i=1}^{M}\begin{bmatrix}\frac{\overline{x}_j^2}{r_{i\bar{j}}^2} & \frac{\overline{x}_j\overline{y}_j}{r_{i\bar{j}}^2} & -\frac{\overline{x}_j}{2r_{i\bar{j}}^2}\\ \frac{\overline{x}_j\overline{y}_j}{r_{i\bar{j}}^2} & \frac{\overline{y}_j^2}{r_{i\bar{j}}^2} & -\frac{\overline{y}_j}{2r_{i\bar{j}}^2}\\ -\frac{\overline{x}_j}{2r_{i\bar{j}}^2} & -\frac{\overline{y}_j}{2r_{i\bar{j}}^2} & \frac{1}{4r_{i\bar{j}}^2}\end{bmatrix}\right\}^{-1} \qquad (2.73)$$

There are negative numbers in (2.73), it can reduce the covariance matrix. Then, the theoretical MSE of DA estimator is larger than TS estimator's. Finally, the noise source of HC algorithm is similar to DA algorithm's expect the new augmented variable. Therefore, the two methods have similar theoretical MSE. Simulation results demonstrate the effectiveness of these algorithms in Section 5.1.



# Chapter 3
# Cooperative Localization Algorithm

In cooperative system, cooperative relay or Ad Hoc short-rang communication among the terminals will be supported [41, 42]. They consider the short-range has lower measurement error interference, therefore it can enhance the location estimation accuracy. Then, they investigate the data fusion of large-scale and small-scale. [26] proposed Cooperative Localization with Optimum Quality of Estimate (CLOQ) which takes advantage of the behavior of the channel to provide accurate indoor positioning. This algorithm uses the quality of ranging and positioning estimates to provide practical and accurate results. More importantly, it reduces error propagation substantially. If Non-Line-Of-Sight (NLOS) exists, cooperative group localization (CGL) scheme is proposed in [13] based on rigid graph theory.

However, cooperative localization is not a well solved problem because the distance measurement between any pairs of unknown positions (mobiles) are utilized to assist in the location estimation. This is much more challenging than the traditional localization where only distance measurements between unknown position (one mobile) and known positions (sensors) are employed for localization. Therefore, [12] devised subspace approach to solve that problem and which can outperform the classical MDS algorithm. The other chooses is cooperative ML estimator which can be solved by joint Newton's algorithm. But the computation cost is quite high because it is nonlinear iterated solution. Base on the linearized algorithm of uncooperative localization, we propose joint Taylor-series expansion algorithm to reduce the cost. But other linear operation can not linearize the cooperative TOA measurement to form joint algorithms. Therefore, we devise divided-and-conquer method to overcome

that problem and it can also reduce computation cost from joint Newton's algorithm. In divided-and-conquer method, we need uncertain virtual sensors to achieve the method. [15] proposed an error propagation aware algorithm to track the extent of the uncertain virtual position error. It sets threshold to decide which virtual can be involve in localization. By recursive position estimation, the more estimated locations are selected as virtual sensor locations until all estimated positions are virtual sensor locations. In fact, some bad virtual sensor positions still provide the information to localization and the selection of threshold has trade-off between computation cost and localization accuracy. Therefore, our proposed algorithm consider all virtual sensors to help by global iteration and we also discuss the computation cost.

Figure 3.1 indicates the cooperative localization system. As before, there are $N$ known positions of sensors and $M$ unknown positions of mobiles. The two individual localization system for mobile $i$ and mobile $j$ are cooperated by cooperative TOA measurement, $d_{ij}$. Then, $r_{ij}$ is real cooperative distance from mobiles $i$ and $j$.

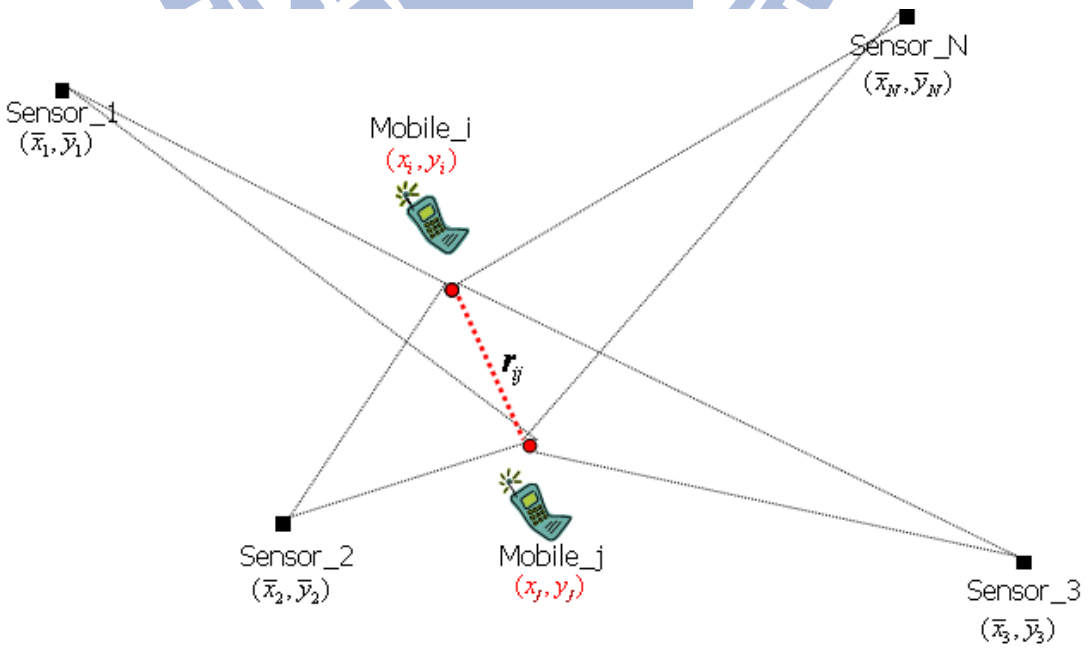



Figure 3.1 Cooperative localization system with cooperative TOA measurement.

The dot line distance denotes cooperative distance that connects two mobiles. All measurements of cooperative localization are illustrated as follows

The uncooperative measurement between mobile *i* and sensor *j* is denoted as

$$d_{i\bar{j}} = r_{i\bar{j}} + n_{i\bar{j}}, \quad i = 1 \sim M, \quad j = 1 \sim N \qquad (3.1)$$

and the cooperative measurement between mobile *i* and mobile *j* is denoted as

$$d_{ij} = r_{ij} + n_{ij}, \quad i < j, \quad i, j = 1 \sim M \qquad (3.2)$$

where $n_{i\bar{j}}$ and $n_{ij}$ is uncooperative and cooperative measurement error belong to AWGN $n_{i\bar{j}} \sim N\left(0, \sigma_{i\bar{j}}^2\right)$ and $n_{ij} \sim N\left(0, \sigma_{ij}^2\right)$

There are *M* positions of mobile will be estimated, the cooperative likelihood function (ML) [42] is similar to uncooperative likelihood function in (2,9) and it is written as follows

$$p\left(\mathbf{d}_{Uncoop}, \mathbf{d}_{Coop} \mid \mathbf{x}\right) = \underbrace{\prod_{i=1}^{M}\prod_{j=1}^{N} \frac{1}{\sqrt{2\sigma_{i\bar{j}}^2}} \exp\left(-\frac{\left(d_{i\bar{j}} - \left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\|\right)^2}{2\sigma_{i\bar{j}}^2}\right)}_{Uncooperation} \cdot \underbrace{\prod_{\substack{i,j=1 \\ j>i}}^{M} \frac{1}{\sqrt{2\sigma_{ij}^2}} \exp\left(-\frac{\left(d_{ij} - \left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)^2}{2\sigma_{ij}^2}\right)}_{Cooperation} \qquad (3.3)$$

where cooperative observation set $\mathbf{d}_{i\_Uncoop}$ is like $\mathbf{d}_i$ in (2.9), cooperative observation set $\mathbf{d}_{Coop} = \left[d_{12}, d_{13}, \cdots, d_{1M}, d_{23}, d_{24}, \cdots d_{2M}, \cdots d_{M-1,M}\right]$ and all positions of mobiles $\mathbf{x} = \left[\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_M\right]^T$.

The cooperative ML criterion searches a $\hat{\mathbf{x}}$ which maximizes likelihood function (3.3),

$$\min_{\hat{\mathbf{x}}} \left\{ \underbrace{\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{1}{2\sigma_{i\bar{j}}^2}\left(d_{i\bar{j}} - \left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\|\right)^2}_{Noncooperation} + \underbrace{\sum_{\substack{i=1 \\ j>i}}^{M} \frac{1}{2\sigma_{ij}^2}\left(d_{ij} - \left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)^2}_{Cooperation} \right\} \qquad (3.4)$$

where $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_1 \quad \hat{\mathbf{x}}_2 \quad \cdots \quad \hat{\mathbf{x}}_M]^T$ is the cooperative ML estimator.

We will introduce a common joint Newton's method to solve it. However, joint Newton's method is nonlinear function and it computation cost is quite high. Therefore we propose two new methods to reduce it, joint Taylor-series expansion algorithm (joint TS) and divide-and-conquer method (divided algorithms). The structure of the rest of this section is as follows. Section 3.1 discusses joint cooperative algorithms. The divide-and-conquer method is proposed in Section 3.2. In Section 3.3, discusses the issue of divide-and-conquer method, compensation of uncertain virtual sensor.

## 3.1 Joint Cooperative Algorithm

According to cooperative algorithms, we will introduce a common nonlinear joint Newton's method to solved (3.4) in 3.1.1 and we propose a joint TS in Section 3.1.2. In 3.1.3, we will illustrate why the other linearized methods can't form joint algorithms.

### 3.1.1 Newton's Method

In order to minimize the object function in (3.4), we can set its gradient function to zero and to get the estimated positions. Let the object function be denoted as

$$G(\mathbf{x}) = \sum_{i=1}^{M}\sum_{\bar{j}=1}^{N} \frac{1}{2\sigma_{i\bar{j}}^2}\left(d_{i\bar{j}} - \left\|\mathbf{x}_i - \bar{\mathbf{x}}_j\right\|\right)^2 + \sum_{\substack{i=1 \\ j>i}}^{M} \frac{1}{2\sigma_{ij}^2}\left(d_{ij} - \left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)^2 \qquad (3.5)$$

and

$$\nabla_{\mathbf{x}}^T G(\mathbf{x}) = \left[\frac{\partial G(\mathbf{x})}{\partial x_1} \quad \frac{\partial G(\mathbf{x})}{\partial y_1} \quad \frac{\partial G(\mathbf{x})}{\partial x_2} \quad \frac{\partial G(\mathbf{x})}{\partial y_2} \quad \cdots \quad \frac{\partial G(\mathbf{x})}{\partial y_M}\right] = \mathbf{0} \qquad (3.6)$$

We can use Newton's method [26] to solve (3.6) and it is donated as follows

$$\hat{\mathbf{x}}_{\text{Joint\_Newton}}(k+1)=\hat{\mathbf{x}}_{\text{Joint\_Newton}}(k)-\left[\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{T}G\left(\hat{\mathbf{x}}_{\text{Joint\_Newton}}(k)\right)\right]_{2M\times 2M}^{-1}\nabla_{\mathbf{x}}^{T}G\left(\hat{\mathbf{x}}_{\text{Joint\_Newton}}(k)\right)$$

(3.7)

where $k$ is the iteration index,

$$\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{T}G(\hat{\mathbf{x}})=\begin{bmatrix}\frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_1\partial x_1} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_1\partial y_1} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_1\partial x_2} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_1\partial y_2} & \cdots & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_1\partial y_M}\\ \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_1\partial x_1} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_1\partial y_1} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_1\partial x_2} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_1\partial y_2} & \cdots & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_1\partial y_M}\\ \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_2\partial x_1} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_2\partial y_1} & \ddots & & & \\ \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_2\partial x_1} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_2\partial y_1} & & \ddots & & \vdots\\ \vdots & \vdots & & & \ddots & \\ \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_M\partial x_1} & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_M\partial y_1} & & \cdots & & \frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_M\partial y_M}\end{bmatrix} \quad (3.8)$$

is the Jacobian matrix [23] whose element is

$$\frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_p\partial x_p}=\sum_{j=1}^{N}\frac{1}{\sigma_{p\bar{j}}^2}\left[\frac{\left(d_{p\bar{j}}-r_{p\bar{j}}\right)\left(x_p-\bar{x}_j\right)^2}{r_{p\bar{j}}^3}+\frac{\left(x_p-\bar{x}_j\right)^2}{r_{p\bar{j}}^2}-\frac{\left(d_{p\bar{j}}-r_{p\bar{j}}\right)}{r_{p\bar{j}}}\right]$$
$$+\sum_{\substack{i=1\\ i<p}}^{p-1}\frac{1}{\sigma_{ip}^2}\left[+\frac{\left(d_{ip}-r_{ip}\right)\left(x_i-x_p\right)^2}{r_{ip}^3}+\frac{\left(x_i-x_p\right)^2}{r_{ip}^2}-\frac{\left(d_{ip}-r_{ip}\right)}{r_{ip}}\right]$$
$$+\sum_{\substack{i=p+1\\ i>p}}^{M}\frac{1}{\sigma_{ip}^2}\left[+\frac{\left(d_{pi}-r_{ip}\right)\left(x_p-x_i\right)^2}{r_{pi}^3}+\frac{\left(x_p-x_i\right)^2}{r_{pi}^2}-\frac{\left(d_{pi}-r_{pi}\right)}{r_{pi}}\right]$$

$$\frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_p\partial y_p}=\sum_{j=1}^{N}\frac{1}{\sigma_{p\bar{j}}^2}\left[\frac{\left(d_{p\bar{j}}-r_{p\bar{j}}\right)\left(x_p-\bar{x}_j\right)\left(y_p-\bar{y}_j\right)}{r_{p\bar{j}}^3}+\frac{\left(x_p-\bar{x}_j\right)\left(y_p-\bar{y}_j\right)}{r_{p\bar{j}}^2}\right]$$
$$+\sum_{\substack{i=1\\ i<p}}^{p-1}\frac{1}{\sigma_{ip}^2}\left[+\frac{\left(d_{ip}-r_{ip}\right)\left(x_i-x_p\right)\left(y_i-y_p\right)}{r_{ip}^3}+\frac{\left(x_i-x_p\right)\left(y_i-y_p\right)}{r_{ip}^2}\right]$$
$$+\sum_{\substack{i=p+1\\ i>p}}^{M}\frac{1}{\sigma_{pi}^2}\left[+\frac{\left(d_{pi}-r_{pi}\right)\left(x_p-x_i\right)\left(y_p-y_i\right)}{r_{pi}^3}+\frac{\left(x_p-x_i\right)\left(y_p-y_i\right)}{r_{pi}^2}\right]$$

$$\frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_p\partial x_q}=-\frac{1}{\sigma_{pq}^2}\left[\frac{\left(d_{pq}-r_{pq}\right)\left(x_p-x_q\right)^2}{r_{pq}^3}+\frac{\left(x_p-x_q\right)^2}{r_{pq}^2}-\frac{\left(d_{pq}-r_{pq}\right)}{r_{pq}}\right]$$

$$\frac{\partial^2 G(\hat{\mathbf{x}})}{\partial x_p \partial y_q} = -\frac{1}{\sigma_{pq}^2}\left[\frac{\left(d_{pq}-r_{pq}\right)\left(x_p-x_q\right)\left(y_p-y_q\right)}{r_{pq}^3}+\frac{\left(x_p-x_q\right)\left(y_p-y_q\right)}{r_{pq}^2}\right]$$

$$\begin{aligned}\frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_p \partial y_p} = &\sum_{j=1}^{N}\frac{1}{\sigma_{p\bar{j}}^2}\left[\frac{\left(d_{p\bar{j}}-r_{p\bar{j}}\right)\left(y_p-\bar{y}_j\right)^2}{r_{p\bar{j}}^3}+\frac{\left(y_p-\bar{y}_j\right)^2}{r_{p\bar{j}}^2}-\frac{\left(d_{p\bar{j}}-r_{p\bar{j}}\right)}{r_{p\bar{j}}}\right]\\ &+\sum_{\substack{i=1\\ i<p}}^{p-1}\frac{1}{\sigma_{ip}^2}\left[\frac{\left(d_{ip}-r_{ip}\right)\left(y_i-y_p\right)^2}{r_{ip}^3}+\frac{\left(y_i-y_p\right)^2}{r_{ip}^2}-\frac{\left(d_{ip}-r_{ip}\right)}{r_{ip}}\right]\\ &+\sum_{\substack{i=p+1\\ i>p}}^{M}\frac{1}{\sigma_{ip}^2}\left[\frac{\left(d_{pi}-r_{ip}\right)\left(y_p-y_i\right)^2}{r_{pi}^3}+\frac{\left(y_p-y_i\right)^2}{r_{pi}^2}-\frac{\left(d_{pi}-r_{pi}\right)}{r_{pi}}\right]\end{aligned}$$

$$\frac{\partial^2 G(\hat{\mathbf{x}})}{\partial y_p \partial y_q} = -\frac{1}{\sigma_{pq}^2}\left[\frac{\left(d_{pq}-r_{pq}\right)\left(y_p-y_q\right)^2}{r_{pq}^3}+\frac{\left(y_p-y_q\right)^2}{r_{pq}^2}-\frac{\left(d_{pq}-r_{pq}\right)}{r_{pq}}\right]$$

Actually, the Jacobian matrix inverse part $\left[\nabla_{\mathbf{x}}^T\nabla_{\mathbf{x}}G\left(\hat{\mathbf{x}}_{\text{Joint\_Newton}}(k)\right)\right]_{2M\times 2M}^{-1}\nabla_{\mathbf{x}}G\left(\hat{\mathbf{x}}_{\text{Joint\_Newton}}(k)\right)$ in (3.7) can be achieved by $\nabla_{\mathbf{x}}^T\nabla_{\mathbf{x}}G\left(\hat{\mathbf{x}}_{\text{Joint\_Newton}}(k)\right)\backslash\nabla_{\mathbf{x}}G\left(\hat{\mathbf{x}}(k)\right)$ of Gaussian elimination method in MATLAB function. Even if the Gaussian elimination method replaces that part, but the size of Jacobian matrix is quite large and the elements have to calculate a lot of summations when it exists multitudinous mobiles. We assume one global iteration in (3.7) needs $F_{2M\times 2M}$ flops. *G* global iterations are needed to converge in (3.7). Then, the total computation cost is given by

$$\text{Joint Newton computation cost} = F_{2M\times 2M}\times G_{\text{joint}}\ \text{flops.} \qquad (3.9)$$

Then, we derive linearized method, joint TS to reduce the calculation complexity $F_{2M\times 2M}$ for Gaussian elimination method of large matrix in following section.

### 3.1.2 Taylor-Series Expansion Algorithm

Now we use Taylor-series approximation method to linearize cooperative nonlinear function. According to (2.17) in Section 2.2.2a, we have *M* uncooperative

TS matrix equation as follows

$$\mathbf{H}_{i\_TS}\mathbf{x}_i = \mathbf{b}_{i\_TS} + \mathbf{n}_{i\_TS} \quad i = 1, 2, ..., M \qquad (3.10)$$

Now, the remaining task for us is to linearize cooperative distances $r_{ij}$ among mobiles. We know the relationship of real distance from mobile *i* to mobile *j* is

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = f_{ij}(\Delta x_{ij}, \Delta y_{ij}) \qquad (3.11)$$

Because there are four variables in a cooperative real distance (includes two positions of mobiles), we can regard the difference variable as a new variable, difference-variable $\Delta x_{ij}$ and $\Delta y_{ij}$.

$$\begin{bmatrix} x_i - x_j \\ y_i - y_j \end{bmatrix} = \begin{bmatrix} \Delta x_{ij} \\ \Delta y_{ij} \end{bmatrix} \qquad (3.12)$$

later we will convert these difference-variable $\Delta x_{ij}, \Delta y_{ij}$ back to four original variable $x_i$, $y_i$, $x_j$, $y_j$.

Apply Taylor-series expansion to (3.11) as follows.

$$f_{ij}(\Delta x_{ij}, \Delta y_{ij}) = f_{ij}(\Delta x_{ij0}, \Delta y_{ij0}) + \left[\nabla^T f_{ij}(\Delta x_{ij0}, \Delta y_{ij0})\right]\Delta + n_{T\_ij} \qquad (3.13)$$

where $(\Delta x_{ij0}, \Delta y_{ij0})$ is the difference reference point , $n_{T\_ij}$ is the higher order truncation error of the Taylor-series expansion for the distance $r_{ij}$,

$$\nabla^T f_{ij}(\Delta x_{ij0}, \Delta y_{ij0}) = \left[\frac{\partial f_{ij}(\Delta x_{ij0}, \Delta y_{ij0})}{\partial \Delta x_{ij}} \quad \frac{\partial f_{ij}(\Delta x_{ij0}, \Delta y_{ij0})}{\partial \Delta y_{ij}}\right] = \left[\frac{\Delta x_{ij0}}{f_{ij}(\Delta x_{ij0}, \Delta y_{ij0})} \quad \frac{\Delta y_{ij0}}{f_{ij}(\Delta x_{ij0}, \Delta y_{ij0})}\right]$$

and

$$\Delta = \begin{bmatrix} \Delta x_{ij} - \Delta x_{ij0} \\ \Delta y_{ij} - \Delta y_{ij0} \end{bmatrix}.$$

Now the cooperative measurement model (3.2) becomes

$$d_{ij} = r_{ij} + n_{ij} = r_{ij0} + \frac{\Delta x_{ij0}}{r_{ij0}}\left(\Delta x_{ij} - \Delta x_{ij0}\right) + \frac{\Delta y_{ij0}}{r_{ij0}}\left(\Delta x_{ij} - \Delta x_{ij0}\right) + n_{Tij} + n_{ij} \qquad (3.14)$$

where $r_{ij0} = f_{ij}(\Delta x_{ij0}, \Delta y_{ij0})$.

Knowing that (3.14) is a linear function of difference-variable $(\Delta x_{ij0}, \Delta y_{ij0})$, we may form a cooperative TS equation

$$(\frac{\Delta x_{ij0}}{r_{ij0}})\Delta x_{ij} + (\frac{\Delta y_{ij0}}{r_{ij0}})\Delta y_{ij} = d_{ij} - \tilde{r}_{ij0} + n_{T\_ij} + n_{ij} \qquad (3.15)$$

where $\tilde{r}_{ij0} = r_{ij0} - \left(\frac{\Delta x_{ij0}}{r_{ij0}}\right)\Delta x_{ij0} - \left(\frac{\Delta y_{ij0}}{r_{ij0}}\right)\Delta y_{ij0}$

The cooperative TS equation in (3.15), distance-variable $\left(\Delta x_{ij}, \Delta y_{ij}\right)^T$ is different form the variables of uncooperative matrixes equation in (2.17) for mobile *i* and mobile *j* with original variables $\mathbf{x}_i$ and $\mathbf{x}_j$. Consequently we convert the difference-variable back to original variable. (3.15) can be written as follow

$$\left[(\frac{\Delta x_{ij0}}{r_{ij0}}) \quad (\frac{\Delta y_{ij0}}{r_{ij0}})\right]\mathbf{x}_i - \left[(\frac{\Delta x_{ij0}}{r_{ij0}}) \quad (\frac{\Delta y_{ij0}}{r_{ij0}})\right]\mathbf{x}_j = d_{ij} - \tilde{r}_{ij0} + n_{T\_ij} + n_{ij} \qquad (3.16)$$

The cooperative TS equation (3.14) has the original variable $\left(\mathbf{x}_i, \mathbf{x}_j\right)^T$ which is like uncooperative TS matrix equation's for mobile *i* and mobile *j*. Therefore, we can combine two equations to form a joint TS matrix equaiton. In case of *M=4*, the joint TS cooperative matrix equation is given by

$$\begin{bmatrix} \mathbf{H}_{1\_TS} & 0 & 0 & 0 \\ 0 & \mathbf{H}_{2\_TS} & 0 & 0 \\ 0 & 0 & \mathbf{H}_{3\_TS} & 0 \\ 0 & 0 & 0 & \mathbf{H}_{4\_TS} \\ \mathbf{h}_{120}^T & -\mathbf{h}_{120}^T & 0 & 0 \\ \mathbf{h}_{130}^T & 0 & -\mathbf{h}_{130}^T & 0 \\ \mathbf{h}_{140}^T & 0 & 0 & -\mathbf{h}_{140}^T \\ 0 & \mathbf{h}_{230}^T & -\mathbf{h}_{230}^T & 0 \\ 0 & \mathbf{h}_{240}^T & 0 & -\mathbf{h}_{240}^T \\ 0 & 0 & \mathbf{h}_{340}^T & -\mathbf{h}_{340}^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ d_{12} - \tilde{r}_{120} \\ d_{13} - \tilde{r}_{130} \\ d_{14} - \tilde{r}_{140} \\ d_{23} - \tilde{r}_{230} \\ d_{24} - \tilde{r}_{240} \\ d_{34} - \tilde{r}_{340} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{1\_TS} \\ \mathbf{n}_{2\_TS} \\ \mathbf{n}_{3\_TS} \\ \mathbf{n}_{4\_TS} \\ n_{T\_12} + n_{12} \\ n_{T\_13} + n_{13} \\ n_{T\_14} + n_{14} \\ n_{T\_23} + n_{24} \\ n_{T\_24} + n_{24} \\ n_{T\_34} + n_{34} \end{bmatrix} \quad (3.17)$$

Where

$$\mathbf{h}_{ij0} = \begin{bmatrix} \frac{\Delta x_{ij0}}{r_{ij0}} & \frac{\Delta y_{ij0}}{r_{ij0}} \end{bmatrix}^T = \begin{bmatrix} \cos\theta_{ij0} & \sin\theta_{ij0} \end{bmatrix}^T.$$

is a cooperative angle vector between two reference points $(x_{i0}, y_{i0})$ and $(x_{j0}, y_{j0})$. Here, there are $C(M,2) = 6$ pairs cooperative measurements between mobiles. We note a angle vector in (3.17) for one pair is $\begin{bmatrix} \mathbf{h}_{ij0} & -\mathbf{h}_{ij0} \end{bmatrix}$ because of the difference-variable in (3.12). In addition, (3,17) can be simplified as

$$\mathbf{H}_{\text{Joint\_TS}}\mathbf{x} = \mathbf{b}_{\text{Joint\_TS}} + \mathbf{n}_{\text{Joint\_TS}} \quad (3.18)$$

where $\mathbf{H}_{\text{Joint\_TS}}$ is joint TS angle matrix.

As before, the joint TS estimator is given by

$$\hat{\mathbf{x}}_{\text{Joint\_TS}} = \left(\mathbf{H}_{\text{Joint\_TS}}{}^T \mathbf{W}_{\text{Joint\_TS}} \mathbf{H}_{\text{Joint\_TS}}\right)^{-1} \mathbf{H}_{\text{Joint\_TS}}{}^T \mathbf{W}_{\text{Joint\_TS}} \mathbf{b}_{\text{Joint\_TS}} \quad (3.19)$$

where the joint TS weighting matrix, $\mathbf{W}_{\text{Joint\_TS}} = E\left[\mathbf{n}_{\text{Joint\_TS}} \mathbf{n}_{\text{Joint\_TS}}^T\right]^{-1}$.

It is like (2.29). We also can get the better reference point set by updating the reference point set from joint TS estimator in (3.19).

$$\hat{\mathbf{x}}_{\text{Joint\_TS}}(k+1) = \left(\mathbf{H}_{\text{Joint\_TS}}{}^T(k) \mathbf{W}_{\text{Joint\_TS}}(k) \mathbf{H}_{\text{Joint\_TS}}(k)\right)^{-1} \mathbf{H}_{\text{Joint\_TS}}{}^T(k) \mathbf{W}_{\text{Joint\_TS}}(k) \mathbf{b}_{\text{Joint\_TS}}(k)$$

(3.20)

In the same reason, the solution (3.20) can be solved by Gaussian elimination method in MATLAB function, $\hat{\mathbf{x}}_{\text{Joint\_TS}} = \sqrt{\mathbf{W}_{\text{Joint\_TS}}}\mathbf{H}_{\text{Joint\_TS}} \setminus \sqrt{\mathbf{W}_{\text{Joint\_TS}}}\mathbf{b}_{\text{Joint\_TS}}$. Even if the mobiles more, the size of joint TS angle matrix larger, but it doesn't do a lot of summations. However, it can reduce the computation cost of joint Newton's method. Simulations are presented in terms of the MSE, convergence rate and computation cost for two joint algorithms in Section 5.2. Alternatively, we also propose a divide-and-conquer method the reduce the cost of calculation complexly (3.7) in Section 3.2

### 3.1.3 Other Joint Linearization Algorithms

We will brief illustrate that distance-augmented method and hyperbolic-canceled method can not achieve joint algorithms. We know the new challenge to form joint algorithm is to linearize the cooperative real distance $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. However, in DA method, we square the measurement model before linearized operation. Therefore, (3.7) can be operated as follows,

$$\begin{aligned} d_{ij}^2 &= (x_i - x_j)^2 + (y_i - y_j)^2 + 2r_{ij}n_{ij} + n_{ij}^2 \\ &= R_i + R_j - 2x_ix_j - 2y_iy_j + 2r_{ij}n_{ij} + n_{ij}^2 \end{aligned} \qquad (3.21)$$

In (3.21) even if we augment the new variable $R_i$ and $R_j$, it still exist the nonlinear term $x_ix_j$ and $y_iy_j$. Therefore, DA method can not achieve joint algorithm. Of course, HC method has the same problem because it has to do square operation. Therefore, we do not have joint DA and joint HC algorithms. However, if $(x_j, y_j)$ are known, (3.21) can be linearized. Therefore, we proposed divide-and-conquer to solve that problem.

## 3.2 Divide-and-Conquer Method

We know that cooperative localization is involved unknown positions connection. It is more difficult than uncooperative localization. Therefore, we can simplify it by divide-and-conquer method. We illustrate the method Figures 3.2 to 3.4.

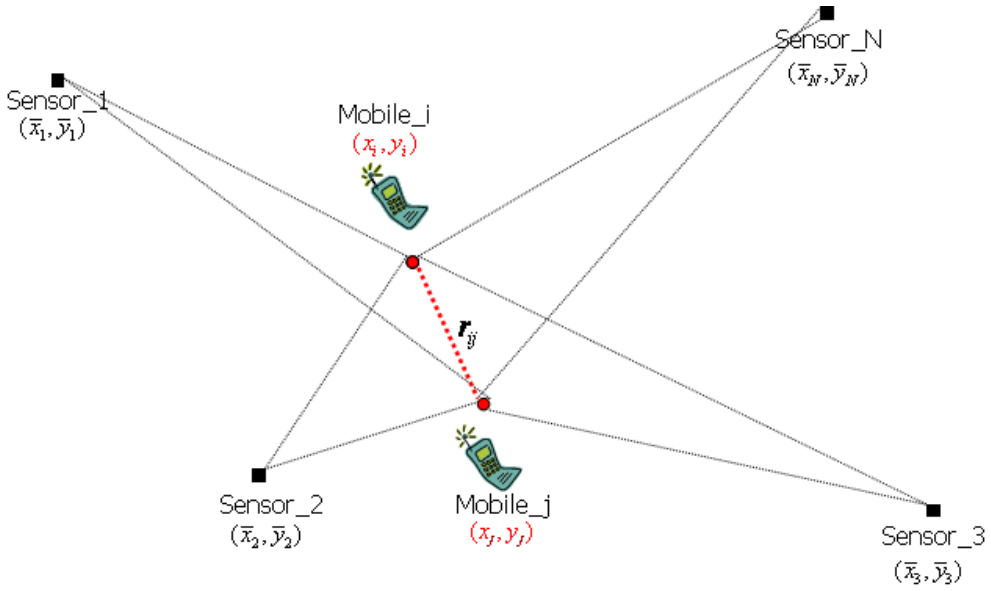



Figure 3.2 The joint localization method.

If the mobile *j* location is know as $(\hat{x}_j, \hat{y}_j)$, we have a individual localization in Figure 3.3.

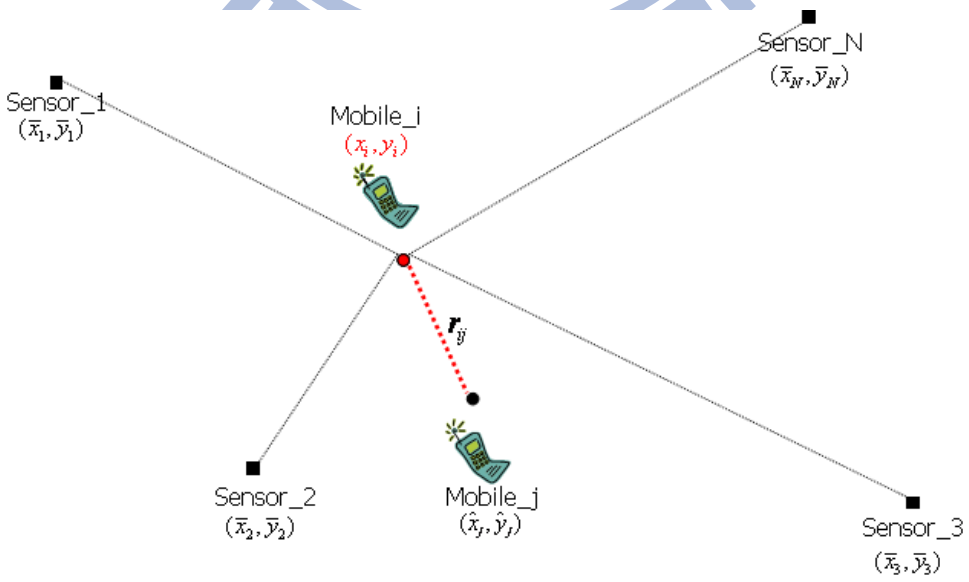



Figure 3.3 A part of divide-and-conquer method.

As before, if mobile *j* location is known, we have another part of divide-and-conquer method in Figure 3.4.

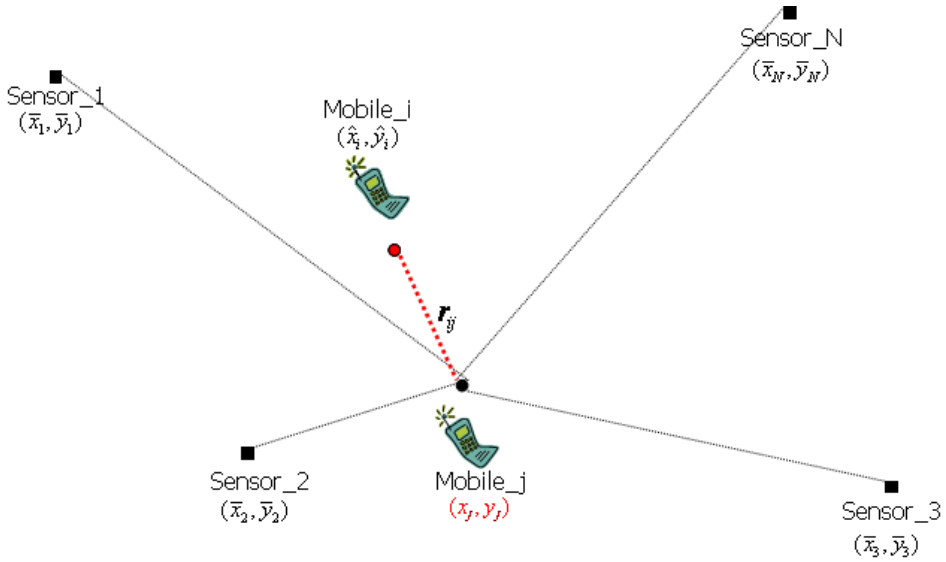



Figure 3.4 Another part of divide-and-conquer method.

However, divide-and-conquer method can reduce the computation cost in (3.7). the system model is described in follows. At first, every position of mobiles can be estimated from uncooperative ML estimator. Then, we have initial virtual sensors.

$$\min_{\mathbf{x}_{i,0}} \underbrace{\left\{ \sum_{j=1}^{N} \frac{1}{2\sigma_{\bar{i}\bar{j}}^2} \left( d_{i\bar{j}} - \left\| \mathbf{x}_i - \bar{\mathbf{x}}_j \right\| \right)^2 \right\}}_{Noncooperation}, i = 1 \sim M \qquad (3.22)$$

From (3.22), we have *M* initial virtual sensors $\hat{\mathbf{x}}_{1,0}, \hat{\mathbf{x}}_{2,0}, \cdots, \hat{\mathbf{x}}_{M,0}$. Next, we divide *M* virtual ML functions to estimate *M* positions of mobiles again. Every function can be achieved by other *M-1* positions of virtual sensors and it can be updated until their locations are converged, we call global iteration. According to the sequential of virtual sensors location updating, there are two type sequential, Jacobi method and Gauss-Seidel method. The detail of divide-and-conquer method and sequential of virtual sensors updating are discuss in Section 3.2.1. A joint Newton's algorithm and three joint linearized algorithms perform the proposed method are discussed in

Section 3.2.2.

### 3.2.1 Two Category of Update Sequence

We will introduce Jacobi method and Gauss-Seidel method for divide-and-conquer method in Section 3.2.1a and Section 3.2.1b.

### 3.2.1a Jacobi Method

Now, we describe the divide-and-conquer method for Jacobi method [23]. After we have initial virtual sensors from uncooperative algorithm, we start form mobile *1*. Then, the positions of virtual sensor *2* to virtual sensor *M* will be helped to estimate the position of mobile *1* at *1*st global iteration. The divided ML estimator for the mobile *1* is given by

$$\min_{\mathbf{x}_{1,1}} \left\{ \sum_{j=1}^{N} \frac{1}{2\sigma_{1\bar{j}}^2} \left( d_{1\bar{j}} - \left\| \mathbf{x}_1 - \bar{\mathbf{x}}_j \right\| \right)^2 + \sum_{l=2}^{M} \frac{1}{2\sigma_{1\hat{l}}^2} \left( d_{1l} - \left\| \mathbf{x}_1 - \hat{\mathbf{x}}_{l,0} \right\| \right)^2 \right\} \tag{3.23}$$

It is similar uncooperative ML estimator (2.11) because its only includes one unknown position $\mathbf{x}_1$ of mobile *1*, while virtual sensors locations $\hat{\mathbf{x}}_{l,0},\ l = 2,3,...,M,$ are already known. After estimating position of mobile *1*, do the same step to get virtual sensor positions $\hat{\mathbf{x}}_{2,1}, \hat{\mathbf{x}}_{3,1}, ..., \hat{\mathbf{x}}_{M,1}$ from mobile *2* to mobile *M*. This procedure is called a global iteration, as shown in the right-hand-side of Figure 3.5. Next, this global iteration can be repeated to update positions from $\hat{\mathbf{x}}_{1,n}, \hat{\mathbf{x}}_{2,n}, ..., \hat{\mathbf{x}}_{M,n}$ to $\hat{\mathbf{x}}_{1,n+1}, \hat{\mathbf{x}}_{2,n+1}, ..., \hat{\mathbf{x}}_{M,n+1}$. The divided ML estimator for mobile *i* at $n^{\text{th}}$ is given by

$$\min_{\mathbf{x}_{i,n+1}} \left\{ \sum_{j=1}^{N} \frac{1}{2\sigma_{i\bar{j}}^2} \left( d_{i\bar{j}} - \left\| \mathbf{x}_i - \bar{\mathbf{x}}_j \right\| \right)^2 + \sum_{\substack{l=1 \\ l \neq i}}^{M} \frac{1}{2\sigma_{i\hat{l}}^2} \left( d_{il} - \left\| \mathbf{x}_i - \hat{\mathbf{x}}_{l,n} \right\| \right)^2 \right\}, i = 1,2,...,M \tag{3.24}$$

where $\hat{\mathbf{x}}_{i,n}$ denotes position of virtual sensor *i* at the $n^{th}$ global iteration. This divide-and-conquer method in (3.24) stops until all positions of virtual sensors are converged.

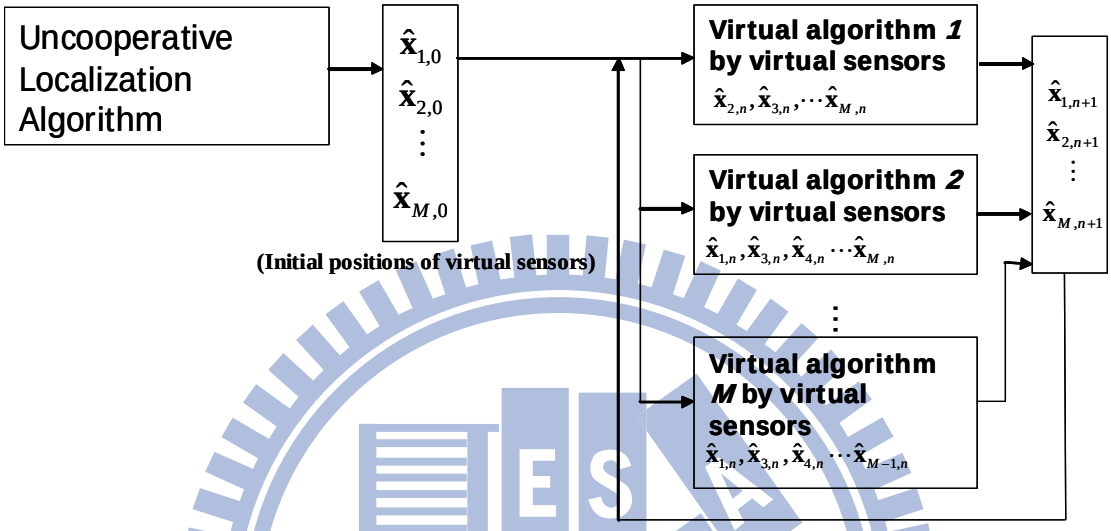



Figure 3.5 The Jacobi method diagram for divide-and-conquer method.

### 3.2.1b Gauss-Seidel Method

The only difference between Gauss-Seidel method [23] and Jacobi method is that in the former, the most recently update step. First, we start from mobile *1*, the positions of initial virtual sensors $\hat{\mathbf{x}}_{2,0}, \hat{\mathbf{x}}_{3,0}, \cdots, \hat{\mathbf{x}}_{M,0}$ and positions of sensors $\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \cdots, \overline{\mathbf{x}}_N$ estimate position of mobile *1* to get virtual sensor *1*, $\hat{\mathbf{x}}_{1,1}$ at $1^{st}$ global iteration. Next, we estimate mobile *2* by position virtual sensors $\hat{\mathbf{x}}_{1,1}, \hat{\mathbf{x}}_{3,0}, \cdots, \hat{\mathbf{x}}_{M,0}$. We can see the position of virtual sensor *1*, $\hat{\mathbf{x}}_{1,1}$ had been updated in virtual sensors locations. In Jacobi method, it uses $\hat{\mathbf{x}}_{1,0}, \hat{\mathbf{x}}_{3,0}, \cdots, \hat{\mathbf{x}}_{M,0}$ to estimate position of mobile *2*. The divided ML of Gauss-Seidel method is given by

$$\min_{\hat{\mathbf{x}}_{i,n+1}} \left\{ \begin{array}{l} \sum_{j=1}^{N} \frac{1}{2\sigma_{i\bar{j}}^2} \left( d_{i\bar{j}} - \left\| \mathbf{x}_{i,n+1} - \bar{\mathbf{x}}_j \right\| \right)^2 \\ + \sum_{l<i} \frac{1}{2\sigma_{i\hat{l},n+1}^2} \left( d_{il} - \left\| \mathbf{x}_{i,n+1} - \hat{\mathbf{x}}_{l,n+1} \right\| \right)^2 + \sum_{l>i} \frac{1}{2\sigma_{i\hat{l},n}^2} \left( d_{il} - \left\| \mathbf{x}_{i,n+1} - \hat{\mathbf{x}}_{l,n} \right\| \right)^2 \end{array} \right\}, i = 1 \sim M$$

(3.25)

Usually, the convergence rate of Gauss-Seidel method is faster than Jacobi method if the method is convergent.

We know that the virtual sensors locations can destroy the localization accuracy because their positions are uncertain. Therefore, we have to compensate the weighted $1/2\sigma_{i\hat{l}}^2$ in the virtual sensor terms $\sum_{\substack{l=1 \\ l\neq i}}^{M} 1/2\sigma_{i\hat{l}}^2 \left( d_{il} - \left\| \mathbf{x}_i - \hat{\mathbf{x}}_{l,n} \right\| \right)^2$. We will derive the compensation in Section 3.3.

### 3.2.2 Divided individual localization: Newton's Algorithm and Three Linearization Algorithms

For easy to describe, we use Newton's method and three divided linearization algorithms to perform divided-and-conquer method under Jacobi method. The divided ML estimator in (3.24) can be achieved by these algorithms, because it is similar with the uncooperative ML estimator in (2.11). Therefore, the Divided Newton's algorithm is described in Section 3.2.2a, three divided linearization algorithms in Sections 3.2.2b, 3.2.2c and 3.2.2d.

### 3.2.2a Newton's Algorithm

In divided ML estimator (3.24), it can be solved by Newton's method. As before, the solution of Newton's method for mobile *i* at $(n+1)^{th}$ global iteration can be written as

$$\hat{\mathbf{x}}_{i,n+1}(k+1)=\hat{\mathbf{x}}_{i,n+1}(k)-\left[\nabla_{\mathbf{x}_i}\nabla_{\mathbf{x}_i}^{T}G_{i,n}\left(\hat{\mathbf{x}}_{i,n+1}(k)\right)\right]_{2\times 2}^{-1}\ \nabla_{\mathbf{x}_i}^{T}G_{i,n}\left(\hat{\mathbf{x}}_{i,n+1}(k)\right) \qquad (3.26)$$

$$G_{i,n}\left(\mathbf{x}_{i,n+1}\right)=\sum_{j=1}^{N}\frac{1}{2\sigma_{i\bar{j}}^{2}}\left(d_{i\bar{j}}-\left\|\mathbf{x}_{i,n+1}-\bar{\mathbf{x}}_{j}\right\|\right)^{2}+\sum_{\substack{l=1\\ l\neq i}}^{M}\frac{1}{2\sigma_{i\hat{l},n}^{2}}\left(d_{il}-\left\|\mathbf{x}_{i,n+1}-\hat{\mathbf{x}}_{l,n}\right\|\right)^{2},$$

$$\nabla_{\mathbf{x}_i}\nabla_{\mathbf{x}_i}^{T}G_{i,n}\left(\hat{\mathbf{x}}_{i}\right)=\begin{bmatrix}\dfrac{\partial^{2}G(\hat{\mathbf{x}}_i)}{\partial x_i\partial x_i} & \dfrac{\partial^{2}G(\hat{\mathbf{x}}_i)}{\partial x_i\partial y_i}\\ \dfrac{\partial^{2}G(\hat{\mathbf{x}}_i)}{\partial x_i\partial y_i} & \dfrac{\partial^{2}G(\hat{\mathbf{x}}_i)}{\partial y_i y_i}\end{bmatrix}$$

is Jacobian matrix with size 2 × 2 and the elements of matrix are

$$\begin{aligned}\frac{\partial^{2}G(\hat{\mathbf{x}})}{\partial x_p\partial x_p}=&\sum_{j=1}^{N}\frac{1}{\sigma_{p\bar{j}}^{2}}\left[\frac{\left(d_{p\bar{j}}-r_{p\bar{j}}\right)\left(x_p-\bar{x}_j\right)^{2}}{r_{p\bar{j}}^{3}}+\frac{\left(x_p-\bar{x}_j\right)^{2}}{r_{p\bar{j}}^{2}}-\frac{\left(d_{p\bar{j}}-r_{p\bar{j}}\right)}{r_{p\bar{j}}}\right]\\&+\sum_{\substack{i=1\\ i<p}}^{p-1}\frac{1}{\sigma_{i\hat{p}}^{2}}\left[+\frac{\left(d_{i\hat{p}}-r_{i\hat{p}}\right)\left(x_i-\hat{x}_p\right)^{2}}{r_{i\hat{p}}^{3}}+\frac{\left(x_i-\hat{x}_p\right)^{2}}{r_{i\hat{p}}^{2}}-\frac{\left(d_{i\hat{p}}-r_{i\hat{p}}\right)}{r_{i\hat{p}}}\right]\\&+\sum_{\substack{i=p+1\\ i>p}}^{M}\frac{1}{\sigma_{ip}^{2}}\left[+\frac{\left(d_{\hat{p}i}-r_{\hat{p}i}\right)\left(\hat{x}_p-x_i\right)^{2}}{r_{\hat{p}i}^{3}}+\frac{\left(\hat{x}_p-x_i\right)^{2}}{r_{\hat{p}i}^{2}}+\frac{\left(d_{\hat{p}i}-r_{\hat{p}i}\right)}{r_{\hat{p}i}}\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial^{2}G(\hat{\mathbf{x}})}{\partial x_p\partial y_p}=&\sum_{j=1}^{N}\frac{1}{\sigma_{p\bar{j}}^{2}}\left[\frac{\left(d_{p\bar{j}}-r_{p\bar{j}}\right)\left(x_p-\bar{x}_j\right)\left(y_p-\bar{y}_j\right)}{r_{p\bar{j}}^{3}}+\frac{\left(x_p-\bar{x}_j\right)\left(y_p-\bar{y}_j\right)}{r_{p\bar{j}}^{2}}\right]\\&+\sum_{\substack{i=1\\ i<p}}^{p-1}\frac{1}{\sigma_{ip}^{2}}\left[+\frac{\left(d_{i\hat{p}}-r_{i\hat{p}}\right)\left(x_i-\hat{x}_p\right)\left(y_i-\hat{y}_p\right)}{r_{i\hat{p}}^{3}}+\frac{\left(x_i-\hat{x}_p\right)\left(y_i-\hat{y}_p\right)}{r_{i\hat{p}}^{2}}\right]\\&+\sum_{\substack{i=p+1\\ i>p}}^{M}\frac{1}{\sigma_{pi}^{2}}\left[+\frac{\left(d_{\hat{p}i}-r_{\hat{p}i}\right)\left(\hat{x}_p-x_i\right)\left(\hat{y}_p-y_i\right)}{r_{\hat{p}i}^{3}}+\frac{\left(\hat{x}_p-x_i\right)\left(\hat{y}_p-y_i\right)}{r_{\hat{p}i}^{2}}\right]\end{aligned}$$

$$\frac{\partial^{2}G(\hat{\mathbf{x}})}{\partial y_p\partial y_p}=\left.\frac{\partial^{2}G(\hat{\mathbf{x}})}{\partial x_p\partial x_p}\right|_{x_p=y_p}$$

As before, the solution of matrix inverse $\left[\nabla_{\mathbf{x}_i}\nabla_{\mathbf{x}_i}^{T}G_{i,n}\left(\hat{\mathbf{x}}_{i,n+1}(k)\right)\right]_{2\times 2}^{-1}\ \nabla_{\mathbf{x}_i}^{T}G_{i,n}\left(\hat{\mathbf{x}}_{i,n+1}(k)\right)$ can be replaced Gaussian elimination method, $\nabla_{\mathbf{x}_i}\nabla_{\mathbf{x}_i}^{T}G_{i,n}\left(\hat{\mathbf{x}}_{i,n+1}(k)\right)\backslash\nabla_{\mathbf{x}_i}^{T}G_{i,n}\left(\hat{\mathbf{x}}_{i,n+1}(k)\right)$ by MATLAB function. Even if the Jacobian matrix will calculate a lot of summations, but the solution is only calculated

$2\times 2$ matrix Gaussian elimination method. It really reduces the large Jacobian matrix Gaussian elimination method (3.8) of joint Newton's method calculation complexity. However, joint Newton's algorithm has to count *M* times. Then, we assume one local iteration for one single mobile in (3.26) needs $F_{2\times 2}$ flops. *L* local iterations are needed for each mobile to converge in (3.24). Once global iteration for all *M* mobiles needs $F_{2\times 2}\times L\times M$ flops. It needs G global iterations to converge all the positions of mobile. Then, the total computation cost is given by

Divided Newton computation cost= $F_{2\times 2}\times L\times M\times G_{devided}$ flops. (3.27)

Contrast with joint Newton's computation cost (3.9) which is larger than the divided Newton computation cost in (3.27) when *M* is large. Computer simulations will demonstrate it in Section 5.3.

### 3.2.2b Taylor-Series Expansion

Actually, divided ML estimator in (3.24) is similar to ML estimator of traditional localization (2.11). In case of cooperative system, the matrix form of divided TS matrix for mobile *i* at $n^{th}$ global iteration time is

$$\mathbf{H}_{i\_DTS,n}\mathbf{x}_{i\_DTS,n+1}=\mathbf{b}_{i\_DTS,n}+\mathbf{n}_{i\_DTS,n} \quad (3.28)$$

where

$$\mathbf{H}_{i\_DTS,n}=\begin{bmatrix}\mathbf{H}_{i\_TS}\\ \mathbf{H}_{i\_VTS,n}\end{bmatrix}, \quad (3.29)$$

$$\mathbf{b}_{i\_DTS,n}=\begin{bmatrix}\mathbf{b}_{i\_TS}\\ \mathbf{b}_{i\_VTS,n}\end{bmatrix}, \quad (3.30)$$

and

$$\mathbf{n}_{i\_DTS,n}=\begin{bmatrix}\mathbf{n}_{i\_TS}\\ \mathbf{n}_{i\_VTS,n}\end{bmatrix}. \quad (3.31)$$

$\mathbf{H}_{i\_DTS,n}$ is a divided angle matrix and $\mathbf{n}_{i\_DTS,n}$ is a divided TS error vector. In case of uncooperative angle matrix $\mathbf{H}_{i\_TS}$ is like (2.17). Without loss of generality, let *i=1*, the virtual TS matrix for mobile *1* is

$$\mathbf{H}_{1\_VTS,n}=\begin{bmatrix} \dfrac{x_{10}-\hat{x}_{2,n}}{r_{10,\hat{2},n}} & \dfrac{y_{10}-\hat{y}_{2,n}}{r_{10,\hat{2},n}} \\ \dfrac{x_{10}-\hat{x}_{\hat{3},n}}{r_{10,\hat{3},n}} & \dfrac{y_{10}-\hat{y}_{3,n}}{r_{10,\hat{3},n}} \\ \vdots & \\ \dfrac{x_{10}-\hat{x}_{\hat{M},n}}{r_{10,\hat{M},n}} & \dfrac{y_{10}-\hat{y}_{\hat{M},n}}{r_{10,\hat{M},n}} \end{bmatrix}=\begin{bmatrix} \cos\theta_{10,\hat{2},n} & \sin\theta_{10,\hat{2},n} \\ \cos\theta_{10,\hat{3},n} & \sin\theta_{10,\hat{3},n} \\ \vdots & \\ \cos\theta_{10,\hat{M},n} & \sin\theta_{10,\hat{M},n} \end{bmatrix}=\begin{bmatrix} \mathbf{a}^T_{10,\hat{2},n} \\ \mathbf{a}^T_{10,\hat{3},n} \\ \vdots \\ \mathbf{a}^T_{10,\hat{3},n} \end{bmatrix}, \qquad (3.32)$$

$$\mathbf{b}_{1\_VTS,n}=\begin{bmatrix} d_{12}-\tilde{r}_{10,\hat{2},n} \\ d_{13}-\tilde{r}_{10,\hat{3},n} \\ \vdots \\ d_{1M}-\tilde{r}_{10,\hat{M},n} \end{bmatrix}, \qquad (3.33)$$

and

$$\mathbf{n}_{1\_VTS,n}=\begin{bmatrix} n_{12}+n_{T\_12}+\hat{e}_{2\_VTS,n} \\ n_{13}+n_{T\_13}+\hat{e}_{3\_VTS,n} \\ \vdots \\ n_{1M}+n_{T\_1M}+\hat{e}_{M\_VTS,n} \end{bmatrix} \qquad (3.34)$$

where $\mathbf{H}_{1\_VTS,n}$ is virtual angle matrix, $\mathbf{n}_{1\_VTS,n}$ is virtual TS error vector, $\left(\hat{x}_{j,n},\hat{y}_{j,n}\right)$ is position of virtual sensor *j* at $n^{\text{th}}$ global iteration, $r_{10,\hat{j},n}$ is the distance between reference point of mobile *1* and virtual sensor *j* location at $n^{\text{th}}$ global iteration, $\hat{e}_{j\_VTS,n}$ is the error of virtual sensor *j* location,

$$\mathbf{a}^T_{1\hat{j},n}=\begin{bmatrix} \dfrac{x_{10}-\hat{x}_{j,n}}{r_{10,\hat{j},n}} & \dfrac{y_{10}-\hat{y}_{j,n}}{r_{10,\hat{j},n}} \end{bmatrix} \qquad (3.35)$$

is a virtual angle vector and

$$\tilde{r}_{10,\hat{j}} = r_{10,\hat{j}} - \left(\frac{x_{10} - \hat{x}_j}{r_{10,\hat{j}}}\right) x_{10} - \left(\frac{y_{10} - \hat{y}_j}{r_{10,\hat{j}}}\right) y_{10}$$

In (3.28), we utilize positions of virtual sensors $\left(\hat{\mathbf{x}}_{2,n}, \hat{\mathbf{x}}_{3,n}, \cdots, \hat{\mathbf{x}}_{M,n},\right)$ at $n^{\text{th}}$ global to get the solution of $n+1^{\text{th}}$ for mobile *1*. As before, the divided TS estimator in (3.26) is given by

$$\hat{\mathbf{x}}_{1\_DWTS,n+1} = \left(\mathbf{H}_{1\_DTS,n}{}^{T}\mathbf{W}_{1\_DTS,n}\mathbf{H}_{1\_DTS,n}\right)^{-1}\mathbf{H}_{1\_DTS,n}{}^{T}\mathbf{W}_{1\_DTS,n}\mathbf{b}_{1\_DTS,n} \qquad (3.36)$$

Assume the reference point $(x_{10}, y_{10})$ is very close to true position $(x_1, y_1)$ and all variances of measurement noises are the same as $\sigma^2$, where

$$\mathbf{W}_{1\_DWTS,n} = \left(E\left[\mathbf{n}_{1\_DTS,n}\mathbf{n}_{1\_DTS,n}{}^{T}\right]\right)^{-1} = \begin{bmatrix} \mathbf{W}_{1\_TS} & 0 \\ 0 & \mathbf{W}_{1\_VTS,n} \end{bmatrix}, \qquad (3.37)$$

$\mathbf{W}_{1\_DTS,n}$ is divided TS weighting includes the $\mathbf{W}_{1\_TS}$ is uncooperative TS weighting in (2.22) and The element of virtual TS weighting $\mathbf{W}_{1\_VTS,n}$ which is gotten from the covariance inverse of virtual TS error vector, $E\left[\mathbf{n}_{1\_VTS,n}\mathbf{n}_{1\_VTS,n}{}^{T}\right]$ whose element is

On-diagonal: $E\left[\mathbf{n}_{1\_VTS,n}\mathbf{n}_{1\_VTS,n}{}^{T}\right]_{jj} = \sigma^2 + \hat{\sigma}^2_{j\_VTS,n}$

As before, if we ignore the statistics $\mathbf{W}_{1\_DTS,n}$, the solution in (3.36) can be further simplified as

$$\hat{\mathbf{x}}_{1\_DTS,n+1} = \left(\mathbf{H}_{1\_DTS,n}{}^{T}\mathbf{H}_{1\_DTS,n}\right)^{-1}\mathbf{H}_{1\_DTS,n}{}^{T}\mathbf{b}_{1\_DTS,n} \qquad (3.38)$$

The compensation of TS weighting $\left(\sigma^2 + \hat{\sigma}^2_{j\_VTS,n}\right)^{-1}$ is gotten by the error of virtual sensor *j*, $\hat{e}_{j\_VTS,n}$ which will be discussed clearly in Section 3.3

### 3.2.2c Distance-Augmented

The divided DA matrix form in divided ML (3.24) for mobile *i* at $n^{\text{th}}$ global

iteration can be denoted as follows

$$\mathbf{H}_{i\_DDA,n}\tilde{\mathbf{x}}_{i\_DDA,n+1} = \mathbf{b}_{i\_DDA,n} + \mathbf{n}_{i\_DDA,n} \qquad (3.39)$$

where

$$\mathbf{H}_{i\_DDA,n} = \begin{bmatrix} \mathbf{H}_{DA} \\ \mathbf{H}_{i\_VDA,n} \end{bmatrix}, \qquad (3.40)$$

$$\mathbf{b}_{i\_DDA,n} = \begin{bmatrix} \mathbf{b}_{i\_DA} \\ \mathbf{b}_{i\_VDA,n} \end{bmatrix}, \qquad (3.41)$$

and

$$\mathbf{n}_{i\_DDA,n} = \begin{bmatrix} \mathbf{n}_{i\_DA} \\ \mathbf{n}_{i\_VDA,n} \end{bmatrix} \qquad (3.42)$$

$\mathbf{H}_{i\_DDA,n}$ is divided coordinate matrix, $\mathbf{n}_{i\_DDA,n}$ is divided DA error vector and uncooperative coordinate matrix $\mathbf{H}_{DA}$ is like (2.38). Without loss of generality, let *i=1*, the virtual DA matrix for mobile *1* is

$$\mathbf{H}_{1\_VDA,n} = \begin{bmatrix} 2\hat{x}_{2,n} & 2\hat{y}_{2,n} & -1 \\ 2\hat{x}_{3,n} & 2\hat{y}_{3,n} & -1 \\ \vdots & \vdots & \vdots \\ 2\hat{x}_{M,n} & 2\hat{y}_{M,n} & -1 \end{bmatrix},$$

$$\tilde{\mathbf{x}}_{i\_DDA,n+1} = \begin{bmatrix} x_i \\ y_i \\ R_i \end{bmatrix}_{n+1},$$

$$\mathbf{b}_{1\_VDA,n} = \begin{bmatrix} \hat{g}_{2,n} - d_{12}^2 \\ \hat{g}_{3,n} - d_{13}^2 \\ \vdots \\ \hat{g}_{M,n} - d_{1M}^2 \end{bmatrix},$$

and

$$\mathbf{n}_{1\_VDA,n}=\begin{bmatrix} 2r_{12}\left(n_{12}+\hat{e}_{2,n}\right)+\left(n_{12}+\hat{e}_{2,n}\right)^2 \\ 2r_{13}\left(n_{13}+\hat{e}_{3,n}\right)+\left(n_{13}+\hat{e}_{3,n}\right)^2 \\ \vdots \\ 2r_{1M}\left(n_{1M}+\hat{e}_{M,n}\right)+\left(n_{1M}+\hat{e}_{M,n}\right)^2 \end{bmatrix}$$

where $\mathbf{H}_{1\_VDA,n}$ is virtual coordinate matrix, $\mathbf{n}_{1\_VDA,n}$ is virtual DA error vector and $\hat{g}_{j,n}=\hat{x}_{j,n}^2+\hat{y}_{j,n}^2$.

As before, the divided DA estimator is given by

$$\hat{\hat{\mathbf{x}}}_{1\_DDA,n+1}=\left(\mathbf{H}_{1\_DDA,n}{}^T\mathbf{W}_{1\_DDA,n}\mathbf{H}_{1\_DDA,n}\right)^{-1}\mathbf{H}_{1\_DDA,n}{}^T\mathbf{W}_{1\_DDA,n}\mathbf{b}_{1\_DDA,n} \qquad (3.43)$$

Assume all variances of measurement noises are the same as $\sigma^2$,

where

$$\mathbf{W}_{1\_DDA,n}=\left(E\left[\mathbf{n}_{1\_DDA,n}\mathbf{n}_{1\_DDA,n}{}^T\right]\right)^{-1}=\begin{bmatrix}\mathbf{W}_{1\_DA} & 0\\ 0 & \mathbf{W}_{1\_VDA,n}\end{bmatrix}, \qquad (3.44)$$

the $\mathbf{W}_{1\_DA}$ is uncooperative DA weighting matrix in (2.45) and the element of $\mathbf{W}_{1\_VDA,n}$ is gotten from the covariance inverse of virtual DA error vector, $E\left[\mathbf{n}_{1\_VDA,n}\mathbf{n}_{1\_VDA,n}{}^T\right]$ whose element is

On-diagonal: $E\left[\mathbf{n}_{1\_DA}\mathbf{n}_{1\_DA}{}^T\right]_{jj}=4r_{1\hat{j}}^2\left(\sigma^2+\hat{\sigma}_{j,n}^2\right)+3\left(\sigma^2+\hat{\sigma}_{j,n}^2\right)$ (3.45)

### 3.2.2d Hyperbolic-Canceled

The divided HC matrix equation for mobile *i* at $n^{th}$ global iteration can be denoted as follows

$$\mathbf{H}_{i\_DHC,n}\mathbf{x}_{i\_DCH,n+1}=\mathbf{b}_{i\_DHC,n}+\mathbf{n}_{i\_DHC,n} \qquad (3.46)$$

where

$$\mathbf{H}_{i\_DHC,n} = \begin{bmatrix} \mathbf{H}_{HC} \\ \mathbf{H}_{i\_VHC,n} \end{bmatrix}, \qquad (3.47)$$

$$\mathbf{b}_{i\_DHC,n} = \begin{bmatrix} \mathbf{b}_{HC} \\ \mathbf{b}_{i\_VHC,n} \end{bmatrix}, \qquad (3.48)$$

$$\mathbf{n}_{i\_DHC,n} = \begin{bmatrix} \mathbf{n}_{HC} \\ \mathbf{n}_{i\_VHC,n} \end{bmatrix} \qquad (3.49)$$

where $\mathbf{H}_{i\_DHC,n}$ is divided coordinate-difference matrix, $\mathbf{n}_{i\_DHC,n}$ is divided HC error vector, uncooperative coordinate-difference matrix $\mathbf{H}_{HC}$ is like (2.45). Without loss of generality, let *i=1*, and sensor *1* is reference sensor, then the virtual HC matrix for mobile *1* is,

$$\mathbf{H}_{1\_VHC,n} = \begin{bmatrix} (\hat{x}_2 - \bar{x}_1) & (\hat{y}_2 - \bar{y}_1) \\ (\hat{x}_3 - \bar{x}_1) & (\hat{y}_3 - \bar{y}_1) \\ \vdots & \vdots \\ (\hat{x}_M - \bar{x}_1) & (\hat{y}_M - \bar{y}_1) \end{bmatrix},$$

$$\mathbf{b}_{1\_VHC,n} = \frac{1}{2}\begin{bmatrix} d_{1\bar{1}}^2 - d_{12}^2 + \hat{g}_2 - \bar{g}_1 \\ d_{1\bar{1}}^2 - d_{13}^2 + \hat{g}_3 - \bar{g}_1 \\ \vdots \\ d_{1\bar{1}}^2 - d_{1M}^2 + \hat{g}_M - \bar{g}_1 \end{bmatrix},$$

and

$$\mathbf{n}_{1\_VHC,n} = \begin{bmatrix} r_{12}\left(n_{12} + \hat{e}_{2,n}\right) - r_{1\bar{1}}n_{1\bar{1}} + \frac{1}{2}\left(\left(n_{12} + \hat{e}_{2,n}\right)^2 - n_{1\bar{1}}^2\right) \\ r_{13}\left(n_{13} + \hat{e}_{3,n}\right) - r_{1\bar{1}}n_{1\bar{1}} + \frac{1}{2}\left(\left(n_{13} + \hat{e}_{3,n}\right)^2 - n_{1\bar{1}}^2\right) \\ \vdots \\ r_{1M}\left(n_{1M} + \hat{e}_{M,n}\right) - r_{1\bar{1}}n_{1\bar{1}} + \frac{1}{2}\left(\left(n_{1M} + \hat{e}_{M,n}\right)^2 - n_{1\bar{1}}^2\right) \end{bmatrix}$$

where $\mathbf{H}_{1\_VHC,n}$ is virtual coordinate-difference matrix and $\mathbf{n}_{1\_VHC,n}$ is virtual HC error vector. As before, the divided HC estimator is given by

$$\hat{\mathbf{x}}_{1\_DHC,n+1} = \left(\mathbf{H}_{1\_DHC,n}{}^{T}\mathbf{W}_{1\_DHC,n}\mathbf{H}_{1\_DHC,n}\right)^{-1}\mathbf{H}_{1\_DHC,n}{}^{T}\mathbf{W}_{1\_DHC,n}\mathbf{b}_{1\_DHC,n} \qquad (3.50)$$

where

$$\mathbf{W}_{1\_DHC,n} = \left(E\left[\mathbf{n}_{1\_DHC,n}\mathbf{n}_{1\_DHC,n}{}^{T}\right]\right)^{-1}, \qquad (3.51)$$

$\mathbf{W}_{1\_DHC,n}$ is divided HC weighting matrix. Then $E\left[\mathbf{n}_{1\_VHC,n}\mathbf{n}_{1\_VHC,n}{}^{T}\right]$ whose element is

On-diagonal: $E\left[\mathbf{n}_{1\_VHC,n}\mathbf{n}_{1\_VHC,n}{}^{T}\right]_{jj} = r_{1\bar{j}}^2\left(\sigma^2 + \hat{\sigma}_{j,n}^2\right) + r_{1\bar{1}}^2\sigma_{1\bar{1}}^2 + \frac{3}{4}(\sigma^2 + \hat{\sigma}_{j,n}^2 - \sigma_{1\bar{1}}^2)$

Off-diagonal: $E\left[\mathbf{n}_{1\_VHC,n}\mathbf{n}_{1\_VHC,n}{}^{T}\right]_{jk} = r_{1\bar{1}}^2\sigma^2 + \frac{3}{4}\sigma^2$

Because $E\left[\mathbf{n}_{1\_VHC,n}\mathbf{n}_{1\_VHC,n}{}^{T}\right]$ is not a diagonal matrix, the divided HC weighting matrix will becomes more complex.

## 3.3 Compensation of Uncertain Virtual Sensor

Actually, if there is a uncertain sensor which will help to estimate mobile locaiton, its variance of uncertain location error had been derived [43] as follow,

The cooperative measurement TOA model between mobile *i* and mobile *j* is given by,

$$d_{ij} = \| \mathbf{x}_i - \mathbf{x}_j \|_2 + n_{ij}, \qquad (3.52)$$

Now, we have a uncertain virtual sensor *j* location, $\hat{\mathbf{x}}_j$ which is surround noise by $\mathbf{x}_j$, then (3.31) can be written as follow

$$d_{ij} = \| \left(\mathbf{x}_i - \hat{\mathbf{x}}_j\right) - \hat{\mathbf{e}}_j \|_2 + n_{ij} \qquad (3.53)$$

where $\hat{\mathbf{x}}_j + \hat{\mathbf{e}}_j = \mathbf{x}_j$ and $\hat{\mathbf{e}}_j = \left[\hat{e}_{jx} \quad \hat{e}_{jy}\right]^T$ are x-plane and y-plane error of virtual coordinate. Here, the coordinate error $\hat{\mathbf{e}}_j$ dependents on the traditional localization algorithm in Section 2.2. For example,

$$\sigma^2{}_{j\_WTS} = trace\left[\operatorname{cov}\left(\mathbf{e}_{j\_WTS}\right)\right] = trace\left[\operatorname{cov}\left(\hat{\mathbf{e}}_j\right)\right]$$

in (2.26) if the uncooperative TS algorithm is used. In (3.53), using virtual sensor $\hat{\mathbf{x}}_j$ instead of true sensor $\mathbf{x}_j$, the virtual coordinate error will involve inside of norm, then we want it get out from the norm. Therefore, apply Taylor-series expansion to (3.53) as follows

$$d_{ij} = \| \mathbf{x}_i - \hat{\mathbf{x}}_j \|_2 - \nabla^T_{\hat{\mathbf{e}}_j} \hat{\mathbf{e}}_j + \text{higher order terms} + n_{ij} \qquad (3.54)$$

The uncertain error is out of norm in (3.54). Assume the initial is very closed the true position, the higher order terms can be omitted, (3.54) can be written as follows

$$d_{ij} \approx \| \mathbf{x}_i - \hat{\mathbf{x}}_j \|_2 - \nabla^T_{\hat{\mathbf{e}}_j} \hat{\mathbf{e}}_j + n_{ij} \qquad (3.55)$$

The total error including measurement noise $n_{ij}$ and virtual sensor error $\nabla^T_{\hat{\mathbf{e}}_j} \hat{\mathbf{e}}_j$. Therefore, the variance of total error is

$$\sigma^2_{i\hat{j}} = E\left[\left(-\nabla^T_{\hat{\mathbf{e}}_j} \hat{\mathbf{e}}_j + n_{ij}\right)^2\right] \qquad (3.56)$$

We assume the measurement noise and virtual sensor error are independent, and the x-plane and y-plane of virtual sensor location error are i,i,d. Then, (3.56) can be written as

$$\sigma^2_{i\hat{j}} = \hat{\sigma}^2_j + \sigma^2_{ij} \qquad (3.57)$$

We have the error variance of virtual sensor $j$ location , $\sigma^2_{\hat{ij}}$ in (3.57). Previous research has derived (3.57) by considering an uncertain sensor. In the divide-and-conquer method, we can apply these virtual weightings compensation $\mathbf{W}_{i\_VTS,n}$, $\mathbf{W}_{i\_VDA,n}$ and $\mathbf{W}_{i\_VHC,n}$ to all virtual sensors to enhance localization accuracy by global iteration. Then, the error variance of virtual sensor $j, \sigma^2_{\hat{ij},n}$ will become smaller and smaller after global iteration. On the other hand, the

position of virtual sensor *j* becomes more accuracy. The computer simulation will improvement the compensation in Section 5.4.2. [15] proposes an error propagation aware algorithm to track the extent of the uncertain virtual position error by compensation in (3.57), but this algorithm doesn't updated it with global iteration. Therefore, in our divided method, we update the compensation by iterating estimation.

# Chapter 4
# Theoretical Analysis of Mean Square Error

Recently, the determination of position accuracy for geolocation is a fundamental issue in wireless sensor networks. The Cramer-Rao bound (CRLB) provides a useful means for the analysis of the limits of localization accuracy. Based on different types of measurements, the CRLB result for cooperative localization can be found in [3] . However, the CRLB for TOA-based cooperative localization is gotten from *F*isher *I*nformation *M*atrix (FIM) [3, 44]. But the matrix becomes more complexity when the number of mobiles increases. Therefore, the eigenvalue view of EFIM is proposed in [44]. Hence, we use the eigenvalue view to propose recursive block matrix inversion to derive a simple approximated EFIM (AEFIM). Then, we can see the more mobiles, the better MSE performance. Next, we further simplify the matrix to get the approximation of cooperative CRLB (AC-CRLB). Besides, we try to analyze the theoretical converged MSE of divided algorithms, but it only success in divided TS algorithm in special case of two mobiles. The rest of the section is organized as follows. In Section 4.1, we present the approximated cooperative CRLB. The theoretical converged MSE for divided algorithms are shown in Section 4.2.

## 4.1 Approximated Cooperative Cramer-Rao Bound

First, we will present the uncooperative CRLB in Section 4.1.1. In Section 4.1.2, the cooperative CRLB will be illustrated. Next, an eigenvalue view [44] on cooperative EFIM for two mobiles is shown in Section 4.1.3. Then, we use the

character to derive a approximated EFIM in Section 4.1.4. Finally, the approximation of cooperative CRLB is derived in Section 4.1.5.

### 4.1.1 Basic Localization CRLB

In Basic localization system (Figure 2.2), we estimated mobile *i* through its uncooperative measurement. Then let $\hat{\mathbf{x}}_i$ denote an estimate of mobile $i, \mathbf{x}_i$ and measurement set $\mathbf{d}_i$ :

$$\mathbf{d}_i = \left[ d_{i\bar{1}}, d_{i\bar{2}} \cdots d_{i\bar{N}} \right] \qquad (4.1)$$

The error covariance matrix of $\hat{\mathbf{e}}_i = \hat{\mathbf{x}}_i - \mathbf{x}_i$ satisfies Information Inequality [25]

$$\mathrm{cov}\left(\hat{\mathbf{e}}_i\right) = E\left[\hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T\right] \geq \mathbf{J}_{\mathbf{x}_i}^{-1} \qquad (4.2)$$

where $\mathbf{J}_{\mathbf{x}_i}$ is the full uncooperative *F*isher *I*nformation *M*atrix (FIM) [25] for mobile $i, \mathbf{x}_i$

$$\mathbf{J}_{\mathbf{x}_i} \triangleq E\left\{ \left[ \frac{\partial}{\partial \mathbf{x}_i} \ln f\left(\mathbf{d}_i \mid \mathbf{x}_i\right) \right] \left[ \frac{\partial}{\partial \mathbf{x}_i} \ln f\left(\mathbf{d}_i \mid \mathbf{x}_i\right) \right]^T \right\} \qquad (4.3)$$

where $f\left(\mathbf{d}_i \mid \mathbf{x}_i\right)$ is the likelihood function in (2.9). Expanding (4.3) to get

$$\mathbf{J}_{\mathbf{x}_i} = \sum_{j=1}^{N} \frac{1}{\sigma_{i\bar{j}}^2} \begin{bmatrix} \dfrac{(x_i - \bar{x}_j)^2}{r_{i\bar{j}}^2} & \dfrac{(x_i - \bar{x}_j)(y_i - \bar{y}_j)}{r_{i\bar{j}}^2} \\ \dfrac{(x_i - \bar{x}_j)(y_i - \bar{y}_j)}{r_{i\bar{j}}^2} & \dfrac{(y_i - \bar{y}_j)^2}{r_{i\bar{j}}^2} \end{bmatrix}$$

$$= \sum_{j=1}^{N} \frac{1}{\sigma_{i\bar{j}}^2} \begin{bmatrix} \cos^2 \theta_{i\bar{j}} & \cos\theta_{i\bar{j}} \sin\theta_{i\bar{j}} \\ \cos\theta_{i\bar{j}} \sin\theta_{i\bar{j}} & \sin^2 \theta_{i\bar{j}} \end{bmatrix} \qquad (4.4)$$

trace of inverse of $\mathbf{J}_{\mathbf{x}_i}$ in (4.4) is defined as the lower bound for MSE. Therefore, Basic localization CRLB is given by

$$\mathrm{CRLB}_{\mathrm{Uncooperative}} = \mathrm{trace}\left[\mathbf{J}_{\mathbf{x}_i}^{-1}\right] \qquad (4.5)$$

The size of full uncooperative FIM is 2×2 for mobile *i*. Then, we have multiple uncooperative CRLB from mobile *1* to mobile *M* in (4.5). However, in cooperative system, the size of cooperative full FIM is complex. Therefore, we derive a simple *A*pproximated *E*quivalent-FIM(AEFIM) and further demonstrate approximation of cooperative CRLB (AC-CRLB) in following section.

### 4.1.2 Cooperative Localization CRLB

As before, the full cooperative FIM is given by cooperative likelihood function (3.3). We rewrite the cooperative likelihood function as follows

$$p\left(\mathbf{d}_{Uncoop},\mathbf{d}_{Coop}\,|\,\mathbf{x}\right)= \underbrace{\prod_{i=1}^{M}\prod_{j=1}^{N}\frac{1}{\sqrt{2\sigma_{i\bar{j}}^{2}}}\exp\left(-\frac{\left(d_{i\bar{j}}-\left\|\mathbf{x}_i-\bar{\mathbf{x}}_j\right\|\right)^2}{2\sigma_{i\bar{j}}^{2}}\right)}_{\text{Uncooperation}}\cdot\underbrace{\prod_{\substack{i=1\\ j>i}}^{M}\frac{1}{\sqrt{2\sigma_{ij}^{2}}}\exp\left(-\frac{\left(d_{ij}-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|\right)^2}{2\sigma_{ij}^{2}}\right)}_{\text{Cooperation}} \tag{3.3}$$

According to (4.3), the full cooperative FIX [3, 25] can be written as follows

$$\mathbf{J}_{\text{Coop},\mathbf{x}(M)}=\begin{bmatrix} \mathbf{J}_{\mathbf{x}_1}+\sum_{\substack{i=1\\ i\neq 1}}^{M}\mathbf{C}_{1i} & -\mathbf{C}_{12} & \cdots & -\mathbf{C}_{1M} \\ -\mathbf{C}_{12} & \mathbf{J}_{\mathbf{x}_2}+\sum_{\substack{i=1\\ i\neq 2}}^{M}\mathbf{C}_{2i} & \cdots & -\mathbf{C}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{C}_{1M} & -\mathbf{C}_{2M} & \cdots & \mathbf{J}_{\mathbf{x}_M}+\sum_{\substack{i=1\\ i\neq M}}^{M}\mathbf{C}_{Mi} \end{bmatrix}_{2M\times 2M} \tag{4.6}$$

where $\mathbf{J}_{\mathbf{x}_i}$ is the full uncooperative FIM in (4.4) for mobile *i* and $\mathbf{C}_{ij}$ is cooperative information matrix between mobile *i* and mobile *j* which is denoted as

$$\mathbf{C}_{ij}=\frac{1}{\sigma_{ij}^{2}}\mathbf{Q}_{\theta_{ij}}\mathbf{Q}_{\theta_{ij}}^{T}=\frac{1}{\sigma_{ij}^{2}}\begin{bmatrix}\cos^2\theta_{ij} & \cos\theta_{ij}\sin\theta_{ij}\\ \cos\theta_{ij}\sin\theta_{ij} & \sin^2\theta_{ij}\end{bmatrix} \tag{4.7}$$

Then, the cooperative CRLB for mobile *i* is

$$\text{CRLB}_{\text{mobile 1}}=\text{trace}\left[\text{upper-left } 2\times 2 \text{ submatrix of } \mathbf{J}^{-1}{}_{\text{Coop},\mathbf{x}(M)}\right] \tag{4.8}$$

Obviously, the size of full cooperative Fisher information matrix (4.6) becomes larger and complexity when the numbers of mobiles increases. It is difficult to obtain cooperative CRLB in (4.8). We will derive a simple AEFIM. Therefore, we can get cooperative easily. Next, we introduce the eigenvalue view on full cooperative FIM for two mobiles in following section.

### 4.1.3 Cooperative Fisher Information for Two Mobiles based on Eigen Decomposition

First, we briefly cite the definition of *E*quivalent *F*isher *I*nformation *M*atrix (EFIM) [44]. Let the original FIM and its inverse be

$$\mathbf{J}_{\text{Coop},\mathbf{x}(M)} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}, \text{ and } \mathbf{J}^{-1}_{\text{Coop},\mathbf{x}(M)} = \begin{bmatrix} \mathbf{T}_{2\times 2} & \mathbf{K} \\ \mathbf{K} & \mathbf{L} \end{bmatrix}_{2M\times 2M}, \quad (4.9)$$

The cooperative CRLB for mobile 1 is given by

$$\text{CRLB}_{\text{mobile 1}} = \text{trace}\left(\mathbf{T}_{2\times 2}\right) \quad (4.10)$$

Let $\mathbf{J}_{\mathbf{e}_1}^{-1}=\mathbf{T}_{2\times 2}$, hence $\mathbf{J}_e$ is an EFIM of $\mathbf{J}_{\text{Coop},\mathbf{x}(M)}$ for mobile *1*. Now, we use eigenvalue decomposition [44] to analysis the CRLB of cooperative localization. We summarize the sight as follows

For *M=2*, the structure of full cooperative FIM is

$$\mathbf{J}_{\text{Coop},\mathbf{x}(2)} = \left[\begin{array}{c|c} \mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2} & -\mathbf{C}_{1,2} \\ \hline -\mathbf{C}_{1,2} & \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2} \end{array}\right]_{4\times 4} \quad (4.11)$$

Eigen-value decomposition on full uncooperative FIM $\mathbf{J}_{\mathbf{x}_1}$, $\mathbf{J}_{\mathbf{x}_2}$ and cooperative information matrix $\mathbf{C}_{1,2}$ from mobile *1* and mobile *2* as follows

$$\mathbf{J}_{\mathbf{x}_i} \triangleq \mathbf{U}_{\beta_i}^T \begin{bmatrix} \lambda_i & 0 \\ 0 & \mu_i \end{bmatrix} \mathbf{U}_{\beta_i}, \; i=1,2 \quad (4.12)$$

where $\lambda_i$ and $\mu_i$ are the eigenvalues of $\mathbf{J}_{\mathbf{x}_i}$, and $\mathbf{U}_{\beta_i} = \begin{bmatrix} \cos\beta_i & \sin\beta_i \\ -\sin\beta_i & \cos\beta_i \end{bmatrix}$ is rotation matrix of an angle $\beta_i$, and the cooperative information matrix can be written as

$$\mathbf{C}_{i,j} \triangleq \mathbf{U}_{\theta_{i,j}}^{T} \begin{bmatrix} v_{i,j} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{U}_{\theta_{i,j}}, \; i=1, j=2 \qquad (4.13)$$

where we define $v_{i,j}$ as the strength of the cooperative information between mobile *i* and mobile *j*. From (4.4) and (4.12), we can know that

$$\lambda_i + \mu_i = \sum_{j=1}^{N} \frac{1}{\sigma_{ij}^{\;2}} \text{ and } v_{i,j} = \frac{1}{\sigma_{ij}^{\;2}} \qquad (4.14)$$

Actually, because uncooperative FIM is related with the angles from sensor, therefore the eigenvalues is also related with the angle from sensors to mobile. The feature will show in Section 4.1.4. We get cooperative CRLB from the full cooperative FIM inverse. By the following matrix inverse lemma on (4.11) [24]

$$\mathbf{J}_{\text{Coop},\mathbf{x}(2)}^{-1} = \begin{bmatrix} \mathbf{J}_{\mathbf{e}_1}^{-1} & -\mathbf{J}_{\mathbf{e}_1}^{-1}\mathbf{C}_{1,2}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}\right)^{-1} \\ -\mathbf{J}_{\mathbf{e}_2}^{-1}\mathbf{C}_{1,2}\left(\mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}\right)^{-1} & \mathbf{J}_{\mathbf{e}_2}^{-1} \end{bmatrix}_{4\times 4} \qquad (4.15)$$

We obtain the EFIM for mobile *1*

$$\mathbf{J}_{\mathbf{e}_1} = \mathbf{J}_{\mathbf{x}_1} + \mathbf{C}_{1,2} - \mathbf{C}_{1,2}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}\right)^{-1}\mathbf{C}_{1,2} \qquad (4.16)$$

After applying eigenvalue decomposition, (4.17) and (4.18) can be shown [44]

$$\mathbf{J}_{e_1} = \mathbf{J}_{\mathbf{x}_1} + \chi_{1,2}\mathbf{C}_{1,2} \qquad (4.17)$$

where the uncertain weighting of mobile *2* to mobile *1* is given by

$$\chi_{1,2} = \frac{1}{1+v_{1,2}\Delta_{\theta_{1,2},\beta_2}} \qquad (4.18)$$

and

$$\Delta_{\theta_{1,2},\beta_2} = \frac{1}{\lambda_2}\cos^2\left(\beta_2-\theta_{1,2}\right) + \frac{1}{\mu_2}\sin^2\left(\beta_2-\theta_{1,2}\right). \qquad (4.19)$$

(4.17) means the EFIM of mobile *1* includes its own full uncooperative FIM, $\mathbf{J}_{\mathbf{x}_1}$ and cooperative information matrix $\mathbf{C}_{1,2}$ from mobile *2*. Because the exact location of mobile *2* is uncertain, $\mathbf{C}_{1,2}$ is de-weighted by the $\chi_{1,2}$ <1. We can see the uncertain cooperative part as follows

$$\chi_{1,2}\mathbf{C}_{1,2} = \left(\chi_{1,2}\cdot v_{1,2}\right)\mathbf{Q}_{\theta_{ij}}\mathbf{Q}_{\theta_{ij}}^{T} = \frac{v_{1,2}}{1+v_{1,2}\Delta_{\theta_{1,2},\beta_2}}\mathbf{Q}_{\theta_{ij}}\mathbf{Q}_{\theta_{ij}}^{T} \qquad (4.20)$$

If bigger $v_{1,2}$, we have better cooperative localization CRLB. The EFIM for mobile *2* can be obtained similarly. Therefore, we can get simple cooperative CRLB from (4.16)

$$\text{CRLB}_{\text{mobile 1}} = \text{trace}\left[\mathbf{J}_{\mathbf{e}_1}^{-1}\right]_{2\times 2} \qquad (4.21)$$

Now, we make a lemma to derive the approximate EFIM. First, the relationship of (4.16) can be written as follow as

$$\mathbf{J}_{\text{Coop},\mathbf{x}(2)}^{-1} = \begin{bmatrix} \left(\mathbf{J}_{\mathbf{x}_1}+\chi_{1,2}\mathbf{C}_{1,2}\right)^{-1} & -\left(\mathbf{J}_{\mathbf{x}_1}+\chi_{1,2}\mathbf{C}_{1,2}\right)^{-1}\mathbf{C}_{1,2}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}\right)^{-1} \\ -\left(\mathbf{J}_{\mathbf{x}_2}+\chi_{2,1}\mathbf{C}_{1,2}\right)^{-1}\mathbf{C}_{1,2}\left(\mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}\right)^{-1} & \left(\mathbf{J}_{\mathbf{x}_2}+\chi_{2,1}\mathbf{C}_{1,2}\right)^{-1} \end{bmatrix} \qquad (4.22)$$

From (4.16) to (4.17), we propose a block matrix inversion lemma as follows,

First, we assume $\mathbf{J}_i$ is large. Therefore, the **Block matrix inversion lemma 1**:

$$\begin{bmatrix} \mathbf{J}_A+\mathbf{C} & -\mathbf{C} \\ -\mathbf{C} & \mathbf{J}_B+\mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} \left(\mathbf{J}_A+\mathbf{C}-\mathbf{C}\left(\mathbf{J}_B+\mathbf{C}\right)^{-1}\mathbf{C}\right)^{-1} & \mathbf{J}_{e,A}^{-1}\mathbf{C}\left(\mathbf{J}_B+\mathbf{C}\right)^{-1} \\ \mathbf{J}_{e,B}^{-1}\mathbf{C}\left(\mathbf{J}_A+\mathbf{C}\right)^{-1} & \left(\mathbf{J}_B+\mathbf{C}-\mathbf{C}\left(\mathbf{J}_A+\mathbf{C}\right)^{-1}\mathbf{C}\right)^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} \left(\mathbf{J}_A+\chi_{B,C}\mathbf{C}\right)^{-1} & \left(\mathbf{J}_A+\chi_{B,C}\mathbf{C}\right)^{-1}\mathbf{C}\left(\mathbf{J}_B+\mathbf{C}\right)^{-1} \\ \left(\mathbf{J}_B+\chi_{A,C}\mathbf{C}\right)^{-1}\mathbf{C}\left(\mathbf{J}_A+\mathbf{C}\right)^{-1} & \left(\mathbf{J}_B+\chi_{A,C}\mathbf{C}\right)^{-1} \end{bmatrix}$$

$$\approx \begin{bmatrix} \left(\mathbf{J}_A+\chi_{B,C}\mathbf{C}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \left(\mathbf{J}_B+\chi_{A,C}\mathbf{C}\right)^{-1} \end{bmatrix} \qquad (4.23)$$

Because $\mathbf{J}_i$ is large, cause off-diagonal block matrixes to approach zero,

$\left(\mathbf{J}_A+\chi_{B,C}\mathbf{C}\right)^{-1}\mathbf{C}\left(\mathbf{J}_B+\mathbf{C}\right)^{-1}\approx 0$ and $\left(\mathbf{J}_B+\chi_{A,C}\mathbf{C}\right)^{-1}\mathbf{C}\left(\mathbf{J}_A+\mathbf{C}\right)^{-1}\approx 0$. For example, assume $N=20,\ \sigma^2=3,\ \beta_1=30°,\ \lambda_1=\mu_1=N/2\sigma^2$, we have uncooperative full FIM of mobile *2*

$$\mathbf{J}_{\mathbf{x}_1}=\begin{bmatrix}16.6667 & 0\\ 0 & 16.6667\end{bmatrix}, \qquad (4.24)$$

$\beta_2=45°,\ \lambda_2:\mu_2=3:1,$ we have uncooperative full FIM of mobile *2*

$$\mathbf{J}_{\mathbf{x}_2}=\begin{bmatrix}16.6667 & 8.3333\\ 8.3333 & 16.6667\end{bmatrix},$$

$\theta_{12}=90°$, we have cooperative information matrix

$$\mathbf{C}_{1,2}=\begin{bmatrix}0 & 0\\ 0 & 0.3333\end{bmatrix} \qquad (4.25)$$

and $\chi_{1,2}=0.9615$. Then the on-diagonal block matrix is

$$\left(\mathbf{J}_{\mathbf{x}_1}+\chi_{1,2}\mathbf{C}_{1,2}\right)^{-1}=\begin{bmatrix}0.6 & 0\\ 0 & 0.0589\end{bmatrix} \qquad (4.26)$$

and the off-diagonal block matrix is

$$\left(\mathbf{J}_{\mathbf{x}_1}+\chi_{1,2}\mathbf{C}_{1,2}\right)^{-1}\mathbf{C}_{1,2}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}\right)^{-1}=\begin{bmatrix}0 & 0\\ -0.0008 & 0.0015\end{bmatrix} \qquad (4.27)$$

Compare (4.26) and (4.27), the off-diagonal block matrix is relatively smaller than on-diagonal block matrix. Then, we will use lemma 1 to simplify the EFIM in following section.

### 4.1.4 Approximation of Equivalent-FIM (AEFIM) Based on Recursive Block Matrix Inversion

Now we will extend from two cooperative mobiles in Section 4.1.2 to multiple mobiles. Exact close form is difficult to obtain. By lemma 1 and eigenvalue view, AEFIM formula is derived for full cooperative FIM.

## Case a: Three Mobiles

The full cooperative FIM for three mobiles is given by

$$\mathbf{J}_{Coop,\mathbf{x}(3)} = \left[\begin{array}{c|cc} \mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}+\mathbf{C}_{1,3} & -\mathbf{C}_{1,2} & -\mathbf{C}_{1,3} \\ \hline -\mathbf{C}_{1,2} & \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3} & -\mathbf{C}_{2,3} \\ -\mathbf{C}_{1,3} & -\mathbf{C}_{2,3} & \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3} \end{array}\right]_{6\times 6} \quad (4.28)$$

We note that mobile *3* has provided additional information matrix $\mathbf{C}_{1,3}$ and $\mathbf{C}_{2,3}$.

We use matrix inverse lemma in (4.22) by partitioning into two groups of mobile *1* and mobiles *2,3*. The full cooperative FIM inverse is given by

$$\mathbf{J}^{-1}_{Coop,\mathbf{x}(3)} = \left[\begin{array}{c|c} \mathbf{J}^{-1}_{e_1} & \mathbf{K}_{2\times 4} \\ \hline \mathbf{K}^T & \mathbf{L}_{4\times 4} \end{array}\right]_{6\times 6} \quad (4.29)$$

Then, the EFIM of mobile *1* is given by

$$\mathbf{J}_{e_1} = \left(\mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}+\mathbf{C}_{1,3}\right) - \left[\mathbf{C}_{1,2} \quad \mathbf{C}_{1,3}\right] \left[\begin{array}{cc} \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3} & -\mathbf{C}_{2,3} \\ -\mathbf{C}_{2,3} & \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3} \end{array}\right]^{-1} \left[\begin{array}{c} \mathbf{C}_{1,2} \\ \mathbf{C}_{1,3} \end{array}\right] \quad (4.30)$$

Obviously, it becomes more difficult because of two cooperative block matrix

$$\left[\begin{array}{cc} \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3} & -\mathbf{C}_{2,3} \\ -\mathbf{C}_{2,3} & \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3} \end{array}\right]^{-1} \quad (4.31)$$

We call (4.31) two block matrix. Now, we apply the matrix inverse lemma *1* again to (4.31),

$$\left[\begin{array}{cc} \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3} & -\mathbf{C}_{2,3} \\ -\mathbf{C}_{2,3} & \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3} \end{array}\right]^{-1}$$

$$= \left[\begin{array}{cc} \left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\,\mathbf{C}_{2,3}\right)^{-1} & -\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}\right)^{-1}\mathbf{C}_{2,3}\left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{2,3}\right)^{-1} \\ -\left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,2}+\tilde{\chi}_{3,2}\mathbf{C}_{2,3}\right)^{-1}\mathbf{C}_{2,3}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}\right)^{-1} & \left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,2}+\tilde{\chi}_{3,2}\,\mathbf{C}_{2,3}\right)^{-1} \end{array}\right]$$

$$\approx \left[\begin{array}{cc} \left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}\right)^{-1} & 0 \\ 0 & \left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,2}+\tilde{\chi}_{3,2}\mathbf{C}_{2,3}\right)^{-1} \end{array}\right] \quad (4.32)$$

This simplified block-diagonal matrix becomes a sum of full uncooperative FIM for

$\mathbf{J}_{\mathbf{x}_i}$ and $\tilde{\chi}_{i,j}$ weighted combination of $\mathbf{C}_{i,j}$. However, the uncertain weighting of mobile *3* to mobile *2*, $\chi_{2,3}$ is replaced by $\tilde{\chi}_{2,3}$ because the **J** on the lemma *1* is $\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}$. Therefore, $\tilde{\chi}_{2,3}$ is involved in the eigen-parameter of $\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}$. Then, (4.30) can be written as

$$\mathbf{J}_{e_1} \approx \left(\mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}+\mathbf{C}_{1,3}\right)-\begin{bmatrix}\mathbf{C}_{1,2} & \mathbf{C}_{1,3}\end{bmatrix}\begin{bmatrix}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}\right)^{-1} & 0\\ 0 & \left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,2}+\tilde{\chi}_{3,2}\mathbf{C}_{2,3}\right)^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{C}_{1,2}\\ \mathbf{C}_{1,3}\end{bmatrix} \quad (4.33)$$

$$=\mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}+\mathbf{C}_{1,3}-\mathbf{C}_{1,2}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}\right)^{-1}\mathbf{C}_{1,2}-\mathbf{C}_{1,3}\left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,2}+\tilde{\chi}_{3,2}\mathbf{C}_{2,3}\right)^{-1}\mathbf{C}_{1,3} \quad (4.34)$$

$$=\mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}-\mathbf{C}_{1,2}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}\right)^{-1}\mathbf{C}_{1,2}+\mathbf{C}_{1,3}-\mathbf{C}_{1,3}\left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,2}+\tilde{\chi}_{3,2}\mathbf{C}_{2,3}\right)^{-1}\mathbf{C}_{1,3} \quad (4.35)$$

$$=\mathbf{J}_{\mathbf{x}_1}+\tilde{\chi}_{1,2}\mathbf{C}_{1,2}+\tilde{\chi}_{1,3}\mathbf{C}_{1,3} \quad (4.36)$$

As before, the cause of uncertain weighting $\tilde{\chi}_{1,2}$ and $\tilde{\chi}_{1,3}$ is that the **J** on the lemma *1* are $\mathbf{J}_{\mathbf{x}_2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}$ and $\mathbf{J}_{\mathbf{x}_3}+\tilde{\chi}_{3,2}\mathbf{C}_{2,3}$, respectively. In fact, we know $\tilde{\chi}_{1,2} > \chi_{1,2}$ and $\tilde{\chi}_{1,3} > \chi_{1,3}$ because $\tilde{\chi}_{1,2}$ and $\tilde{\chi}_{1,3}$ have the help from $\tilde{\chi}_{2,3}\mathbf{C}_{2,3}$. But the help is weak. Then, we can see the approximated uncertain weighting as follows

$$\tilde{\chi}_{1,2} \approx \chi_{1,2} \quad (4.37)$$

$$\tilde{\chi}_{1,3} \approx \chi_{1,3} \quad (4.38)$$

Therefore, we derive the approximated EFIM (AEFIM) for mobile *1*

$$\mathbf{J}_{e_1} \approx \mathbf{J}_{\mathbf{x}_1}+\chi_{1,2}\mathbf{C}_{1,2}+\chi_{1,3}\mathbf{C}_{1,3} \quad (4.39)$$

We can see that the AEFIM of mobile *1* includes full uncooperative FIM of mobile *1* add cooperative fisher information from mobile *2* and mobile *3* with their uncertain

weighting $\chi_{1,2}$ and $\chi_{1,3}$.

## Case b: Four Mobiles

The full cooperative FIM for four mobiles is given by

$$\mathbf{J}_{Coop,\mathbf{x}(4)} = \begin{bmatrix} \mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}+\mathbf{C}_{1,3}+\mathbf{C}_{1,4} & -\mathbf{C}_{1,2} & -\mathbf{C}_{1,3} & -\mathbf{C}_{1,4} \\ -\mathbf{C}_{1,2} & \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4} & -\mathbf{C}_{2,3} & -\mathbf{C}_{2,4} \\ -\mathbf{C}_{1,3} & -\mathbf{C}_{2,3} & \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4} \\ -\mathbf{C}_{1,4} & -\mathbf{C}_{2,4} & -\mathbf{C}_{3,4} & \mathbf{J}_{\mathbf{x}_4}+\mathbf{C}_{1,4}+\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{bmatrix}_{8\times 8}$$

(4.40)

The EFIM of mobile *1* can be donated as

$$\mathbf{J}_{e_1} = \mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}+\mathbf{C}_{1,3}+\mathbf{C}_{1,4} - \begin{bmatrix} \mathbf{C}_{1,2} & \mathbf{C}_{1,3} & \mathbf{C}_{1,4} \end{bmatrix} \begin{bmatrix} \mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4} & -\mathbf{C}_{2,3} & -\mathbf{C}_{2,4} \\ -\mathbf{C}_{2,3} & \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4} \\ -\mathbf{C}_{2,4} & -\mathbf{C}_{3,4} & \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,4}+\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C}_{1,2} \\ \mathbf{C}_{1,3} \\ \mathbf{C}_{1,4} \end{bmatrix}$$

(4.41)

As before, the difficult part in (4.34) is three cooperative block matrix

$$\left[\begin{array}{c|cc} \mathbf{J}_{A,2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4} & -\mathbf{C}_{2,3} & -\mathbf{C}_{2,4} \\ \hline -\mathbf{C}_{2,3} & \mathbf{J}_{A,3}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4} \\ -\mathbf{C}_{2,4} & -\mathbf{C}_{3,4} & \mathbf{J}_{A,4}+\mathbf{C}_{1,4}+\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{array}\right]^{-1}$$

(4.42)

Apply the lemma *1* on (4.42)

$$\left[\begin{array}{c|cc} \mathbf{J}_{A,2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4} & -\mathbf{C}_{2,3} & -\mathbf{C}_{2,4} \\ \hline -\mathbf{C}_{2,3} & \mathbf{J}_{A,3}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4} \\ -\mathbf{C}_{2,4} & -\mathbf{C}_{3,4} & \mathbf{J}_{A,4}+\mathbf{C}_{1,4}+\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{array}\right]^{-1} = \left[\begin{array}{c|c} \mathbf{A}_1^{-1} & \mathbf{A}_2 \\ \hline \mathbf{A}_2^{T} & \mathbf{A}_3^{-1} \end{array}\right]$$

(4.43)

As before, $\mathbf{A}_1$ and $\mathbf{A}_2$ can be written as

$$\mathbf{A}_1 = \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4} \qquad (4.44)$$

$$\mathbf{A}_2 \approx \begin{bmatrix} \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad (4.45)$$

Next, we have $\mathbf{A}_3$

$$\mathbf{A}_3 = \begin{bmatrix} \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4} \\ -\mathbf{C}_{3,4} & \mathbf{J}_{\mathbf{x}_4}+\mathbf{C}_{1,4}+\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{bmatrix} - \begin{bmatrix} \mathbf{C}_{2,3} \\ \mathbf{C}_{2,4} \end{bmatrix} \left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4}\right)^{-1} \begin{bmatrix} \mathbf{C}_{2,3} & \mathbf{C}_{2,4} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4}-\mathbf{C}_{2,3}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4}\right)^{-1}\mathbf{C}_{2,4} \\ -\mathbf{C}_{3,4}-\mathbf{C}_{2,4}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4}\right)^{-1}\mathbf{C}_{2,3} & \mathbf{J}_{\mathbf{x}_4}+\mathbf{C}_{1,4}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4} \\ -\mathbf{C}_{3,4} & \mathbf{J}_{\mathbf{x}_4}+\mathbf{C}_{1,4}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{bmatrix} + \begin{bmatrix} 0 & -\mathbf{C}_{2,3}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4}\right)^{-1}\mathbf{C}_{2,4} \\ -\mathbf{C}_{2,4}\left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4}\right)^{-1}\mathbf{C}_{2,3} & 0 \end{bmatrix}$$

$$\approx \begin{bmatrix} \mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4} \\ -\mathbf{C}_{3,4} & \mathbf{J}_{\mathbf{x}_4}+\mathbf{C}_{1,4}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{bmatrix}$$

(4.46)

As before, apply lemma 1 on (4.46) to obtain $\mathbf{A}_3^{-1}$

$$\mathbf{A}_3^{-1} \approx \begin{bmatrix} \left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}+\tilde{\chi}_{3,4}\mathbf{C}_{3,4}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \left(\mathbf{J}_{\mathbf{x}_4}+\mathbf{C}_{1,4}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4}+\tilde{\chi}_{3,4}\mathbf{C}_{3,4}\right)^{-1} \end{bmatrix}$$

Summarize $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$, the three cooperative block matrix in (4.43) can be simplify as

$$\left[\begin{array}{c|cc} \mathbf{J}_{A,2}+\mathbf{C}_{1,2}+\mathbf{C}_{2,3}+\mathbf{C}_{2,4} & -\mathbf{C}_{2,3} & -\mathbf{C}_{2,4} \\ \hline -\mathbf{C}_{2,3} & \mathbf{J}_{A,3}+\mathbf{C}_{1,3}+\mathbf{C}_{2,3}+\mathbf{C}_{3,4} & -\mathbf{C}_{3,4} \\ -\mathbf{C}_{2,4} & -\mathbf{C}_{3,4} & \mathbf{J}_{A,4}+\mathbf{C}_{1,4}+\mathbf{C}_{2,4}+\mathbf{C}_{3,4} \end{array}\right]^{-1}$$

$$\approx \begin{bmatrix} \left(\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4}\right)^{-1} & 0 & 0 \\ & \left(\mathbf{J}_{\mathbf{x}_3}+\mathbf{C}_{1,3}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}+\tilde{\chi}_{3,4}\mathbf{C}_{3,4}\right)^{-1} & 0 \\ 0 & 0 & \left(\mathbf{J}_{\mathbf{x}_4}+\mathbf{C}_{1,4}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4}+\tilde{\chi}_{3,4}\mathbf{C}_{3,4}\right)^{-1} \end{bmatrix}$$

(4.47)

From (4.43) to (4.47) we use recursive block matrix inversion on lemma *1*.It is also a simplified block-diagonal matrix. The EFIM of mobile *1* in (4.41) can be written as

follows

$$\mathbf{J}_1 \approx \mathbf{J}_{A,1}+\mathbf{C}_{1,2}+\mathbf{C}_{1,3}+\mathbf{C}_{1,4}-\begin{bmatrix}\mathbf{C}_{1,2} & \mathbf{C}_{1,3} & \mathbf{C}_{1,4}\end{bmatrix}\begin{bmatrix}\left(\mathbf{J}_{A,2}+\mathbf{C}_{1,2}+\tilde{\chi}_{2,3}\mathbf{C}_{2,3}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4}\right)^{-1} & 0 & 0\\ 0 & \left(\mathbf{J}_{A,3}+\mathbf{C}_{1,3}+\chi_{2,3}\mathbf{C}_{2,3}+\chi_{3,4}\mathbf{C}_{3,4}\right)^{-1} & 0\\ 0 & 0 & \left(\mathbf{J}_{A,4}+\mathbf{C}_{1,4}+\tilde{\chi}_{2,4}\mathbf{C}_{2,4}+\tilde{\chi}_{3,4}\mathbf{C}_{3,4}\right)^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{C}_{1,2}\\ \mathbf{C}_{1,3}\\ \mathbf{C}_{1,4}\end{bmatrix}$$

$$= \mathbf{J}_{\mathbf{x}_1}+\tilde{\chi}_{1,2}\mathbf{C}_{1,2}+\tilde{\chi}_{1,3}\mathbf{C}_{1,3}+\tilde{\chi}_{1,4}\mathbf{C}_{1,4} \qquad (4.48)$$

As before, the AEFIM of mobile *1* is given by

$$\mathbf{J}_1 \approx \mathbf{J}_{A,1}+\chi_{1,2}\mathbf{C}_{1,2}+\chi_{1,3}\mathbf{C}_{1,3}+\chi_{1,4}\mathbf{C}_{1,4}. \qquad (4.49)$$

From (4.43) to (4.47) we use recursive block matrix inversion on lemma *1* to get the simply formulation. In fact, there are two main assumptions: (1) The off-diagonal block matrix are zero after inverse operation (lemma 1). (2) The approximated uncertain weighting (4.37) and (4.38). We summarize the AEFIM for *M* mobiles in following case.

## Case c: *M* Mobiles

Summarize, the full cooperative FIM becomes for *M* mobiles

$$\mathbf{J}_{Coop,\mathbf{x}(M)} = \begin{bmatrix}\mathbf{J}_{\mathbf{x}_i}+\mathbf{C}_{1,2}+\cdots\mathbf{C}_{1,M} & -\mathbf{C}_{1,2} & \cdots & -\mathbf{C}_{1,M}\\ -\mathbf{C}_{1,2} & \mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\cdots\mathbf{C}_{2,M} & \cdots & -\mathbf{C}_{2,M}\\ \vdots & \vdots & \ddots & \vdots\\ -\mathbf{C}_{1,M} & \mathbf{C}_{2,M} & \cdots & \mathbf{J}_{\mathbf{x}_M}+\mathbf{C}_{1,M}+\cdots\mathbf{C}_{M-1,M}\end{bmatrix}_{2M\times 2M}$$

(4.50)

The EFIM for mobile *1* is denoted as

$$\mathbf{J}_{e_1}=\mathbf{J}_{\mathbf{x}_1}+\mathbf{C}_{1,2}+\cdots\mathbf{C}_{1,M}-\begin{bmatrix}\mathbf{C}_{1,2} & \mathbf{C}_{1,3} & \cdots & \mathbf{C}_{1,M}\end{bmatrix}\begin{bmatrix}\mathbf{J}_{\mathbf{x}_2}+\mathbf{C}_{1,2}+\cdots\mathbf{C}_{2,M} & \cdots & -\mathbf{C}_{2,M}\\ \vdots & \ddots & \vdots\\ \mathbf{C}_{2,M} & \cdots & \mathbf{J}_{\mathbf{x}_M}+\mathbf{C}_{1,M}+\cdots\mathbf{C}_{M-1,M}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{C}_{1,2}\\ \mathbf{C}_{1,3}\\ \vdots\\ \mathbf{C}_{1,M}\end{bmatrix}$$

(4.51)

As before, by recursive block matrix inversion, $(2M-2)\times(2M-2)$ block matrix in

(4.46) can be partitioned into upper-left single block $2\times 2$ and lower-right $(2M-4)\times(2M-4)$ block matrices, and $(2M-4)\times(2M-4)$ block matrix can be further partitioned into upper-left single block and lower-right $(2M-6)\times(2M-6)$ block matrices, and so on. Therefore, (4.46) can be simplified as

$$\mathbf{J}_{e_1} \approx \mathbf{J}_{A.1} + \mathbf{C}_{1,2} + \cdots \mathbf{C}_{1,M} - \begin{bmatrix} \mathbf{C}_{1,2} & \mathbf{C}_{1,3} & \cdots & \mathbf{C}_{1,M} \end{bmatrix} \begin{bmatrix} \left(\mathbf{J}_{A,2} + \mathbf{C}_{1,2} + \cdots \tilde{\chi}_{2,M}\mathbf{C}_{2,M}\right)^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \left(\mathbf{J}_{A.M} + \mathbf{C}_{1,M} + \cdots \tilde{\chi}_{M-1,M}\mathbf{C}_{M-1,M}\right)^{-1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C}_{1,2} \\ \mathbf{C}_{1,3} \\ \vdots \\ \mathbf{C}_{1,M} \end{bmatrix}$$

(4.52)

$$= \mathbf{J}_{\mathbf{x}_1} + \tilde{\chi}_{1,2}\mathbf{C}_{1,2} + \cdots + \tilde{\chi}_{1M}\mathbf{C}_{1,M} \qquad (4.53)$$

$$\approx \mathbf{J}_{\mathbf{x}_1} + \chi_{1,2}\mathbf{C}_{1,2} + \cdots + \chi_{1M}\mathbf{C}_{1,M} \qquad (4.54)$$

From the AEFIM of mobile *1* for *M* mobiles , $\mathbf{J}_{e_1}$ (4.49), it is a linear combination of an full uncooperative FIM $\mathbf{J}_{\mathbf{x}_1}$ and other cooperative information matrixes from mobile *2* to mobile *M*, $\mathbf{C}_{1,2}, \mathbf{C}_{1,3} \ldots, \mathbf{C}_{1,M}$ with their approximated uncertain weighting, $\chi_{1,2}, \chi_{1,3} \ldots, \chi_{1,M}$. We can see that the more mobiles, the better localization accuracy. Therefore, the cooperative CRLB for *M>2* mobiles can be obtain easily

$$\text{CRLB}_{\text{mobile 1}} \approx \text{trace}\left[\mathbf{J}_{\mathbf{e}_1}^{-1}\right]_{2\times 2} \qquad (4.55)$$

### 4.1.5 Approximated Cooperative CRLB in Case of a Central Mobile

The research utilizes the eigenvalue view to observe the cooperative FIM, but it can see the cooperation benefit only for two mobiles and do not derive a close from of cooperative CRLB. Then, we discuss cooperative CRLB for special case, a central mobile. Figure 4.1 shows the central mobile *1* in a localization system. All sensor locations are uniformly located around mobile *1*

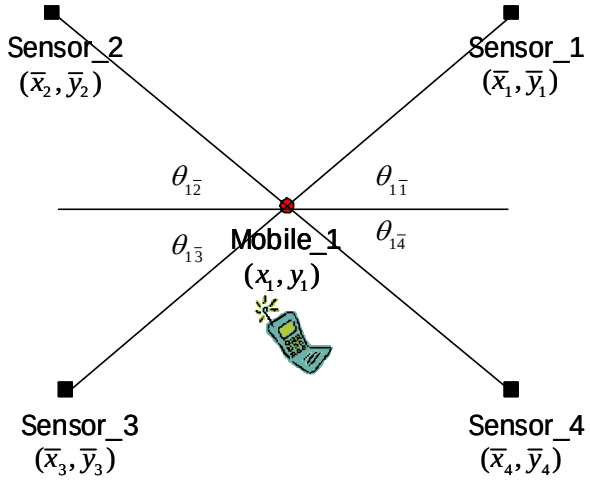



Figure 4.1 The angle relationship between sensors and mobile *1*.

Apply the AEFIM result, the cooperative CRLB (4.55) can be simplified as

$$\text{CRLB}_{\text{mobile } i} \approx \text{trace}\left[\left(\mathbf{J}_{e_1}^{-1}\right)\right] \qquad (4.55)$$

where $\mathbf{J}_{e_1} \approx \mathbf{J}_{\mathbf{x}_1} + \chi_{1,2}\mathbf{C}_{1,2} + \cdots + \chi_{1M}\mathbf{C}_{1,M}$ is given in (4.54).

Here we make some assumptions as follows

Assumptions: 1. i.i.d. noise.

2. Because Mobile *1* is located near the center of all sensors. Therefore, the eigenvalues of uncooperative EFIM of mobile *1* are the same, $\lambda_1 = \mu_1$. Therefore,

$$\mathbf{J}_{\mathbf{x}_1} = \sum_{j=1}^{N} \frac{1}{\sigma^2} \begin{bmatrix} \cos^2\theta_{1\bar{j}} & \cos\theta_{1\bar{j}}\sin\theta_{1\bar{j}} \\ \cos\theta_{1\bar{j}}\sin\theta_{1\bar{j}} & \sin^2\theta_{1\bar{j}} \end{bmatrix} = \lambda_1 \mathbf{I}, \text{ and } \lambda_1 = \frac{N}{2\sigma^2}. \qquad (4.56)$$

Form figure 4.2, we can see the on-diagonal are $\sum_{j=2}^{M}\cos^2\theta_{1\bar{j}}$ and $\sum_{j=2}^{M}\sin^2\theta_{1\bar{j}}$ which have similar value in uniform angle situation. Then, the off-diagonal is

$$\sum_{j=2}^{M}\cos\theta_{1\bar{j}}\sin\theta_{1\bar{j}} = 1/2\sum_{j=2}^{M}\sin 2\theta_{1\bar{j}} \approx 0.$$

Finally, we have AEFIM of mobile as be written as

$$\mathbf{J}_{e_1} \approx \frac{N}{2\sigma^2}\mathbf{I} + \chi_{1,2}\mathbf{C}_{1,2} + \cdots + \chi_{1M}\mathbf{C}_{1,M} \qquad (4.57)$$

We separate two cases to discuss.

**Case 1**: Same uncertain weighting, $\chi_{1,2} = \chi_{1,3} = \cdots = \chi_{1,M} = \chi$ .

$$\mathbf{J}_{e_1} \approx \frac{N}{2\sigma^2}\mathbf{I} + \chi\left(\mathbf{C}_{1,2} + \cdots + \mathbf{C}_{1,M}\right) \qquad (4.58)$$

As before, we do eigenvalue composition on $\left(\mathbf{C}_{1,2} + \cdots + \mathbf{C}_{1,M}\right)$

$$\left(\mathbf{C}_{1,2} + \cdots + \mathbf{C}_{1,M}\right) \triangleq \mathbf{U}_{\phi_1}^T \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \mathbf{U}_{\phi_1} \qquad (4.59)$$

where $\alpha + \beta = \frac{M-1}{\sigma^2}$ .

The AEFIM of mobile 1 can be written as follows

$$\mathbf{J}_{e_1} \approx \mathbf{U}_{\phi_1}^T \begin{bmatrix} \frac{N}{2\sigma^2} + \chi\alpha & 0 \\ 0 & \frac{N}{2\sigma^2} + \chi\beta \end{bmatrix} \mathbf{U}_{\phi_1} \qquad (4.60)$$

The cooperative CRLB is given by

$$\text{AC-CRLB}_{\text{mobile 1}} = \frac{1}{\frac{N}{2\sigma^2} + \chi\alpha} + \frac{1}{\frac{N}{2\sigma^2} + \chi\beta} \qquad (4.61)$$

From (4.61), we have maximum and minimum value as follows

$$\text{Omni direction}: \ \alpha = \beta = \frac{M-1}{2\sigma^2} \Rightarrow \text{AC-CRLB}_{\text{mobile 1}} = \frac{4\sigma^2}{N + \chi(M-1)} \qquad (4.62)$$

$$\text{Beam}: \ \alpha = \frac{M-1}{\sigma^2}, \ \beta = 0 \Rightarrow \text{AC-CRLB}_{\text{mobile 1}} = \frac{2\sigma^2}{N + 2\chi(M-1)} + \frac{2\sigma^2}{N} \qquad (4.63)$$

From (4.62) and (4.63) we can know the Omni direction is better than beam at same uncertain mobiles help. Figures 4.2 and 4.3 show that omni-direction and beam of cooperative mobiles with the same cooperative reliable positions.

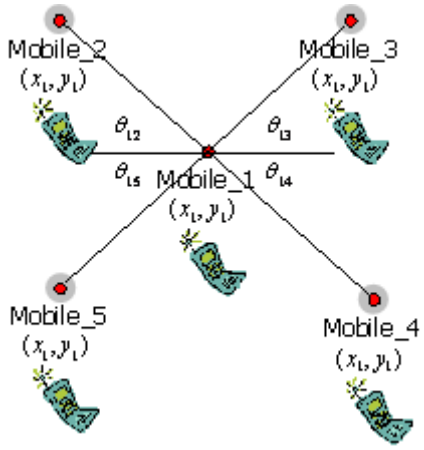



Figure 4.2 Omni-direction type with same reliable cooperative positions.

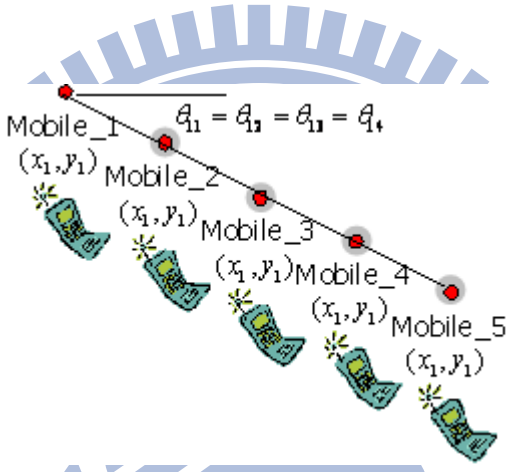



Figure 4.3 Beam type with same reliable cooperative positions.

**Case 2**: Difference uncertain weighting, $\chi_{1,2} \neq \chi_{1,3} \neq \cdots \neq \chi_{1,M}$.

Assume beam type of cooperative mobiles. Figure 4.4 show that.

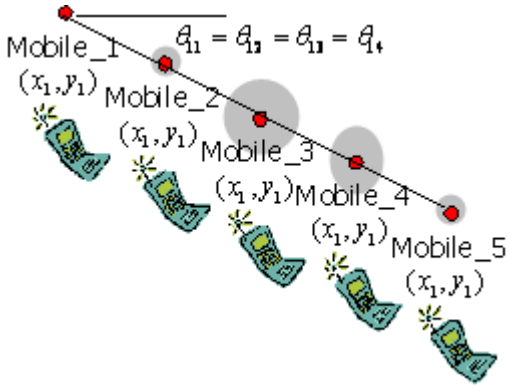



Figure 4.4 Beam type with difference reliable cooperative positions.

The AEFIM of mobile *1* is given by

$$\mathbf{J}_{e_1} \approx \frac{N}{2\sigma^2}\mathbf{I} + \sum_{j=2}^{M} \chi_{1,j}\mathbf{C} \qquad (4.64)$$

$$= \frac{N}{2\sigma^2}\mathbf{I} + \sum_{j=2}^{M} \chi_{1,j}\mathbf{U}_\theta^T \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{U}_\theta$$

$$= \mathbf{U}_\theta^T \begin{bmatrix} \frac{1}{\sigma^2}\sum_{j=2}^{M} \chi_{1,j} + \frac{N}{2\sigma^2} & 0 \\ 0 & \frac{N}{2\sigma^2} \end{bmatrix} \mathbf{U}_\theta$$

The cooperative CRLB of mobile *1* can be denoted as

$$\text{AC-CRLB}_{\text{mobile 1}} = \frac{2\sigma^2}{N}\left(1 + \frac{1}{1 + \frac{N}{2}\sum_{j=2}^{M} \chi_{1,j}}\right) \qquad (4.65)$$

From the results of situations (4.62), (4.63) and (4.65), these parameters, numbers of mobiles and sensors, noise variance and uncertain weighting $\chi_{1,j}$ can affect localization accuracy.

We derive AC-CRLB successfully and find some factors which can improve the cooperative localization accuracy. The AC-CRLB is shown in computer simulation, in terms of noise variance, numbers of mobiles and cooperative angles. We also compare the difference between cooperative CRLB from full cooperative FIM, AEFIM and AC-CRLB.

## 4.2 Converged Theoretical Mean Square Error for Divided Linearized Algorithms

Previously, we proposed the divide-and-conquer method for cooperative localization system. It can perform low computation cost and available MSE performance. Now, we try to analyze the converged MSE for Divided linearized

algorithms. However, the most analysis of these algorithms is quite hard but we still try to confess the converged theoretical MSE for divided TS, DA and HC. Especially, we derive it successfully for divided TS in special case *M=2*. Simulation results demonstrate the converged theoretical MSE.

The rest of the section is organized as follows. Convergence analysis of theoretical MSE will be described for three linearized divide-and-conquer algorithms: Taylor-series expansion algorithm in Section 4.2.1, distance-augmented algorithm in Section 4.2.2, Hyperbolic-canceled algorithm in Section 4.2.3.

### 4.2.1 Taylor-Series Expansion Algorithm

First, we assume all variance of measurement error are the same as $\sigma^2$ and the reference points are quite close to true points, $(x_{10}, y_{10}) \approx (x_1, y_1)$, $(x_{20}, y_{20}) \approx (x_2, y_2)$, …, $(x_{M0}, y_{M0}) \approx (x_M, y_M)$. Consider two estimations from divided TS, weighted estimation, $\hat{\mathbf{x}}_{i\_DWTS,n+1}$ in (3.36) and unweighted estimation, $\hat{\mathbf{x}}_{i\_DTS,n+1}$ (3.38) for mobile *i* at *n+1*th iteration. Now, we derive the convergence analysis of MSE. Without loss of generality, let *i*=1. The covariance matrix of weighted estimation, $\mathbf{e}_{1\_DWTS,n+1} = \hat{\mathbf{x}}_{1\_DWTS,n+1} - \mathbf{x}_1$, conditioned on position of virtual sensors *2* to *M*, $\hat{\mathbf{x}}_{2,n}, \hat{\mathbf{x}}_{3,n}, \cdots, \hat{\mathbf{x}}_{M,n}$ is denoted as,

$$\begin{aligned} \operatorname{cov}\left(\mathbf{e}_{1\_DWTS,n+1} \mid \hat{\mathbf{x}}_{2,n}, \hat{\mathbf{x}}_{3,n}, \cdots, \hat{\mathbf{x}}_{M,n}\right) &= \left(\mathbf{H}_{1\_DTS,n}{}^{T} \mathbf{W}_{1\_DTS,n} \mathbf{H}_{1\_DTS,n}\right)^{-1} \\ &= \left[\frac{1}{\sigma^2}\mathbf{H}_{1\_TS}{}^{T}\mathbf{H}_{1\_TS} + \sum_{j=2}^{M} \frac{1}{\sigma^2 + \hat{\sigma}^2_{j\_VTS,n}} \mathbf{a}_{10,\hat{j},n}\mathbf{a}_{10,\hat{j},n}{}^{T}\right]^{-1} \end{aligned} \tag{4.66}$$

where $\mathbf{H}_{1\_DTS,n}{}^{T} = \left[\mathbf{H}_{1\_TS,n}{}^{T} \quad \mathbf{a}_{10,\hat{2},n} \quad \mathbf{a}_{10,\hat{3},n} \quad \cdots \quad \mathbf{a}_{10,\hat{M},n}\right]$ is divided angle matrix in (3.29) and $\mathbf{W}_{i\_DTS,n}$ is divided TS weighting in (3.37).

After unconditioning $\hat{\mathbf{x}}_{2,n},\hat{\mathbf{x}}_{3,n},\cdots,\hat{\mathbf{x}}_{M,n}$, the MSE of mobile *i* at $n+1^{\text{th}}$ iteration is

$$\hat{\sigma}^2_{i\_DWTS,n+1}=\text{trace}\left\{E\left\{\text{cov}\left(\mathbf{e}_{1\_DWTS,n+1}\,|\,\hat{\mathbf{x}}_{2\_VTS,n},\hat{\mathbf{x}}_{3\_VTS,n},\cdots,\hat{\mathbf{x}}_{M\_VTS,n}\right)\right\}\right\} \quad (4.67)$$

Then, the another error covariance of unweighted estimation $\mathbf{e}_{1\_DTS,n+1}=\hat{\mathbf{x}}_{1\_DTS,n+1}-\mathbf{x}_1$ conditioned on position of virtual sensors *2* to *M*, $\hat{\mathbf{x}}_{2,n},\hat{\mathbf{x}}_{3,n},\cdots,\hat{\mathbf{x}}_{M,n}$ is given by

$$\begin{aligned}&\text{cov}\left(\mathbf{e}_{1\_DTS,n+1}\,|\,\hat{\mathbf{x}}_{2,n},\hat{\mathbf{x}}_{3,n},\cdots,\hat{\mathbf{x}}_{M,n}\right)\\&=\left(\mathbf{H}_{1\_DTS,n}{}^T\mathbf{H}_{1\_DTS,n}\right)^{-1}\mathbf{H}_{1\_DTS,n}{}^T\mathbf{W}_{1\_DTS,n}{}^{-1}\mathbf{H}_{1\_DTS,n}\left(\mathbf{H}_{1\_DTS,n}{}^T\mathbf{H}_{1\_DTS,n}\right)^{-1}\\&=\mathbf{A}_n\mathbf{B}_n\mathbf{A}_n \quad (4.68)\end{aligned}$$

Where $\mathbf{A}_n=\left[\mathbf{H}_{1\_TS}^T\mathbf{H}_{1\_TS}+\sum_{j=2}^{M}\mathbf{a}_{10,\hat{j},n}\mathbf{a}_{10,\hat{j},n}^T\right]^{-1}$,

$$\mathbf{B}_n=\left[\sigma^2\mathbf{H}_{1\_TS}^T\mathbf{H}_{1\_TS}+\sum_{j=2}^{M}\left(\sigma^2+\sigma^2_{\hat{j}\_VTS,n}\right)\mathbf{a}_{10,\hat{j},n}\mathbf{a}_{10,\hat{j},n}^T\right]$$

Then the MSE is

$$\hat{\sigma}^2_{1\_DTS,n+1}=\text{trace}\left\{E\left\{\text{cov}\left(\mathbf{e}_{1\_DTS,n+1}\,|\,\hat{\mathbf{x}}_{2,n},\hat{\mathbf{x}}_{3,n},\cdots,\hat{\mathbf{x}}_{M,n}\right)\right\}\right\} \quad (4.69)$$

It is very hard to analyze (4.67) and (4.69) because the virtual sensors locations have complex statistic. In order to simplify our analysis, we will consider the special case of *M=2* in Section 4.2.1a and Section 4.2.1b, respectively.

## 4.2.1a Weighted Estimation for Two Mobiles

In case of *M=2* mobiles, the error covariance of weighted estimation $\mathbf{e}_{1\_DWTS,n+1}$ for mobile *1* at $1^{\text{st}}$ global iteration in (4.67) becomes a rank-one update of an inverse matrix,

$$\operatorname{cov}\left[\mathbf{e}_{1\_DWTS,1}|\hat{\mathbf{x}}_{2,0}\right]=\left[\frac{1}{\sigma^2}\mathbf{H}_{1\_TS}{}^{T}\mathbf{H}_{1\_TS}+\frac{1}{\sigma^2+\hat{\sigma}^2_{2\_VTS,0}}\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]^{-1}$$

(4.70)

In (4.70), we can see $\left(1/\sigma^2\,\mathbf{H}_{1\_TS}{}^{T}\mathbf{H}_{1\_TS}\right)^{-1}=\operatorname{cov}\left(\mathbf{e}_{1\_WTS}\right)$ is theoretical error covariance of uncooperative TS estimation in (2.63). Therefore, the virtual sensor *2* must provides benefits for localization accuracy. Now, we use the eigenvalue decomposition on $\mathbf{H}_{1\_TS}{}^{T}\mathbf{H}_{1\_TS}$,

$$\mathbf{H}_{1\_TS}{}^{T}\mathbf{H}_{1\_TS}\triangleq\mathbf{U}^{T}_{\beta_{10}}\begin{bmatrix}\lambda_{10} & 0\\ 0 & \mu_{10}\end{bmatrix}\mathbf{U}_{\beta_{10}} \quad (4.71)$$

Based on (4.71), (4.70) can be written as follows,

$$\operatorname{cov}\left[\mathbf{e}_{1\_DWTS,1}|\hat{\mathbf{x}}_{2,0}\right]$$

$$=\left[\frac{1}{\sigma^2}\mathbf{U}^{T}_{\beta_{10}}\begin{bmatrix}\lambda_{10} & 0\\ 0 & \mu_{10}\end{bmatrix}\mathbf{U}_{\beta_{10}}+\frac{1}{\sigma^2+\hat{\sigma}^2_{2\_VTS,0}}\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]^{-1} \quad (4.72)$$

$$=\left[\frac{1}{\sigma^2}\mathbf{U}^{T}_{\beta_{10}}\begin{bmatrix}\lambda_{10} & 0\\ 0 & \mu_{10}\end{bmatrix}\mathbf{U}_{\beta_{10}}+\frac{1}{\sigma^2+\hat{\sigma}^2_{2\_VTS,0}}\mathbf{U}^{T}_{\beta_{10}}\mathbf{U}_{\beta_{10}}\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\mathbf{U}^{T}_{\beta_{10}}\mathbf{U}_{\beta_{10}}\right]^{-1} \quad (4.73)$$

$$=\mathbf{U}^{T}_{\beta_{10}}\left(\frac{1}{\sigma^2}\begin{bmatrix}\lambda_{10} & 0\\ 0 & \mu_{10}\end{bmatrix}+\frac{1}{\sigma^2+\hat{\sigma}^2_{2\_VTS,0}}\mathbf{C}_{\beta_{10},\hat{2},0}\mathbf{C}^{T}_{\beta_{10},\hat{2},0}\right)^{-1}\mathbf{U}_{\beta_{10}} \quad (4.74)$$

Where $\mathbf{C}_{\beta_{10},\hat{2},0}=\mathbf{U}_{\beta_{10}}\mathbf{a}_{10,\hat{2},0}$, and $\mathbf{C}^{T}_{\beta_{10},\hat{2},0}\mathbf{C}_{\beta_{10},\hat{2},0}=\mathbf{I}$.

In order to simplify (4.74), we assume the mobile *1* is located at the center around all sensors. As before, $\lambda_{10}=\mu_{10}$. (4.74) can be simplified as.

$$\operatorname{cov}\left[\mathbf{e}_{1\_DWTS,1}|\hat{\mathbf{x}}_{2,0}\right]=\mathbf{U}^{T}_{\beta_{10}}\left(\frac{\lambda_{10}}{\sigma^2}\mathbf{I}+\frac{1}{\sigma^2+\hat{\sigma}^2_{2\_VTS,0}}\mathbf{C}_{\beta_{10},\hat{2},0}\mathbf{C}^{T}_{\beta_{10},\hat{2},0}\right)^{-1}\mathbf{U}_{\beta_{10}} \quad (4.75)$$

Apply matrix inverse lemma [24] on inverse term on (4.75)

matrix inverse lemma: $\left(\mathbf{A}+a\mathbf{C}\mathbf{C}^{T}\right)^{-1}=\mathbf{A}^{-1}-a\mathbf{A}^{-1}\mathbf{C}\left(\mathbf{I}+a\mathbf{C}^{T}\mathbf{A}^{-1}\mathbf{C}\right)^{-1}\mathbf{C}^{T}\mathbf{A}^{-1}$

we have

$$\left(\frac{\lambda_{10}}{\sigma^2}\mathbf{I}+\frac{1}{\sigma^2+\hat{\sigma}^2_{2\_VTS,0}}\mathbf{C}_{\beta_{10},\hat{2},0}\mathbf{C}^T_{\beta_{10},\hat{2},0}\right)^{-1}$$

$$=\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\frac{\sigma^4}{\lambda_{10}^2\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}\mathbf{C}_{\beta_{10},\hat{2},0}\left(\mathbf{I}+\frac{\sigma^2}{\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}\mathbf{C}^T_{\beta_{10},\hat{2},0}\mathbf{C}_{\beta_{10},\hat{2},0}\right)^{-1}\mathbf{C}^T_{\beta_{10},\hat{2},0}$$

(4.76)

$$=\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\frac{\sigma^4}{\lambda_{10}^2\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}\mathbf{C}_{\beta_{10},\hat{2},0}\left(\mathbf{I}+\frac{\sigma^2}{\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}\mathbf{I}\right)^{-1}\mathbf{C}^T_{\beta_{10},\hat{2},0} \quad (4.77)$$

$$=\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\frac{\sigma^4}{\lambda_{10}^2\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}\mathbf{C}_{\beta_{10},\hat{2},0}\left(\frac{\sigma^2+\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}{\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}\right)^{-1}\mathbf{C}^T_{\beta_{10},\hat{2},0} \quad (4.78)$$

$$=\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\frac{\sigma^4}{\lambda_{10}^2\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}\mathbf{C}_{\beta_{10},\hat{2},0}\left(\frac{\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}{\sigma^2+\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)}\right)\mathbf{C}^T_{\beta_{10},\hat{2},0} \quad (4.79)$$

$$=\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\frac{\sigma^4}{\lambda_{10}\left[\sigma^2+\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)\right]}\mathbf{C}_{\beta_{10},\hat{2},0}\mathbf{C}^T_{\beta_{10},\hat{2},0} \quad (4.80)$$

Then, the error covariance can be written as follows

$$\mathrm{cov}\left[\mathbf{e}_{1\_DWTS,1}|\hat{\mathbf{x}}_{2,0}\right]=\mathbf{U}^T_{\beta_{10}}\left(\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\frac{\sigma^4}{\lambda_{10}\left[\sigma^2+\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)\right]}\mathbf{C}_{\beta_{10},\hat{2},0}\mathbf{C}^T_{\beta_{10},\hat{2},0}\right)\mathbf{U}_{\beta_{10}}$$

(4.81)

Next, the variance of weighted divided TS estimator is

$$\sigma^2_{1\_DWTS,1}=\mathrm{trace}\left\{E\left[\mathrm{cov}\left[\mathbf{e}_{1\_DWTS,1}|\hat{\mathbf{x}}_{2,0}\right]\right]\right\} \quad (4.82)$$

$$=E\left\{\mathrm{trace}\left[\mathrm{cov}\left[\mathbf{e}_{1\_DWTS,1}|\hat{\mathbf{x}}_{2,0}\right]\right]\right\} \quad (4.83)$$

(4.82) and (4.83) have same consequence. The we use (4.83) on (4.81), (4.82) can be written as

$$\sigma^2_{1\_DWTS,1}=E\left\{\mathrm{trace}\left[\mathbf{U}^T_{\beta_{10}}\left(\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\frac{\sigma^4}{\lambda_{10}\left[\sigma^2+\lambda_{10}\left(\sigma^2+\hat{\sigma}^2_{2\_VTS,0}\right)\right]}\mathbf{C}_{\beta_{10},\hat{2},0}\mathbf{C}^T_{\beta_{10},\hat{2},0}\right)\mathbf{U}_{\beta_{10}}\right]\right\}$$

(4.84)

$$= E\left\{\text{trace}\left(\frac{\sigma^2}{\lambda_{10}}\mathbf{I} - \frac{\sigma^4}{\lambda_{10}\left[\sigma^2 + \lambda_{10}\left(\sigma^2 + \hat{\sigma}^2_{2\_VTS,0}\right)\right]}\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^T\right)\right\} \quad (4.85)$$

$$= E\left\{\text{trace}\left(\frac{\sigma^2}{\lambda_{10}}\mathbf{I}\right) - \text{trace}\left(\frac{\sigma^4}{\lambda_{10}\left[\sigma^2 + \lambda_{10}\left(\sigma^2 + \hat{\sigma}^2_{2\_VTS,0}\right)\right]}\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^T\right)\right\} \quad (4.86)$$

$$= \text{trace}\left(\frac{\sigma^2}{\lambda_{10}}\mathbf{I}\right) - E\left\{\text{trace}\left(\frac{\sigma^4}{\lambda_{10}\left[\sigma^2 + \lambda_{10}\left(\sigma^2 + \hat{\sigma}^2_{2\_VTS,0}\right)\right]}\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^T\right)\right\} \quad (4.87)$$

$$= \frac{2\sigma^2}{\lambda_{10}} - E\left\{\text{trace}\left(\frac{\sigma^4}{\lambda_{10}\left[\sigma^2 + \lambda_{10}\left(\sigma^2 + \hat{\sigma}^2_{2\_VTS,0}\right)\right]}\mathbf{I}\right)\right\} \quad (4.88)$$

$$= \frac{2\sigma^2}{\lambda_{10}} - \frac{\sigma^4}{\lambda_{10}\left[\sigma^2 + \lambda_{10}\left(\sigma^2 + \hat{\sigma}^2_{2\_VTS,0}\right)\right]} \quad (4.89)$$

$$= \frac{2\sigma^2}{\lambda_{10}} - \frac{\sigma^4}{\lambda_{10}^2} \cdot \frac{1}{\left(1 + 1/\lambda_{10}\right)\sigma^2 + \hat{\sigma}^2_{2\_VTS,0}} \quad (4.90)$$

Actually, we know $2\sigma^2/\lambda_{10} = \sigma^2_{1\_WTS} = \hat{\sigma}^2_{1\_VTS,0}$ in (4.90) which is error variance of uncooperative weighted TS estimation and is also the initial value in global iteration. Then, (4.90) can be written as follows

$$\sigma^2_{1\_DWTS,1} = \sigma^2_{1\_WTS} - \frac{\sigma^4}{\lambda_{10}^2} \cdot \frac{1}{\left(1 + 1/\lambda_{10}\right)\sigma^2 + \sigma^2_{2\_WTS}} \quad (4.91)$$

The MSE of divided TS for mobile *2* can be obtained similarly,

$$\sigma^2_{2\_DWTS,1} = \sigma^2_{2\_WTS} - \frac{\sigma^4}{\lambda_{20}^2} \cdot \frac{1}{\left(1 + 1/\lambda_{20}\right)\sigma^2 + \sigma^2_{1\_WTS}} \quad (4.92)$$

We know that the MSE of error estimations is also the variance of uncertain virtual sensors, $\sigma^2_{1\_DWTS,1} = \hat{\sigma}^2_{1\_VTS,1}$ and $\sigma^2_{2\_DWTS,1} = \hat{\sigma}^2_{2\_VTS,1}$. Therefore, the MSE at $2^{nd}$ global iteration for mobile *1*can be written as

$$\sigma^2_{1\_DWTS,2} = \sigma^2_{1\_WTS} - \frac{\sigma^4}{\lambda_{10}^2} \cdot \frac{1}{\left(1+ 1/\lambda_{10}\right)\sigma^2 + \hat{\sigma}^2_{2\_VTS,1}} \quad (4.93)$$

$$= \sigma^2_{1\_WTS} - \frac{\sigma^4}{\lambda_{10}^2} \cdot \frac{1}{\left(1+ 1/\lambda_{10}\right)\sigma^2 + \sigma^2_{2\_WTS} - \frac{\sigma^4}{\lambda_{20}^2} \cdot \frac{1}{\left(1+ 1/\lambda_{20}\right)\sigma^2 + \sigma^2_{1\_WTS}}} \quad (4.94)$$

At $3^{rd}$ global iteration,

$$\sigma^2_{1\_DWTS,3} = \sigma^2_{1\_WTS} - \frac{\sigma^4}{\lambda_{10}^2} \cdot \frac{1}{\left(1+ 1/\lambda_{10}\right)\sigma^2 + \hat{\sigma}^2_{2\_VTS,2}} \quad (4.95)$$

$$= \sigma^2_{1\_WTS} - \frac{\sigma^4}{\lambda_{10}^2} \cdot \frac{1}{\left(1+ 1/\lambda_{10}\right)\sigma^2 + \sigma^2_{2\_WTS} - \frac{\sigma^4}{\lambda_{20}^2} \cdot \frac{1}{\left(1+ 1/\lambda_{20}\right)\sigma^2 + \sigma^2_{1\_WTS} - \frac{\sigma^4}{\lambda_{10}^2} \cdot \frac{1}{\left(1+ 1/\lambda_{10}\right)\sigma^2 + \sigma^2_{2\_WTS}}}} \quad (4.96)$$

The iteration formulation can written as

$$\sigma^2_{1\_DWTS,n+1} = \sigma^2_{1\_WTS} - \frac{\sigma^4}{\lambda_{10}^2} \cdot \frac{1}{\left(1+ 1/\lambda_{10}\right)\sigma^2 + \hat{\sigma}^2_{2\_VTS,n}} \quad (4.97)$$

The MSE of mobile *2* has the similar result. We can see that the variance of uncertain virtual sensor location, $\hat{\sigma}^2_{2\_VTS,n}$ becomes smaller by global iteration. Therefore, the MSE will become smaller, $\sigma^2_{1\_DWTS,n+1}$ when the variance of uncertain location is tending downward.

The theoretical formulation has no closed form. In the simulation, it can be converged by only three times global iteration later.

## 4.2.1b Unweighted Estimation for Two Mobiles

In case of *M=2* mobiles, the error covariance of unweighted divided TS estimation $\mathbf{e}_{1\_DTS,n+1}$ in for mobile *1* at $1^{st}$ global iteration in (4.68) becomes

$$\operatorname{cov}\left(\mathbf{e}_{1\_DTS,1}\,|\,\hat{\mathbf{x}}_{2,0}\right)$$

$$=\left(\mathbf{H}_{1\_DTS,0}{}^{T}\mathbf{H}_{1\_DTS,0}\right)^{-1}\mathbf{H}_{1\_DTS,0}{}^{T}\mathbf{W}_{1\_DTS,0}{}^{-1}\mathbf{H}_{1\_DTS,0}\left(\mathbf{H}_{1\_DTS,0}{}^{T}\mathbf{H}_{1\_DTS,0}\right)^{-1}$$

$$=\left[\mathbf{H}_{1\_TS}^{T}\mathbf{H}_{1\_TS}+\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]^{-1}\left[\sigma^{2}\mathbf{H}_{1\_TS}^{T}\mathbf{H}_{1\_TS}+\left(\sigma^{2}+\hat{\sigma}_{2\_VTS,0}^{2}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]\left[\mathbf{H}_{1\_TS}^{T}\mathbf{H}_{1\_TS}+\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]^{-1}$$

(4.98)

As before, assume $\mathbf{H}_{1\_TS}^{T}\mathbf{H}_{1\_TS}=\lambda_{10}\mathbf{I}$. Then the error covariance in (4.93) can be written as

$$\operatorname{cov}\left(\mathbf{e}_{1\_DTS,1}\,|\,\hat{\mathbf{x}}_{2,0}\right)$$

$$=\left[\lambda_{10}\mathbf{I}+\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]^{-1}\left[\sigma^{2}\lambda_{10}\mathbf{I}+\left(\sigma^{2}+\hat{\sigma}_{2\_VTS,0}^{2}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]\left[\lambda_{10}\mathbf{I}+\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]^{-1}\qquad(4.99)$$

Apply matrix inverse lemma on inverse term in (4.99), we have

$$\left[\lambda_{10}\mathbf{I}+\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]^{-1}=\frac{1}{\lambda_{10}}\mathbf{I}-\frac{\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}}{\lambda_{10}^{2}\left(1+\lambda_{10}\right)}\qquad(4.100)$$

Expansion of (4.99) by (4.100), the covariance matrix is given by

$$\operatorname{cov}\left(\mathbf{e}_{1\_DTS,1}\,|\,\hat{\mathbf{x}}_{2,0}\right)$$

$$=\left(\frac{1}{\lambda_{10}}\mathbf{I}-\frac{\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}}{\lambda_{10}^{2}\left(1+\lambda_{10}\right)}\right)\left[\sigma^{2}\lambda_{10}\mathbf{I}+\left(\sigma^{2}+\hat{\sigma}_{2\_VTS,0}^{2}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]\left(\frac{1}{\lambda_{10}}\mathbf{I}-\frac{\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}}{\lambda_{10}^{2}\left(1+\lambda_{10}\right)}\right)\qquad(4.101)$$

$$=\left[\sigma^{2}\mathbf{I}+\left(\frac{\left(\sigma^{2}+\hat{\sigma}_{2\_VTS,0}^{2}\right)}{\lambda_{10}}-\frac{\sigma^{2}}{\left(1+\lambda_{10}\right)}-\frac{\left(\sigma^{2}+\hat{\sigma}_{2\_VTS,0}^{2}\right)}{\left(1+\lambda_{10}\right)}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]\left(\frac{1}{\lambda_{10}}\mathbf{I}-\frac{\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}}{\lambda_{10}^{2}\left(1+\lambda_{10}\right)}\right)$$

(4.102)

$$=\left[\sigma^{2}\mathbf{I}+\left(\frac{\left(1+\lambda_{10}\right)\left(\sigma^{2}+\hat{\sigma}_{2\_VTS,0}^{2}\right)-\lambda_{10}\sigma^{2}-\left(\sigma^{2}+\hat{\sigma}_{2\_VTS,0}^{2}\right)}{\lambda_{10}\left(1+\lambda_{10}\right)}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]\left(\frac{1}{\lambda_{10}}\mathbf{I}-\frac{\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}}{\lambda_{10}^{2}\left(1+\lambda_{10}\right)}\right)$$

(4.103)

$$=\left[\sigma^{2}\mathbf{I}+\frac{\hat{\sigma}_{2\_VTS,0}^{2}}{\left(1+\lambda_{10}\right)}\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}\right]\left(\frac{1}{\lambda_{10}}\mathbf{I}-\frac{\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^{T}}{\lambda_{10}^{2}\left(1+\lambda_{10}\right)}\right)\qquad(4.104)$$

$$\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\left(\frac{\sigma^2}{\lambda_{10}\left(1+\lambda_{10}\right)}-\frac{\hat{\sigma}^2_{2\_VTS,0}}{\lambda_{10}\left(1+\lambda_{10}\right)}+\frac{\hat{\sigma}^2_{2\_VTS,0}}{\lambda_{10}\left(1+\lambda_{10}\right)^2}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^T \qquad (4.105)$$

$$\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\left(\frac{\sigma^2\left(1+\lambda_{10}\right)-\lambda_{10}\hat{\sigma}^2_{2\_VTS,0}}{\lambda_{10}\left(1+\lambda_{10}\right)^2}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^T \qquad (4.106)$$

Next, the MSE at 1st global iteration is

$$\sigma^2_{1\_DTS,1}=\text{trace}\left\{E\left[\text{cov}\left[\mathbf{e}_{1\_DTS,1}|\hat{\mathbf{x}}_{2,0}\right]\right]\right\} \qquad (4.107)$$

$$=E\left\{\text{trace}\left[\text{cov}\left[\mathbf{e}_{1\_DTS,1}|\hat{\mathbf{x}}_{2,0}\right]\right]\right\} \qquad (4.108)$$

Then, the MSE of estimation for mobile *1* based on (4.108) can be written as follows

$$\sigma^2_{1\_DTS,1}=E\left\{\text{trace}\left[\frac{\sigma^2}{\lambda_{10}}\mathbf{I}-\left(\frac{\sigma^2\left(1+\lambda_{10}\right)-\lambda_{10}\hat{\sigma}^2_{2\_VTS,0}}{\lambda_{10}\left(1+\lambda_{10}\right)^2}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^T\right]\right\} \qquad (4.109)$$

$$=E\left\{\text{trace}\left(\frac{\sigma^2}{\lambda_{10}}\mathbf{I}\right)-\text{trace}\left[\left(\frac{\sigma^2\left(1+\lambda_{10}\right)-\lambda_{10}\hat{\sigma}^2_{2\_VTS,0}}{\lambda_{10}\left(1+\lambda_{10}\right)^2}\right)\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^T\right]\right\} \qquad (4.110)$$

$$=\text{trace}\left(\frac{\sigma^2}{\lambda_{10}}\mathbf{I}\right)-E\left\{\left(\frac{\sigma^2\left(1+\lambda_{10}\right)-\lambda_{10}\hat{\sigma}^2_{2\_VTS,0}}{\lambda_{10}\left(1+\lambda_{10}\right)^2}\right)\text{trace}\left[\mathbf{a}_{10,\hat{2},0}\mathbf{a}_{10,\hat{2},0}{}^T\right]\right\} \qquad (4.111)$$

$$=\frac{2\sigma^2}{\lambda_{10}}-\left(\frac{\sigma^2\left(1+\lambda_{10}\right)-\lambda_{10}\hat{\sigma}^2_{2\_VTS,0}}{\lambda_{10}\left(1+\lambda_{10}\right)^2}\right) \qquad (4.112)$$

$$=\frac{2\sigma^2}{\lambda_{10}}-\frac{\sigma^2}{\lambda_{10}\left(1+\lambda_{10}\right)}+\frac{\hat{\sigma}^2_{2\_VTS,0}}{\left(1+\lambda_{10}\right)^2} \qquad (4.113)$$

The MSE of estimation for mobile *2* has the similar result

$$\sigma^2_{2\_DTS,1}=\frac{2\sigma^2}{\lambda_{20}}-\frac{\sigma^2}{\lambda_{20}\left(1+\lambda_{20}\right)}+\frac{\hat{\sigma}^2_{1\_VTS,0}}{\left(1+\lambda_{20}\right)^2} \qquad (4.114)$$

As before, $2\sigma^2/\lambda_{10}=\hat{\sigma}^2_{1\_VTS,0}=\sigma^2_{1\_TS}$ and $2\sigma^2/\lambda_{20}=\hat{\sigma}^2_{2\_VTS,0}=\sigma^2_{2\_TS}$, they are the uncooperative MSE. Therefore, the MSE at *2*nd global iteration is denoted as

$$\sigma^2_{1\_DTS,2}=\sigma^2_{1\_TS}-\frac{\sigma^2}{\lambda_{10}\left(1+\lambda_{10}\right)}+\frac{\hat{\sigma}^2_{2\_VTS,1}}{\left(1+\lambda_{10}\right)^2} \qquad (4.115)$$

$$= \sigma_{1\_TS}^2 - \frac{\sigma^2}{\lambda_{10}(1+\lambda_{10})} + \frac{1}{(1+\lambda_{10})^2}\left(\sigma_{2\_TS}^2 - \frac{\sigma^2}{\lambda_{20}(1+\lambda_{20})} + \frac{\sigma_{1\_TS}^2}{(1+\lambda_{20})^2}\right) \quad (4.116)$$

$$= \sigma_{1\_TS}^2 - \frac{\sigma^2}{\lambda_{10}(1+\lambda_{10})} + \frac{\sigma_{2\_TS}^2}{(1+\lambda_{10})^2}$$

$$+ \frac{1}{(1+\lambda_{10})^2}\left(-\frac{\sigma^2}{\lambda_{20}(1+\lambda_{20})} + \frac{\sigma_{1\_TS}^2}{(1+\lambda_{20})^2}\right) \quad (4.117)$$

and the MSE at 3rd global iteration is

$$\sigma_{1\_DTS,3}^2 = \sigma_{1\_TS}^2 - \frac{\sigma^2}{\lambda_{10}(1+\lambda_{10})} + \frac{\hat{\sigma}_{2\_VTS,2}^2}{(1+\lambda_{10})^2} \quad (4.118)$$

$$= \sigma_{1\_TS}^2 - \frac{\sigma^2}{\lambda_{10}(1+\lambda_{10})} + \frac{1}{(1+\lambda_{10})^2}\left(\frac{2\sigma^2}{\lambda_{20}} - \frac{\sigma^2}{\lambda_{20}(1+\lambda_{20})} + \frac{\sigma_{1\_TS}^2}{(1+\lambda_{20})^2} + \frac{1}{(1+\lambda_{20})^2}\left(-\frac{\sigma^2}{\lambda_{10}(1+\lambda_{10})} + \frac{\sigma_{2\_TS}^2}{(1+\lambda_{10})^2}\right)\right) \quad (4.119)$$

$$= \sigma_{1\_TS}^2 - \frac{\sigma^2}{\lambda_{10}(1+\lambda_{10})} + \frac{\sigma_{2\_TS}^2}{(1+\lambda_{10})^2}$$

$$+ \frac{1}{(1+\lambda_{10})^2}\left(-\frac{\sigma^2}{\lambda_{20}(1+\lambda_{20})} + \frac{\sigma_{1\_TS}^2}{(1+\lambda_{20})^2}\right)$$

$$+ \frac{1}{(1+\lambda_{20})^4}\left(-\frac{\sigma^2}{\lambda_{10}(1+\lambda_{10})} + \frac{\sigma_{2\_TS}^2}{(1+\lambda_{10})^2}\right) \quad (4.120)$$

We have cooperative benefit of mobile *1* is

$$b_{10} = -\frac{\sigma^2}{\lambda_{10}(1+\lambda_{10})} + \frac{\hat{\sigma}_{2\_VTS,0}^2}{(1+\lambda_{10})^2} < 0 \quad (4.121)$$

$$b_{20} = -\frac{\sigma^2}{\lambda_{20}(1+\lambda_{20})} + \frac{\hat{\sigma}_{1\_VTS,0}^2}{(1+\lambda_{20})^2} < 0 \quad (4.122)$$

Then, the converged MSE closed form of mobile *1* is denoted as

$$\sigma_{1\_DTS,n}^2 = \sigma_{1\_TS}^2 + \sum_{\substack{k=1 \\ k=odd}}^{n} \frac{b_{10}}{(1+\lambda_{20})^{2(k-1)}} + \sum_{\substack{k=1 \\ k=even}}^{n} \frac{b_{20}}{(1+\lambda_{10})^{2(k-1)}} \quad (4.123)$$

Then, the converged MSE of unweighted divided TS estimator is calculated as

$$\sigma_{1\_DTS}^2 = \sigma_{1\_TS}^2 + \frac{\left(1+\lambda_{20}\right)^2}{2+\lambda_{20}} b_{10} + \frac{1+\lambda_{10}}{2+\lambda_{10}} b_{20} \qquad (4.124)$$

The converged MSE for mobile *2* can be obtained similarly.

The computer simulation will shown the derivation result in Section 5.4

### 4.2.2 Distance-Augmented Method

In this section, we discuss that the converged MSE for estimation of divided distance-augmented (DDA) algorithm in (3.43). As previous assumption, all measurement error variance are the same as $\sigma^2$. Without loss of generality, let *i*=1, the error covariance of DDA estimation, $\mathbf{e}_{1\_DDA,n+1} = \hat{\mathbf{x}}_{1\_DDA,n+1} - \mathbf{x}_1$ for mobile *1* at *1*st global iteration, conditioned on positions of virtual sensors *2* to *M*, $\hat{\mathbf{x}}_{2,n}, \hat{\mathbf{x}}_{3,n}, \cdots, \hat{\mathbf{x}}_{M,n}$ is denoted as,

$$\mathrm{cov}\left(\mathbf{e}_{1\_DDA,n+1} \mid \hat{\mathbf{x}}_{2,n}, \hat{\mathbf{x}}_{3,n}, \cdots, \hat{\mathbf{x}}_{M,n}\right) = \left(\mathbf{H}_{1\_DDA,n}{}^T \mathbf{W}_{1\_DDA,n} \mathbf{H}_{1\_DDA,n}\right)^{-1} \qquad (4.125)$$

$$= \left[ \mathbf{H}_{1\_DA}{}^T \mathbf{W}_{1\_DA} \mathbf{H}_{1\_DA} + \sum_{j=2}^{M} \frac{1}{4r_{1j}^2\left(\sigma^2 + \hat{\sigma}_{j,n}^2\right) + 3\left(\sigma^2 + \hat{\sigma}_{j,n}^2\right)} \hat{\mathbf{c}}_{j,n} \hat{\mathbf{c}}_{j,n}{}^T \right]_{3\times 3}^{-1} \qquad (4.126)$$

where $\mathbf{H}_{1\_DDA,n}$ is divided coordinate matrix in (3.43), $\mathbf{W}_{1\_DDA,n}$ is divided DA weighting in (3.45), $\mathbf{H}_{1\_DA}{}^T \mathbf{W}_{1\_DA} \mathbf{H}_{1\_DA}$ is error covariance matrix inverse of uncooperative weighted DA estimation in (2.48) and $\hat{\mathbf{c}}_{j,n} = \begin{bmatrix} 2\hat{x}_{j,n} & 2\hat{y}_{j,n} & -1 \end{bmatrix}^T$ is a virtual coordinate vector. After unconditioning $\hat{\mathbf{x}}_{2,n}, \hat{\mathbf{x}}_{3,n}, \cdots, \hat{\mathbf{x}}_{M,n}$, the theoretical MSE for mobile *1* at $n+1^{\text{th}}$ global iteration is given by

$$\hat{\sigma}_{i\_DDA,n+1}^2 = \mathrm{trace}\left\{ E\left[ \mathrm{cov}\left(\mathbf{e}_{1\_DDA,n+1} \mid \hat{\mathbf{x}}_{2,n}, \hat{\mathbf{x}}_{3,n}, \cdots, \hat{\mathbf{x}}_{M,n}\right)\right]\right\}_{2\times 2} \qquad (4.127)$$

(4.127) is quite difficult to analyze. The main course is that the error covariance

matrix of divided DA estimation is $3\times 3$, however the MSE is obtain by take trace on error covariance with size $2\times 2$. As before, another reason is that the statistic of virtual coordinate vector is very complex.

### 4.2.3 Hyperbolic Positioning Algorithm

As before, we discuss the error covariance of divided HC estimation, $\mathbf{e}_{1\_DHC,n+1}=\hat{\mathbf{x}}_{1\_DHC,n+1}-\mathbf{x}_1$ for mobile *1* at $1^{st}$ global iteration, conditioned on positions of virtual sensors *2* to *M*, $\hat{\mathbf{x}}_{2,n},\hat{\mathbf{x}}_{3,n},\cdots,\hat{\mathbf{x}}_{M,n}$ is denoted as,

$$\mathrm{cov}\left(\mathbf{e}_{1\_DHC,n+1}\,|\,\hat{\mathbf{x}}_{2,n},\hat{\mathbf{x}}_{3,n},\cdots,\hat{\mathbf{x}}_{M,n}\right)=\left(\mathbf{H}_{1\_DHC,n}{}^{T}\mathbf{W}_{1\_DHC,n}\mathbf{H}_{1\_DHC,n}\right)^{-1} \quad (4.128)$$

where $\mathbf{H}_{i\_DHC,n}$ is divided difference-coordinate matrix in (3.29) and $\mathbf{W}_{i\_DHC,n}$ is divided CH weighting matrix in (3.37).

After unconditioning $\hat{\mathbf{x}}_{2,n},\hat{\mathbf{x}}_{3,n},\cdots,\hat{\mathbf{x}}_{M,n}$, the theoretical MSE for mobile *1* at $n+1^{th}$ global iteration is given by

$$\hat{\sigma}^2_{i\_DHC,n+1}=\mathrm{trace}\left\{E\left[\mathrm{cov}\left(\mathbf{e}_{1\_DHC,n+1}\,|\,\hat{\mathbf{x}}_{2,n},\hat{\mathbf{x}}_{3,n},\cdots,\hat{\mathbf{x}}_{M,n}\right)\right]\right\} \quad (4.129)$$

We know that the divided CH weighting matrix is not a diagonal anymore, then the error covariance is very to analyze.

From AEFIM, we can discover that more mobiles will enhance the localization accuracy. Then, we further derive AC-CRLB. The numbers of sensors, of mobiles, measurement error variance, and cooperative angle will affect the theoretical cooperative MSE performance. Next, we try to analyze the converged MSE of divided linearized algorithm estimations. However, most of them are very harsh to evaluate the theoretical MSE but we still demonstrate the converged MSE of divided TS estimation successfully for special case *M=2*. Although we have on the theoretical

MSE of some divided algorithms, we still use computer simulation to show their MSE performance.

# Chapter 5
# Computer Simulations

Now, in uncooperative localization system, we will study the theoretical MSE performance of three linearized algorithms. Numerical examples are presented in Section 5.1 to evaluate the performance of these algorithms by comparing nonlinearized algorithm and CRLB. However, in unweighted linearized algorithms, TS has the best MSE performance and DA has the worst MSE. After the compensation of weighting, the MSE performance of DA and HC will be close to TS. That means the weighting matrix can improve the localization accuracy.

Next, in cooperative system, the ML estimator can be solved by joint Newton's algorithm (3.7). But it cost too much when many mobiles exist. Therefore, we propose joint TS (3.20) and divided-and-conquer method to reduce the computer cost. However, besides divided Newton's algorithm (3.26), we also can utilize TS, DA and HC to perform divide-and-conquer method (3.36), (3.43) and (3.50). In Sections 5.2 and 5.3, the computer simulation will show these two new algorithms can reduce the computer cost and the MSE performances are still very good. Beside, we know that the CRLB provides a useful means for the analysis of the limits of localization accuracy. But it is obtained by full cooperative FIM (4.6). However, the cooperative FIM is quite complicate. Therefore, we derive AEFIM (4.54) and AC-CRLB (4.64). From AC-CRLB, we can see that the number of mobiles and cooperative angles can affect the MSE performance. Computer simulations are included to contrast the performance of the derivations with true CRLB. Finially, we also evaluate the theoretical converged MSE of divided TS algorithm for tow mobiles.

## 5.1 The Theoretical MSE of Three Uncooperative Linearized Algorithms

We compare the theoretical MSE of three uncooperative linearized algorithms to traditional CRLB. Note that for an uncooperative localization system, the additive noises $\{n_{i\bar{j}}\}$ are zero-mean white Gaussian processes and their variance are the same $\sigma^2$. There are 15 unknown positions of mobiles. We focus on the localization performance at central mobile (10, 10) and other mobiles are uniformly located in a 20m $\times$ 20m area with 4 known positions of sensors (0,0)m, (0,20), (20,0)m and (20,20)m. The localization system is shown in Figure 5.1. All simulation results are averages of 100 independent runs.

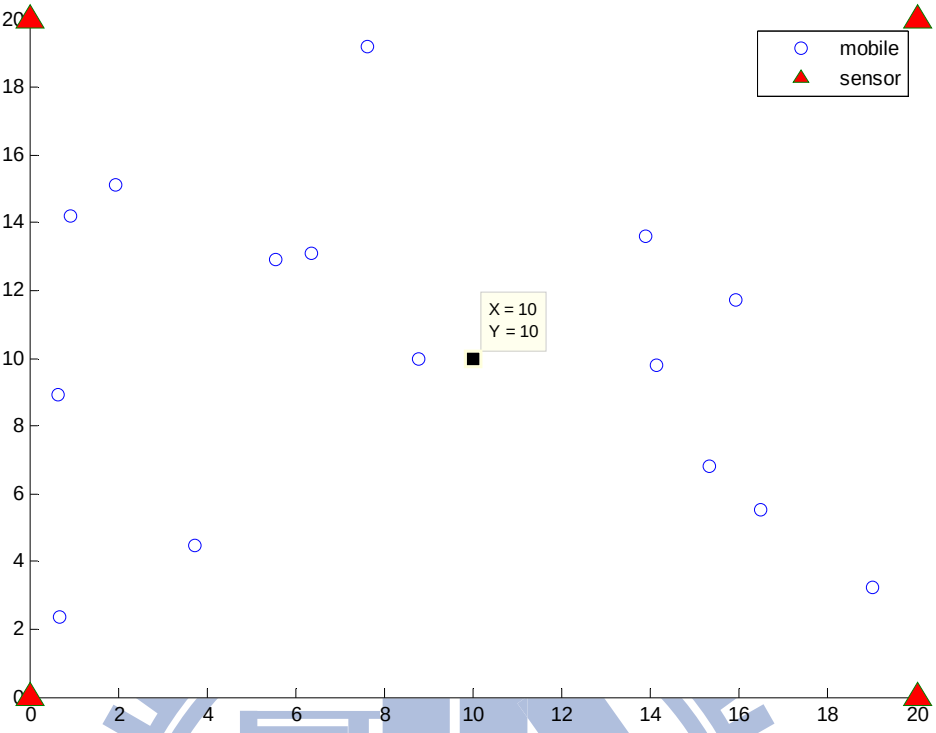



Figure 5.1 Positions of the sensors and mobiles.

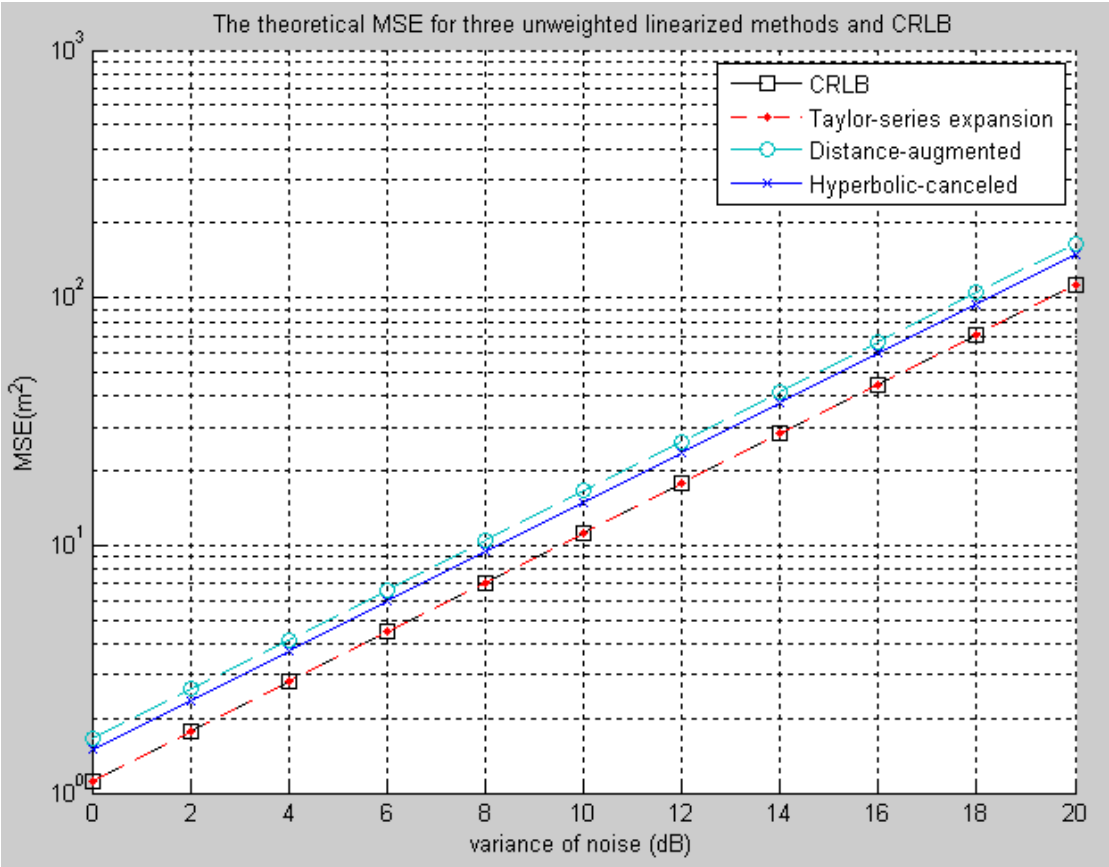



Figure 5.2 The theoretical MSE versus noise variance for three unweighted linearized methods.

The the theoretical MSE of three unweighted liearized methods and CRLB are plotted in Figure 5.2. It can be observed that distance-augmented method has the worst MSE performance because its noise source is expanded by real distance and the augmented distance variable is not independent on true position. The MSE of hyperbolic-canceled method is better than distance-augmented method but worse than Taylor-series expansion method because it is only involve in noise source effect which is like distance-augmented method's. However, Taylor-series expansion method has the best theoretical MSE because its noise source is not expanded by real distance and it is very close to CRLB.

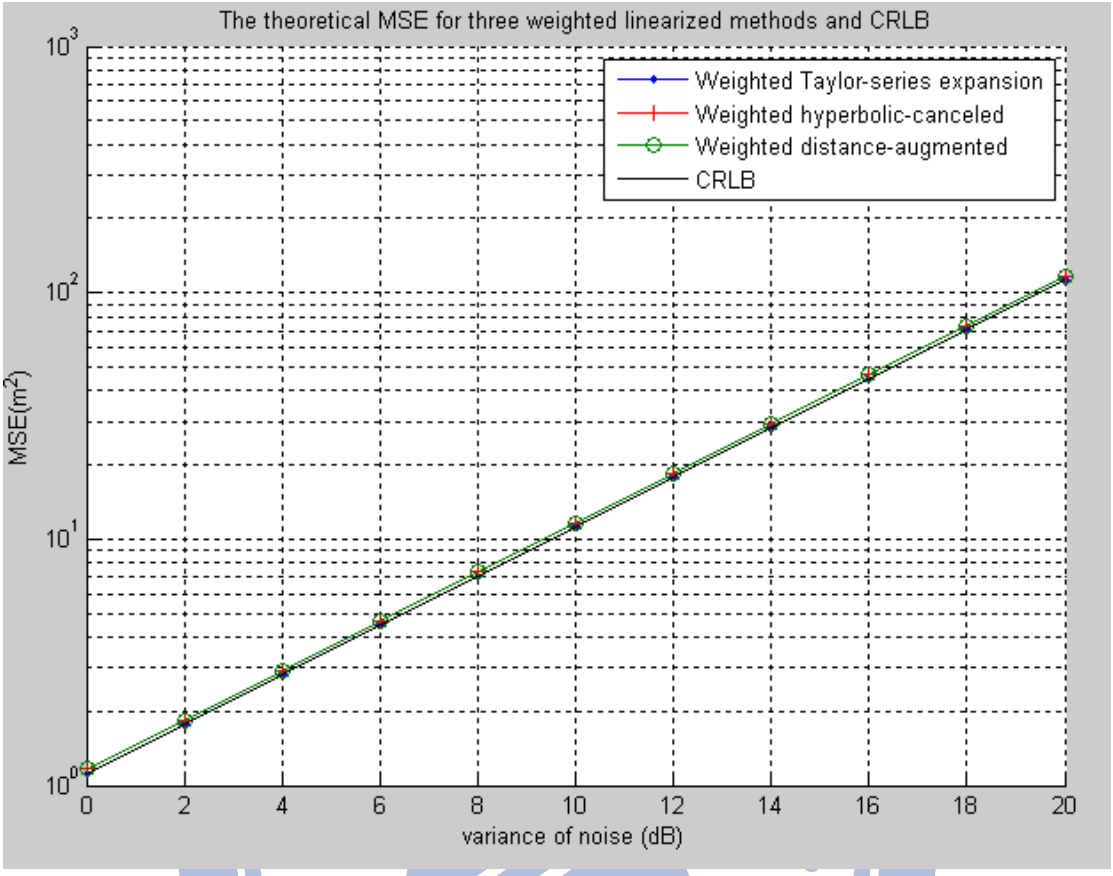



Figure 5.3 The theoretical MSE versus noise variance for three weighted linearized methods and CRLB.

The the theoretical MSE of three weighted liearized methods and CRLB are plotted in Figure 5.3. After being weighted, the MSE performance of hyperbolic-canceled method and distance-augmented method are improved but still worse than Taylor-series expansion method. Then, Taylor-series expansion's MSE maintain the same because we assume the noise variance are the same and the Taylor-series weighting becomes $\mathbf{W}_{j\_TS} = \frac{1}{\sigma^2}\mathbf{I}_{N\times N}$. However, the localization error covariance for weighted matrix and unweighted matrix are the same.

$$\begin{aligned}\operatorname{cov}\left(\hat{\mathbf{x}}_{j\_WTS}-\mathbf{x}\right)&=\left(\mathbf{H}_{j\_TS}{}^{T}\mathbf{W}_{j\_TS}\mathbf{H}_{j\_TS}\right)^{-1}\\&=\left(\mathbf{H}_{j\_TS}{}^{T}\frac{1}{\sigma^{2}}\mathbf{I}_{N\times N}\mathbf{H}_{j\_TS}\right)^{-1}\\&=\sigma^{2}\left(\mathbf{H}_{j\_TS}{}^{T}\mathbf{H}_{j\_TS}\right)^{-1}\end{aligned}$$

$$\begin{aligned}\operatorname{cov}\left(\hat{\mathbf{x}}_{j\_TS}-\mathbf{x}\right)&=\left(\mathbf{H}_{j-TS}{}^{T}\mathbf{H}_{j-TS}\right)^{-1}\mathbf{H}_{j-TS}{}^{T}\mathbf{W}_{j-TS}{}^{-1}\mathbf{H}_{j-TS}\left(\mathbf{H}_{j-TS}{}^{T}\mathbf{H}_{j-TS}\right)^{-1}\\&=\left(\mathbf{H}_{j-TS}{}^{T}\mathbf{H}_{j-TS}\right)^{-1}\mathbf{H}_{j-TS}{}^{T}\sigma^{2}\mathbf{I}_{N\times N}\mathbf{H}_{j-TS}\left(\mathbf{H}_{j-TS}{}^{T}\mathbf{H}_{j-TS}\right)^{-1}\\&=\sigma^{2}\left(\mathbf{H}_{j-TS}{}^{T}\mathbf{H}_{j-TS}\right)^{-1}\end{aligned}$$

The simulation will demonstrate the theoretical MSE in Figure 5.4.

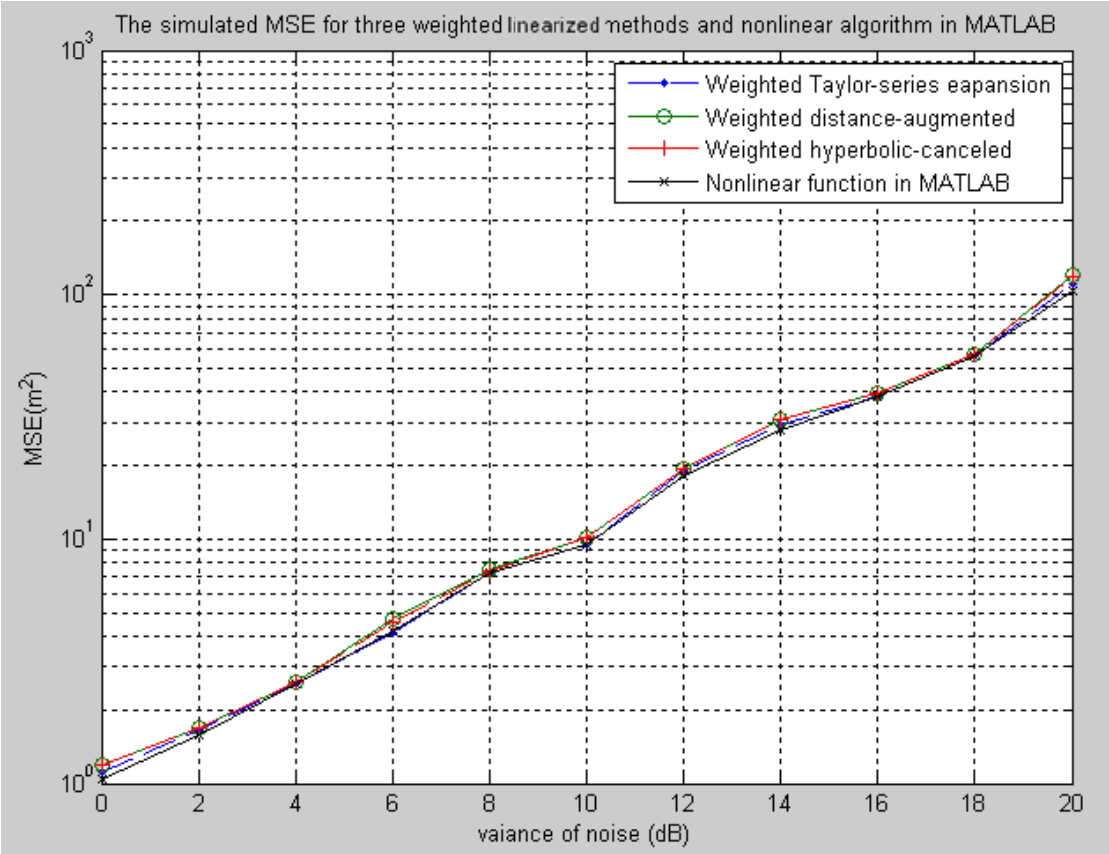



Figure 5.4 The simulated MSE versus noise variance for three weighted linearized methods and nonlinear algorithm in MATLAB.

The simulated MSE of three uncooperative liearized methods and one nonlinear algorithm MATLAB function to solve ML estimator (2.11) are plotted in Figure 5.4. As theoretical MSE, the simulated MSE performance of distance-augmented method and hyperbolic-canceled method are very close but worse than Taylor-series expansion. However, the nonlinear function in MATLAB is fminunc () and the Taylor-series expansion's simulated MSE is quit close to each other.

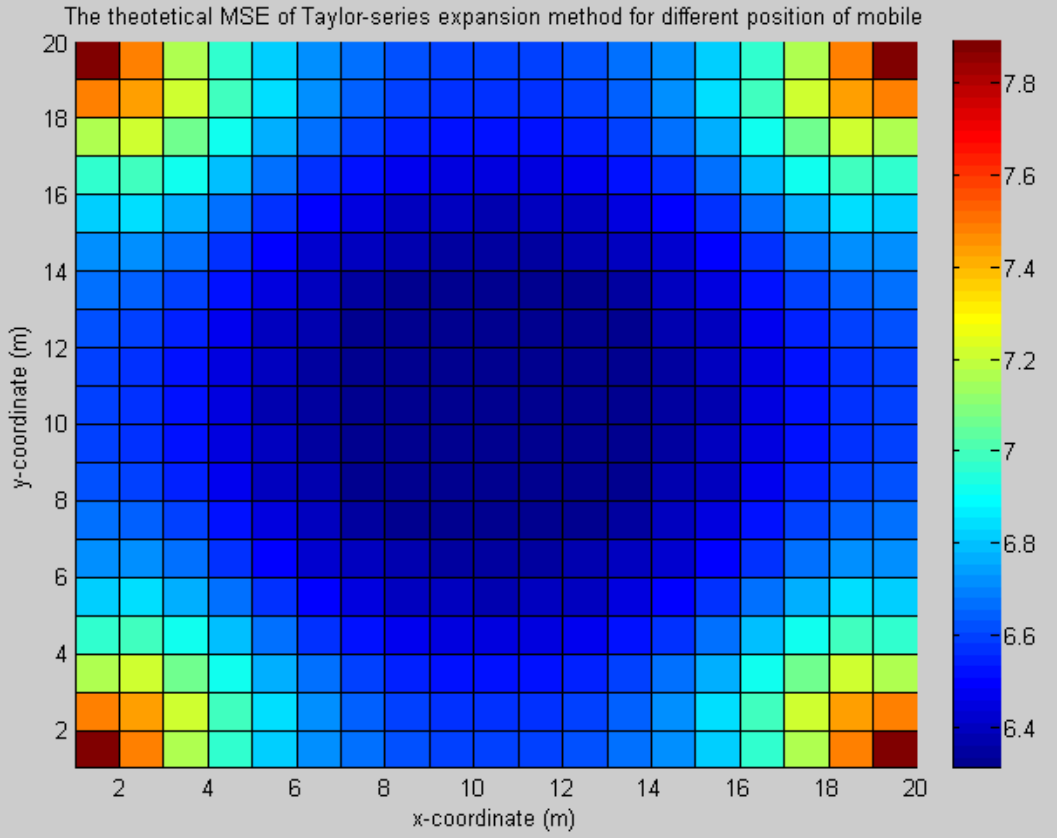



Figure 5.5 The theoretical MSE of Taylor-series expansion method for different position of mobile.

The theoretical MSE of Taylor-series expansion method for different position of mobile is plotted in Figure 5.5. The noise variance is 8 dB. However, the center has the best MSE performance and the four corners have the worst performance.

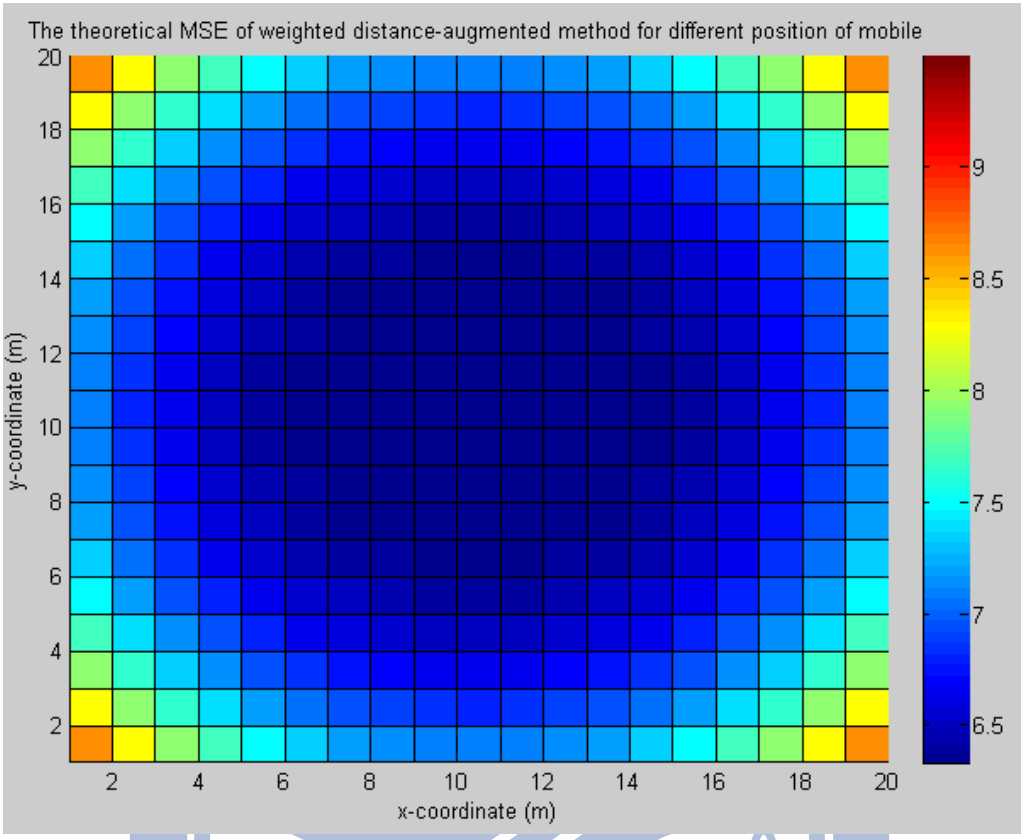



Figure 5.6 The theoretical MSE of weighted distance-augmented algorithm for different position of mobile.

The theoretical MSE of weighted distance-augmented method for different position of mobile is shown in Figure 5.6. It has the same result which is like the MSE of Taylor-series expansion method. Moreover, the all MSE of any place are worse than Taylor-series expansion method.

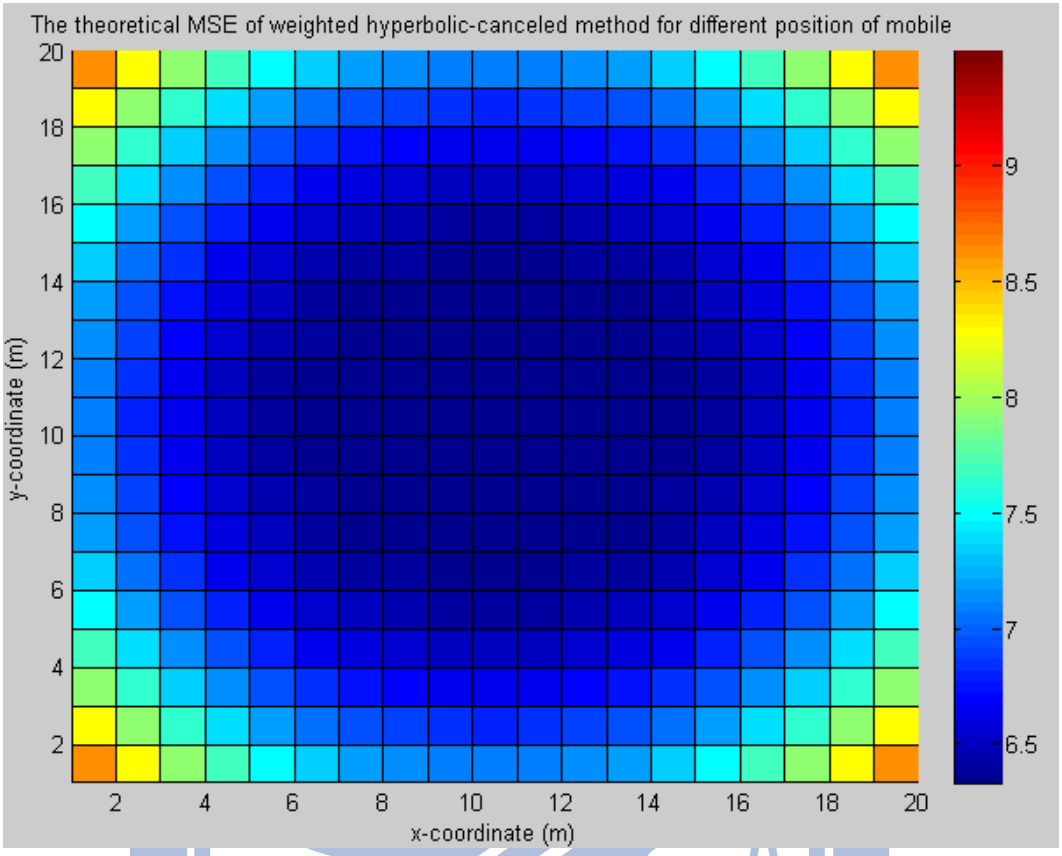
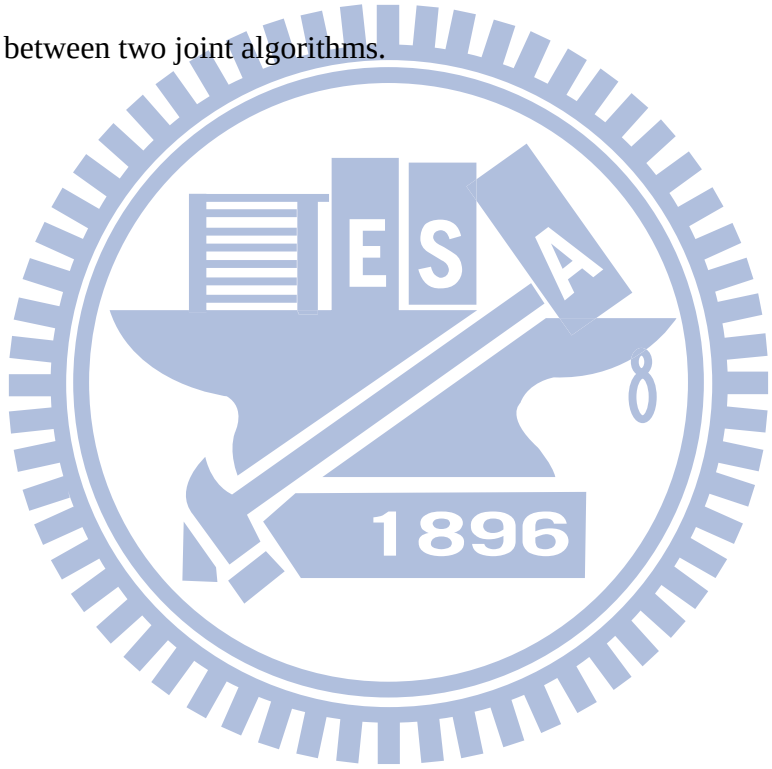



Figure 5.7 The theoretical MSE of weighted hyperbolic-canceled method for different position of mobile.

The theoretical MSE of weighted hyperbolic-canceled method for different position of mobile is shown in Figure 5.7. The MSE is similar to distance-augmented method's.

## 5.2 Comparison of Joint Newton's Algorithm and Joint Taylor-Series Expansion Algorithm

The simulation scenario and assumption of noise are designed such as Figure 5.1 In this section, we present some simulations for two joint algorithms; joint Taylor-series expansion algorithm and joint Newton's algorithm which are described in the previous section. We consider the convergence rate, computation cost and MSE performance between two joint algorithms.



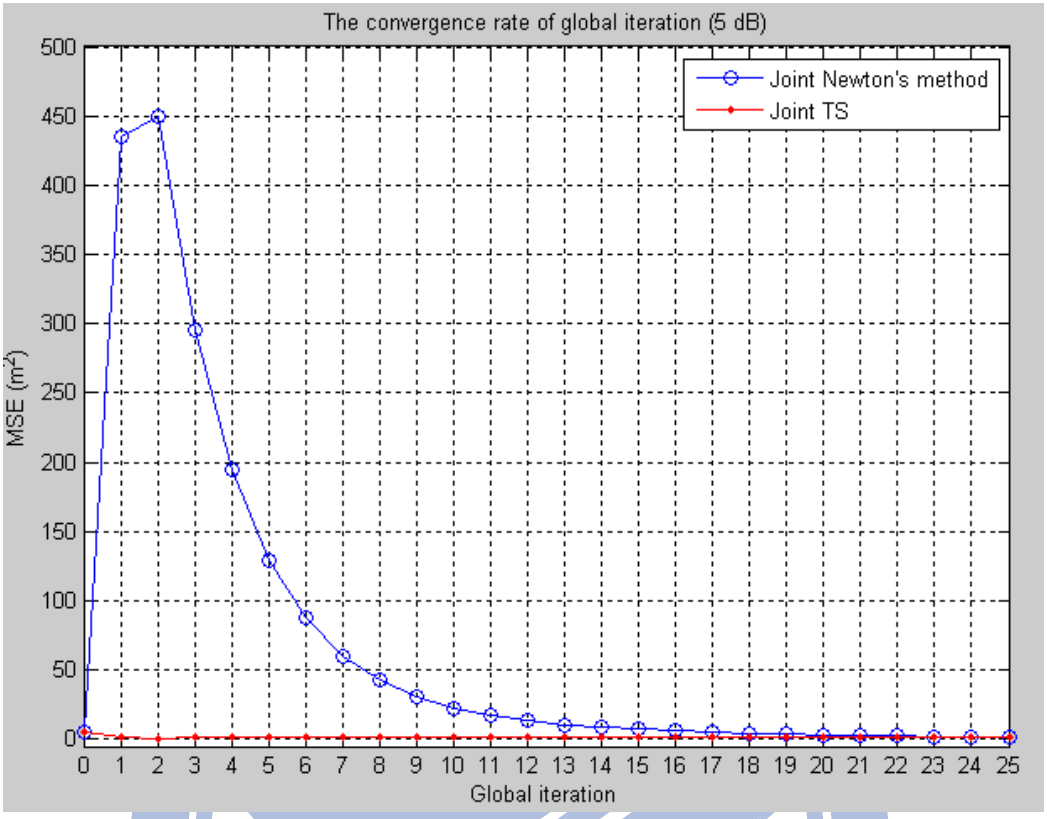



Figure 5.8 The MSE vs. convergence rate for joint Newton's method and joint TS with good initial guess set.

Figure 5.8 shows the MSE vs. convergence rate for joint Newton's method in (3.7) and joint TS in (3.20) with good initial guess set. ($\sigma^2_{\text{initial\_guess}} = 3$). We can see that joint TS convergence rate is faster than joint Newton's method. Joint TS needs an average of three iterations to converge but joint Newton's method needs more than about twenty global iterations to converge. In our simulations, Newton's method may not converge and oscillation can happen occasionally. In the end of the iteration, these two methods have about the same MSE performance.

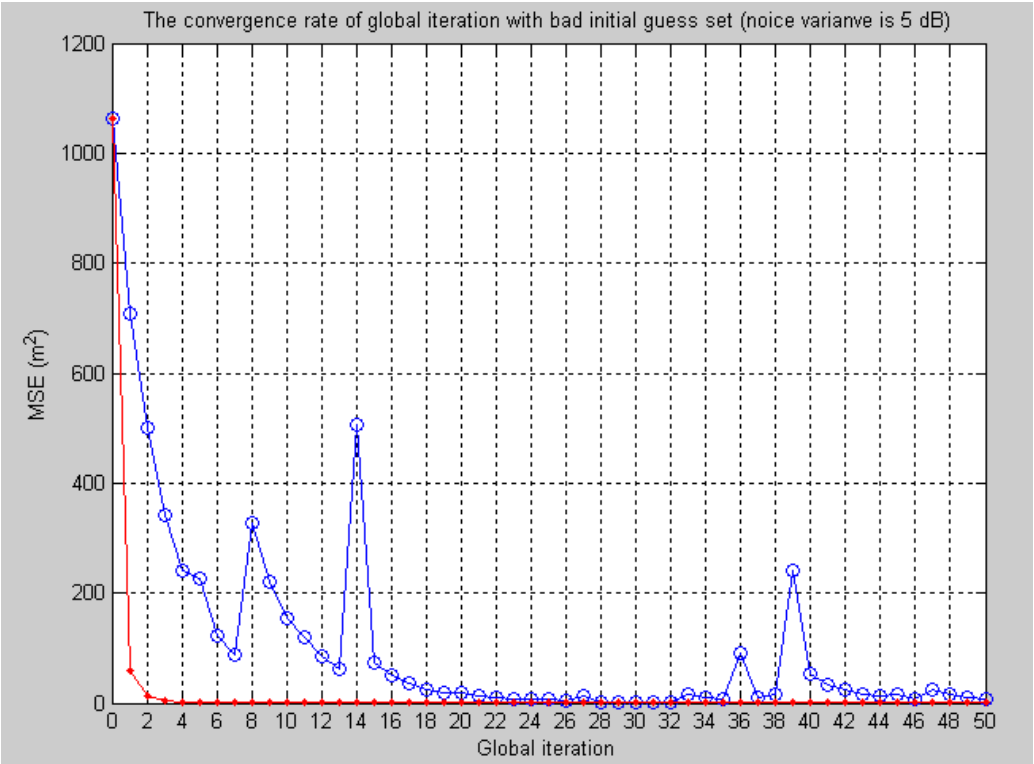



Figure 5.9 The MSE vs. convergence rate for joint Newton's method and joint TS with bad initial guess set.

The MSE vs. convergence rate for joint Newton's method and joint TS with bad initial guess set. is plotted in Figure 5.9. Obviously, the convergence rate of joint TS is still faster than joint Newton. Unfortunately, the joint Newton's method have oscillation happened. Therefore, the joint Newton's method is no guarantee of global solution.

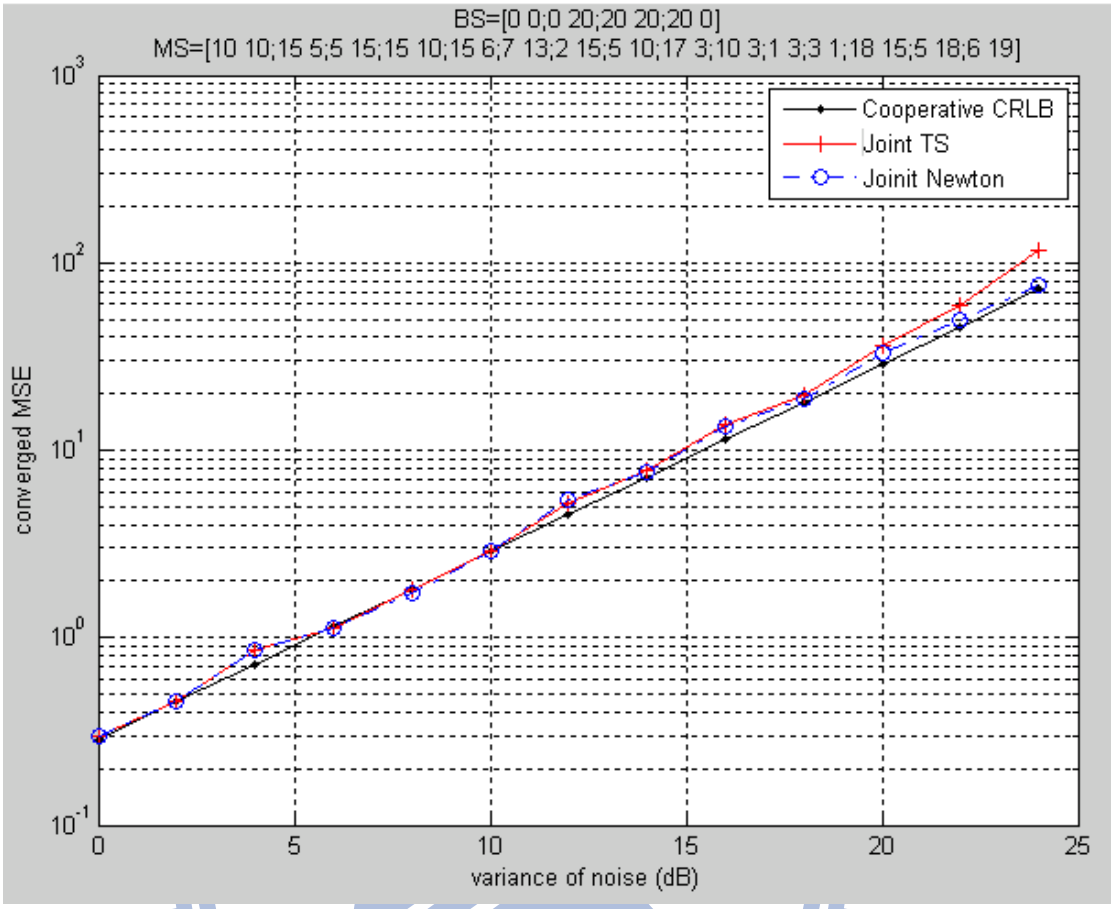



Figure 5.10 The converged MSE vs. The variance of noise.

Figure 5.10 shows the converged MSE vs. the variance of noise for joint TS and joint Newton's method and cooperative CRLB. Under low variance of noise, the convergence MSE of Joint TS and joint Newton's method are closed to cooperative CRLB (is obtained form (4.6)). But in higher variance of noise (up to about 15 dB), the MSE performance of joint TS is the worst but joint Newton's method is still close to cooperative CRLB.

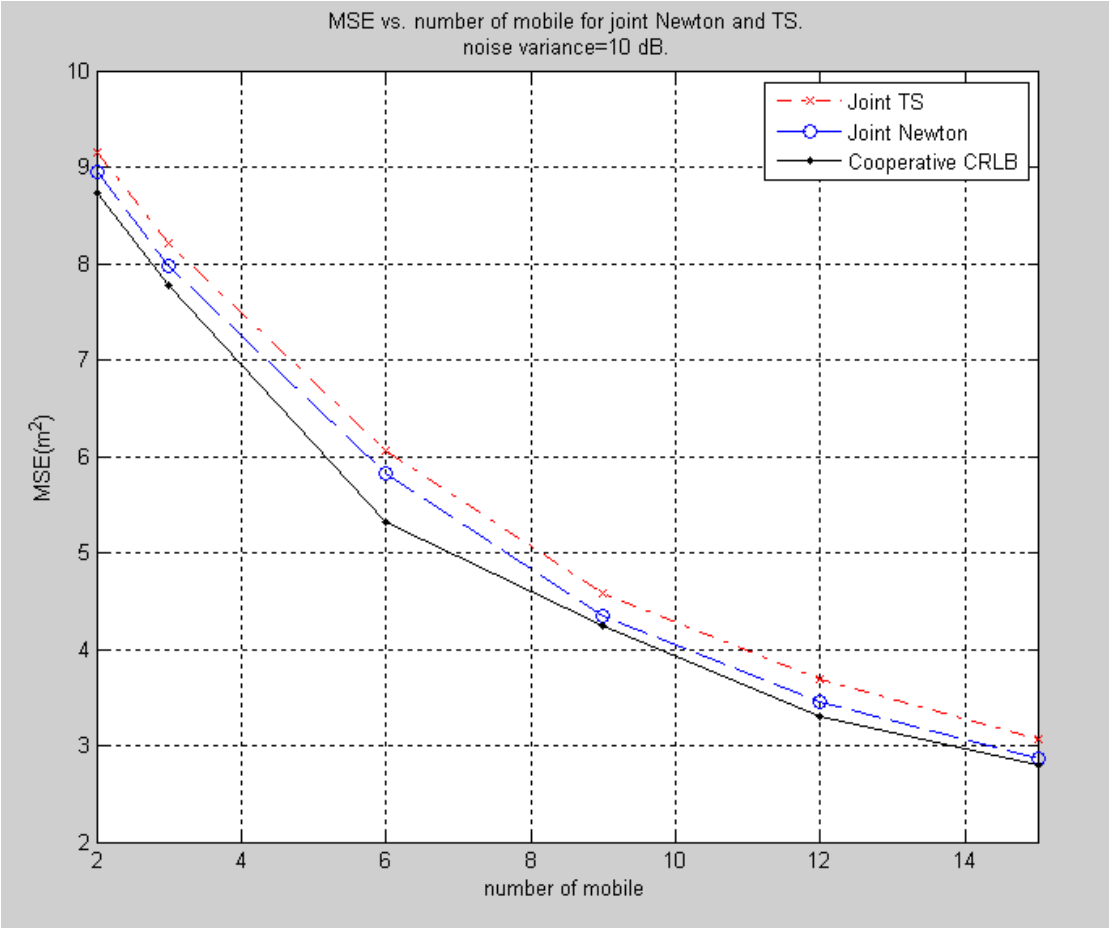



Figure 5.11 MSE vs. the number of mobile for joint Newton's and TS.

MSE vs. the number of mobile for joint Newton's and TS are plotted in Figure 5.11. We can see the localization accuracy of joint Newton's method is better than joint TS. However, the MSE of joint TS is still quite good.

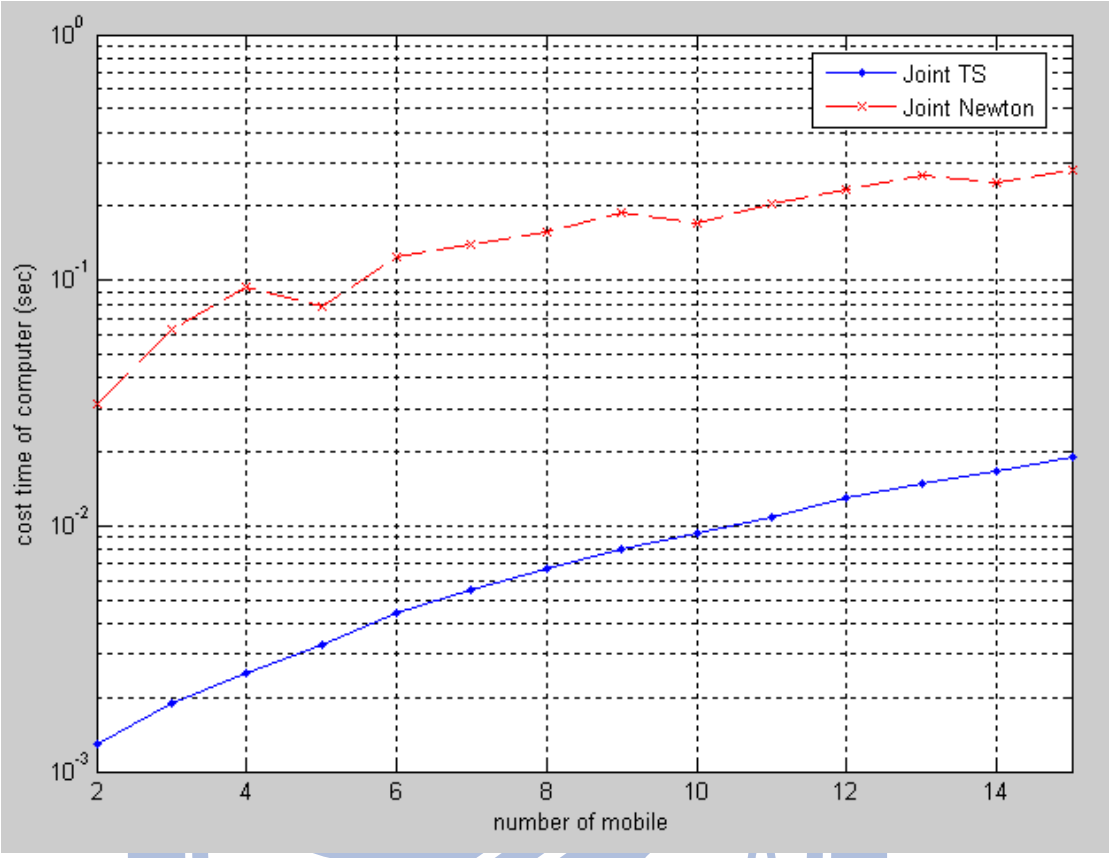



Figure 5.12 The computation cost vs. the number of mobile for joint TS and joint Newton on once global iteration.

Figure 5.12 shows the computation cost vs. number of mobile for joint TS and joint Newton. The computer equivalent we use as follows,

System: Microsoft Windows XP Professional Version 2002 Service Pack 3.

Hardware: AMD phenom (tm) II X3 710 Processor 2.61 GHz, 3.25 GB RAM.

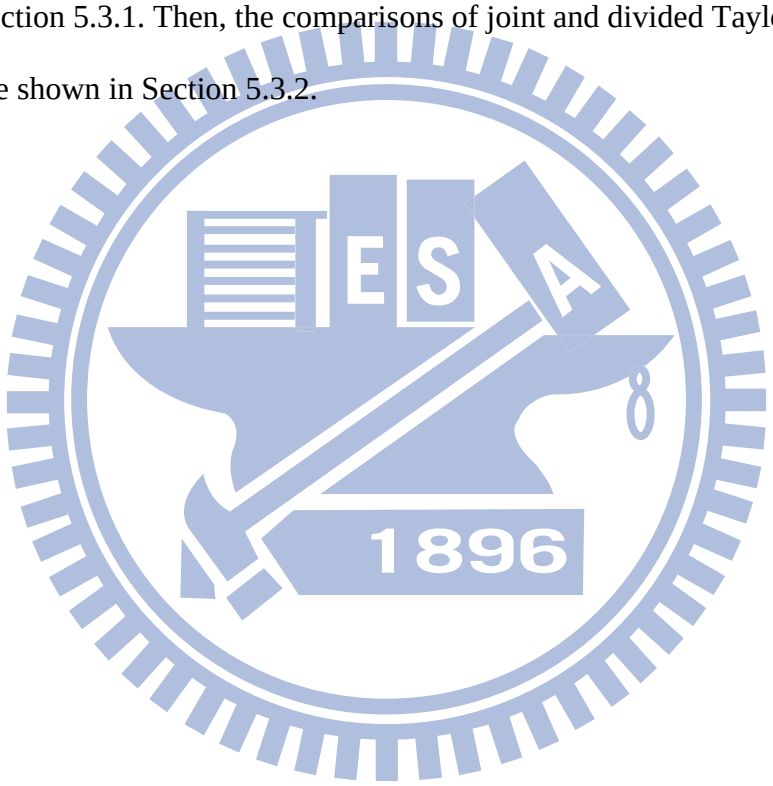
Software: MATLAB Version 7.5.0.342 (R2007b).

We compare the cost time of computer at *1*st global iteration between joint Newton's method (3.7) and joint Newton's method (3.20). However, the computation of matrix inverses (3.7) and (3.20) are replaced Gaussian elimination method, $\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{Hb}$ is replaced by $\mathbf{H}\backslash\mathbf{b}$ in MATLAB function. Obviously, the joint TS spends less

computation cost than joint Newton's method. Summarize the joint TS from figures 5.8 to 5.11, it can perform a quite good MSE and low computation cost. Therefore, joint TS is a very good algorithm for cooperative localization.

## 5.3 Comparison of Joint Algorithms and Divide-and-Conquer Algorithms

We compare the MSE performance and cost for joint and divided Newton's method in Section 5.3.1. Then, the comparisons of joint and divided Taylor-series expansion are shown in Section 5.3.2.



### 5.3.1 Newton's Method

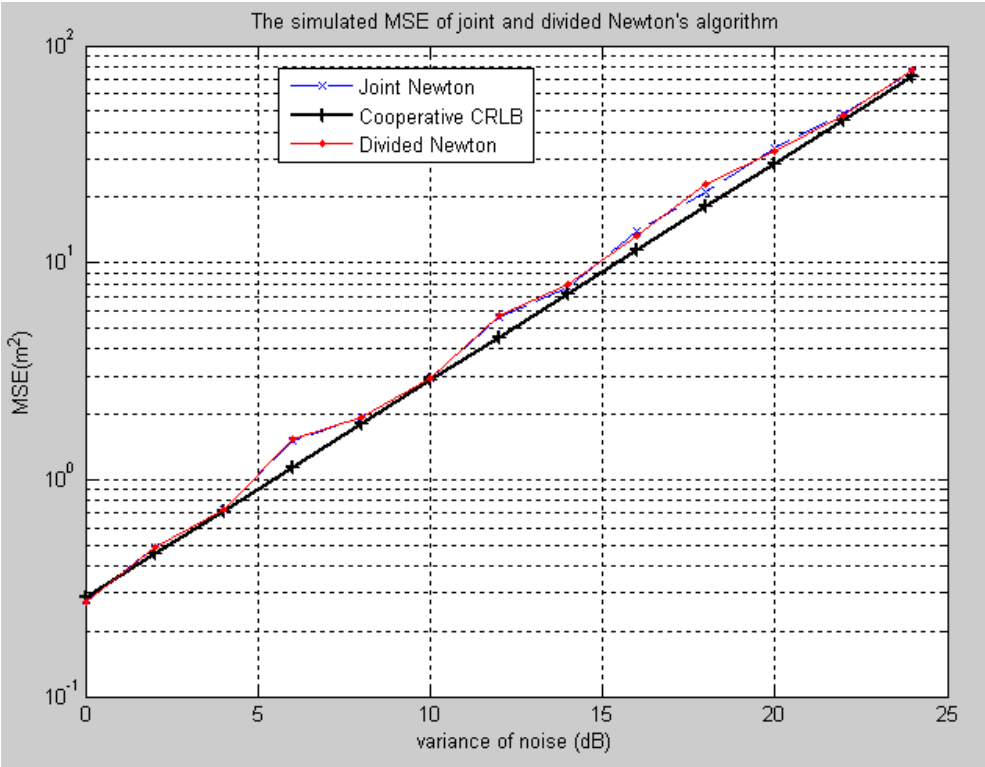



Figure 5.13 The MSE vs. noise variance for joint and divided Newton's algorithm.

Figure 5.13 Shows the MSE of joint and divide Newton's algorithm. The simulated MSE of joint Newton's algorithm in (3.7) and divided Newton's algorithm (3.26) are very close to each other and also close to CRLB. The CRLB is calculated from full cooperative FIM (4.7).

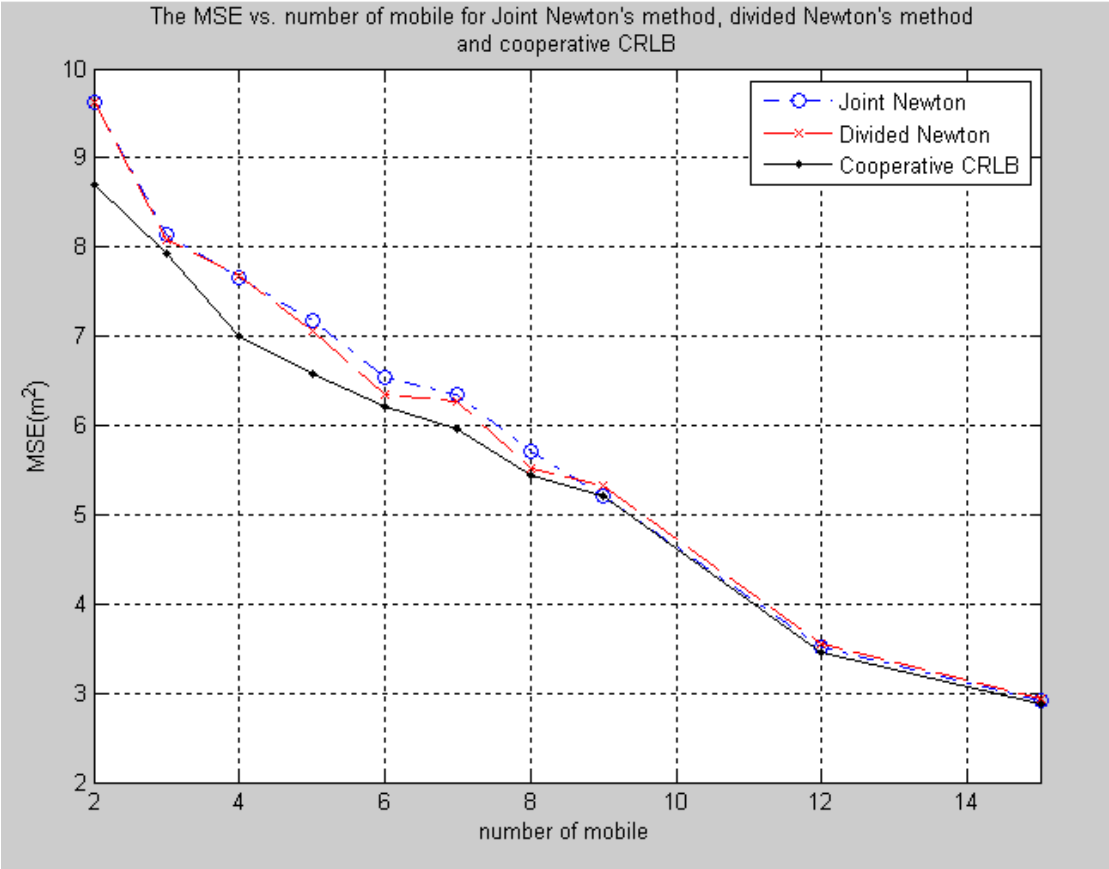



Figure 5.14 The MSE vs. number of mobile for joint Newton's algorithm, divided Newton's algorithm and cooperative CRLB.

The MSE vs. number of mobile for joint Newton's algorithm, divided Newton's algorithm and cooperative CRLB are plotted in Figure 5.14. We can see the more mobiles, the better MSE performance. Then, in few mobiles situation, the MSE performance of divided Newton's algorithm is better than joint Newton's. However, in more mobiles situation, the MSE performance has opposite result, but the both MSE performances are quiet close to cooperative CRLB.

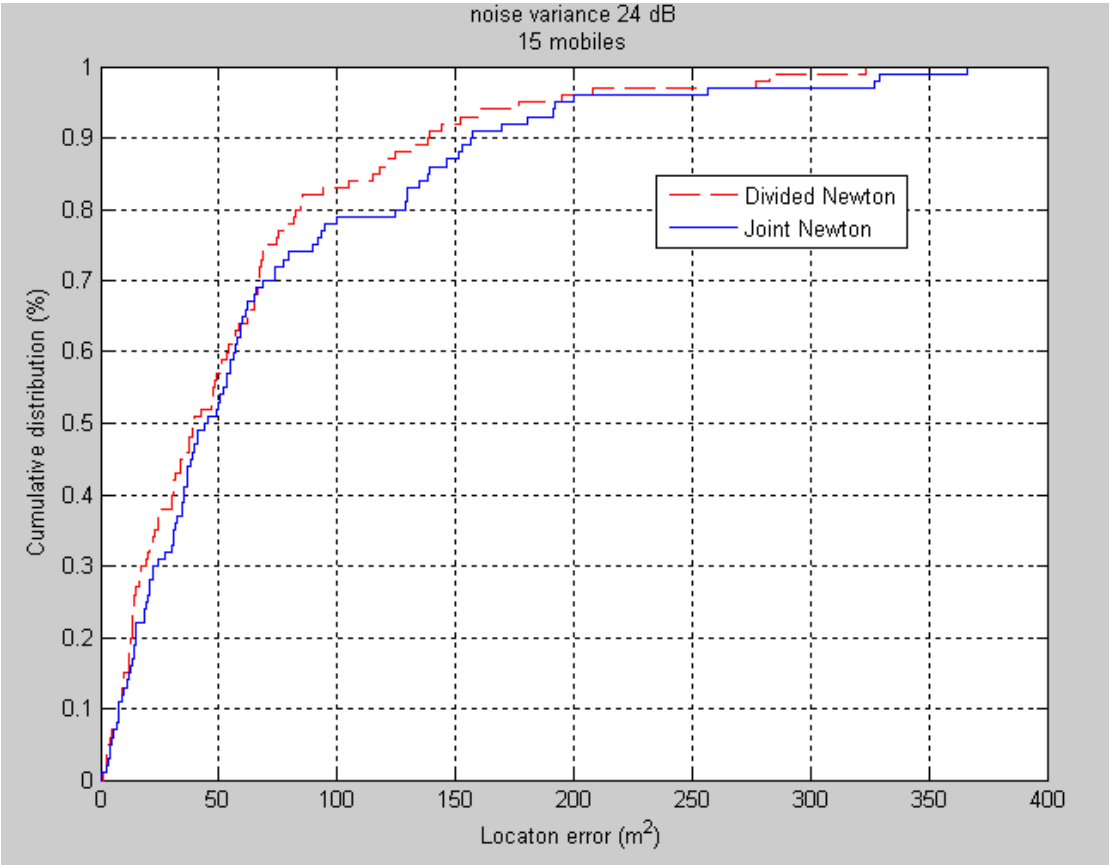



Figure 5.15 CDF comparison of joint and divided Newton's algorithm.

The CDF comparison of joint and divided Newton's algorithm is plotted in Figure 5.15. We can see the localization accuracy of joint Newton's algorithm is a little better than divided Newton's algorithm.

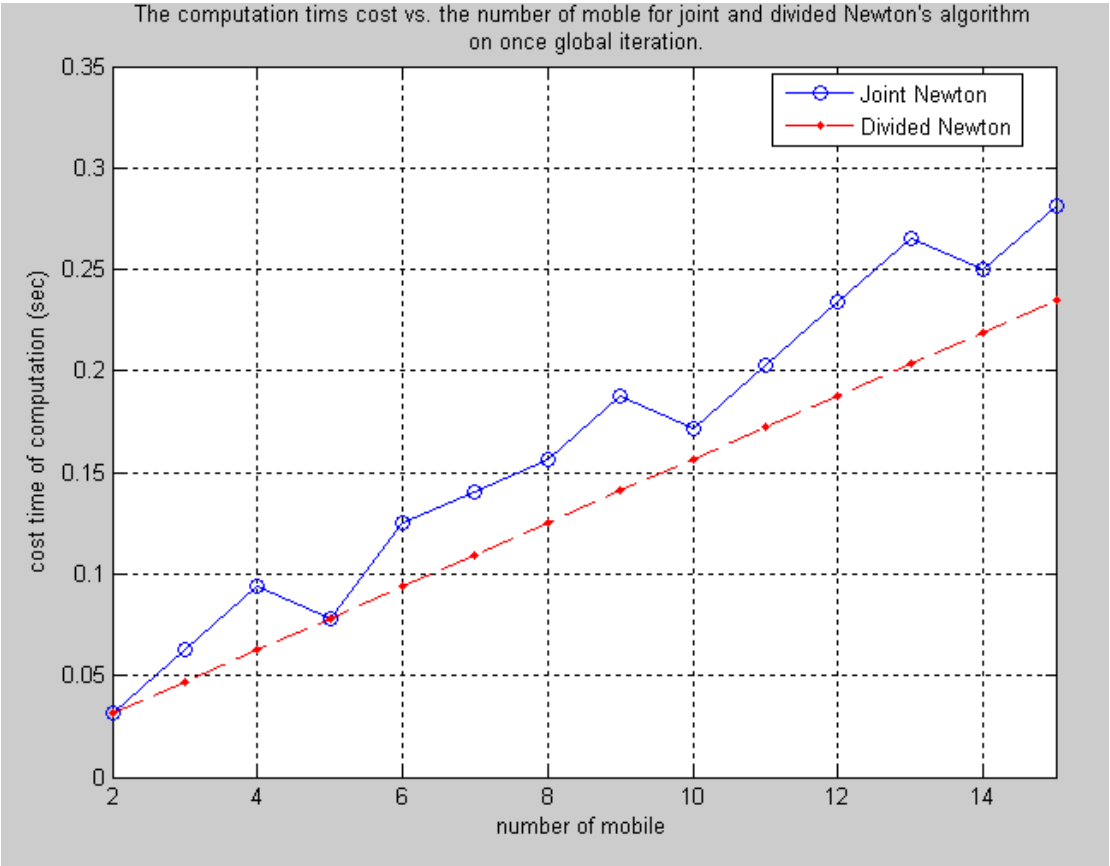



Figure 5.16 The computation time cost vs. the number of mobile for joint and divided Newton's algorithm on once global iteration.

Figure 5.16 shows the computation cost vs. number of mobile for joint and divided Newton's algorithm on once global iteration. We can see the computer time cost of joint Newton's algorithm $F_{2M\times 2M}\times G_{\text{joint}}$ (3.9) is higher than divided Newton's algorithm's cost $F_{2\times 2}\times L\times M\times G_{devided}$ (3.27) when $G_{\text{joint}}=G_{devided}$=1. Obviously, the divided Newton's algorithm reduces the computer time cost. Based on experience, the divided algorithm has only three times of global iteration to get converged, $G_{devided}=3$ and the joint algorithm has at least eight times of global iteration, $G_{\text{joint\_two mobiles}}=8$ to get converged on two mobiles. Therefore, the cost time of divided Newton's algorithm, $F_{2\times 2}\times L\times M\times G_{devided}$ is smaller cost time of joint

Newton's algorithm, $F_{2M\times 2M}\times G_{\text{joint}}$.

### 5.3.2 Taylor-Series Expansion Method

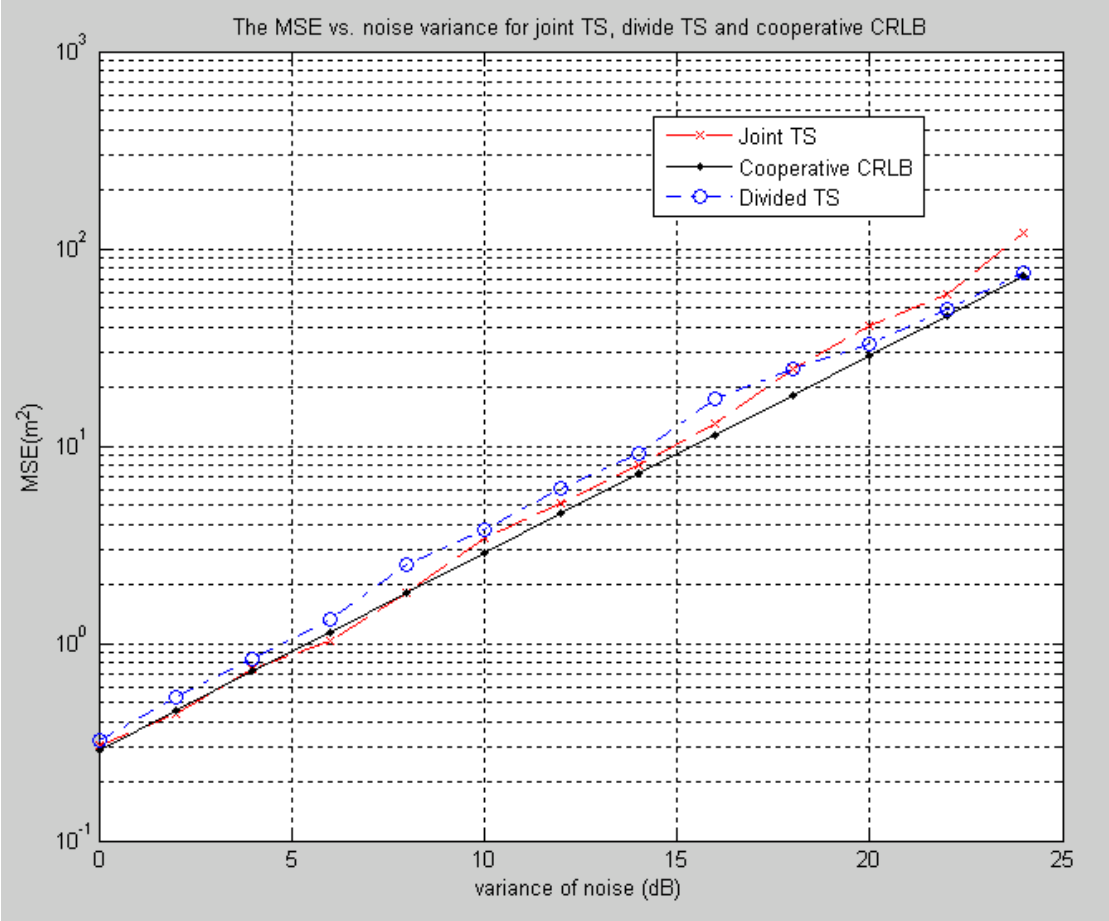



Figure 5.17 The MSE vs. noise variance for divided TS, joint TS and cooperative CRLB.

Figure 5.17 shows the MSE vs. noise variance for divided TS, joint TS and cooperative CRLB. In low noise variance, the MSE performance of joint TS is better than divided TS. However, in high noise variance, joint TS has worse MSE performance.

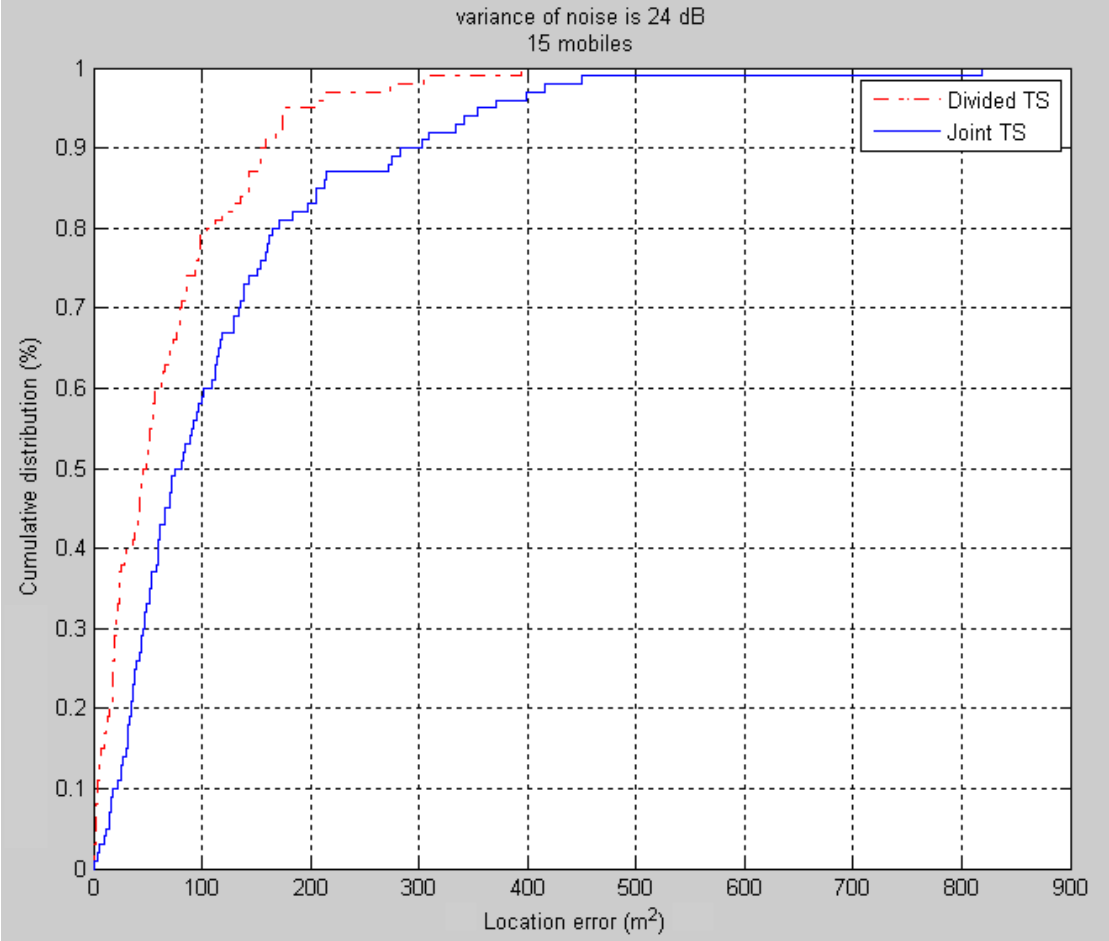



Figure 5.18 The CDF comparison of joint and divided TS algorithms.

The CDF comparison of joint and divided TS algorithms is plotted in Figure 5.18. The divided TS has the better localization accuracy. The result is similar to the comparison of joint and divided Newton's algorithm in Fig. 15.

## 5.4 Comparison of Divide-and-Conquer Algorithms

We have one divided nonlinear algorithm; divided Newton's algorithm and three divided linear algorithms; divided TS, AD and HC. Because we don't have the theoretical converged MSE of them, we use computer simulation to show the MSE performance and the convergence rate of Jacobi method and Gauss-Seidel method in following section. Previous description, we showed we can use compensation of uncertain virtual sensor to enhance the localization accuracy. However, divided Newton's algorithm can not achieve it because we don't have the theoretical MSE (3.26). On the contrary, the other three divided linear algorithms can accomplish the compensation. Therefore, we show the computer simulation to show the result by divided TS. The other divided linear algorithms have same outcome.

The rest of the paper is organized as follows. The convergence MSE and rate of divided Newton's algorithm, TS, DA and HC are shown in Section 5.4.1, 5.4.2, 5.4.3 and 5.4.4.

### 5.4.1 Newton's Method

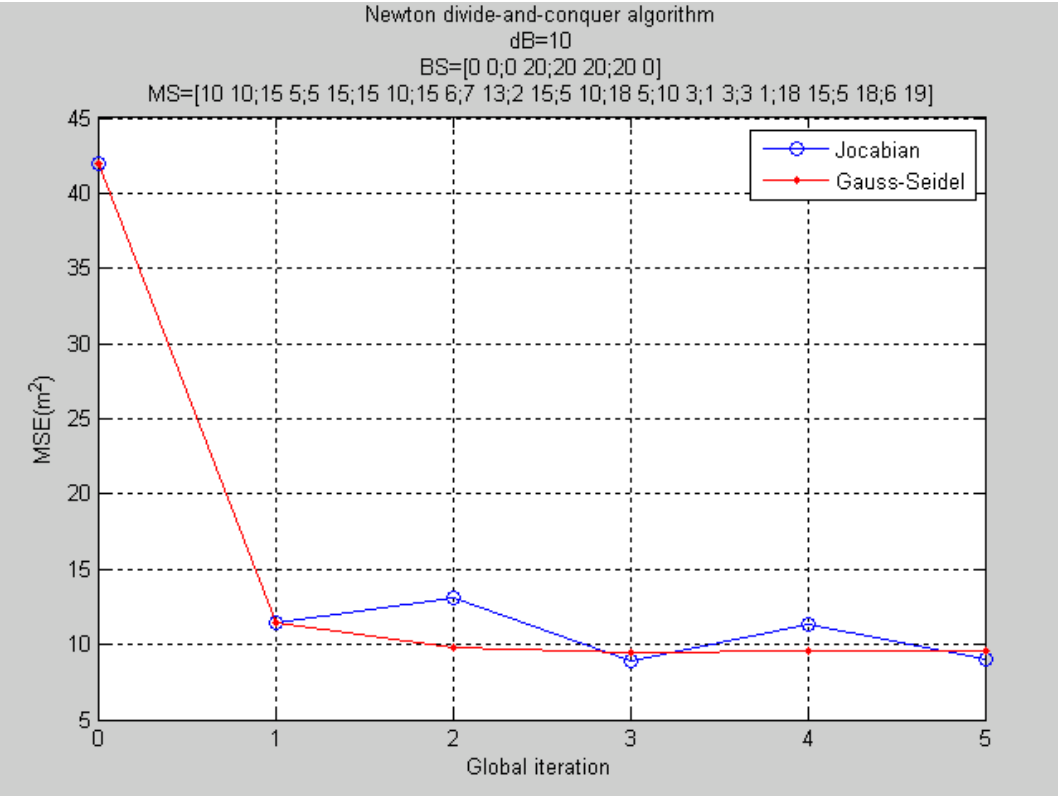



Figure 5.19 The MSE vs. Global iteration for Compare Jacobi and Gauss-Seidel.

Figure 5.19 shows the convergence rate for two update schemes, Jacobi method (3.24) and Gauss-Seidel method (3.25). Obviously, the convergence rate of Gauss-Seidel method is faster than Jacobi method.

### 5.4.2 Taylor-Series Expansion Algorithm

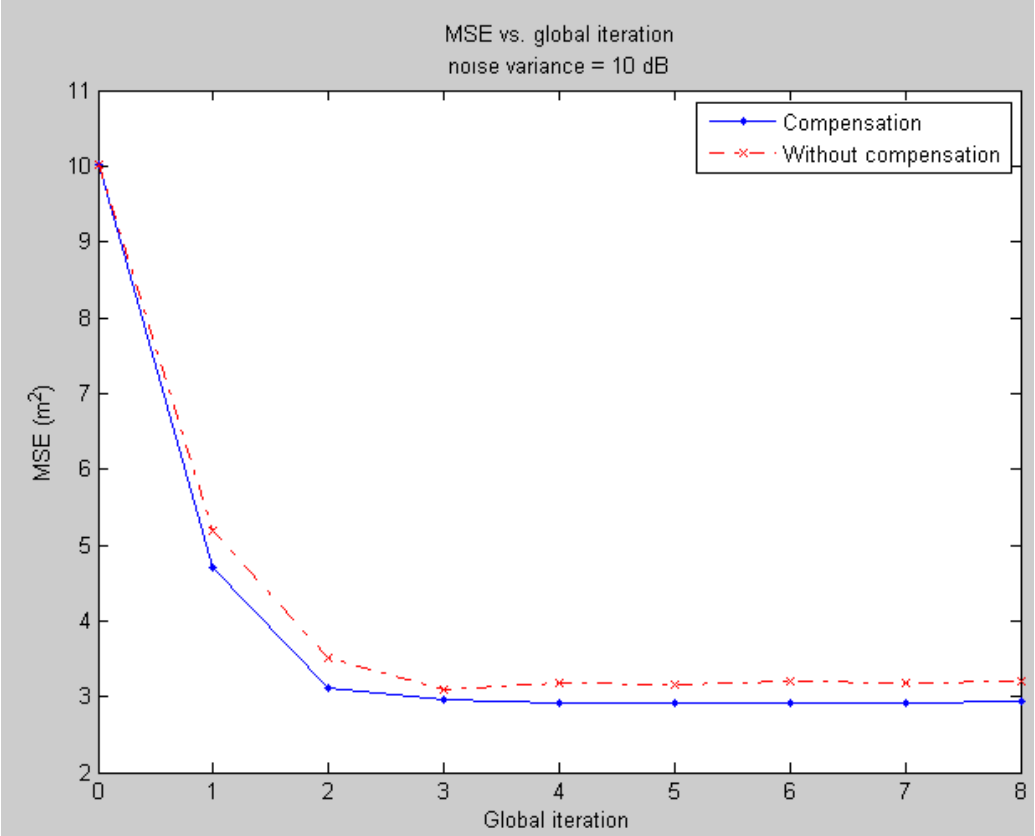



Figure 5.20 The MSE vs. global iteration for divided TS.

Figure 5.20 shows the MSE vs. global iteration for divided TS. The compensation of uncertain virtual sensor (3.17) can improve the MSE performance.

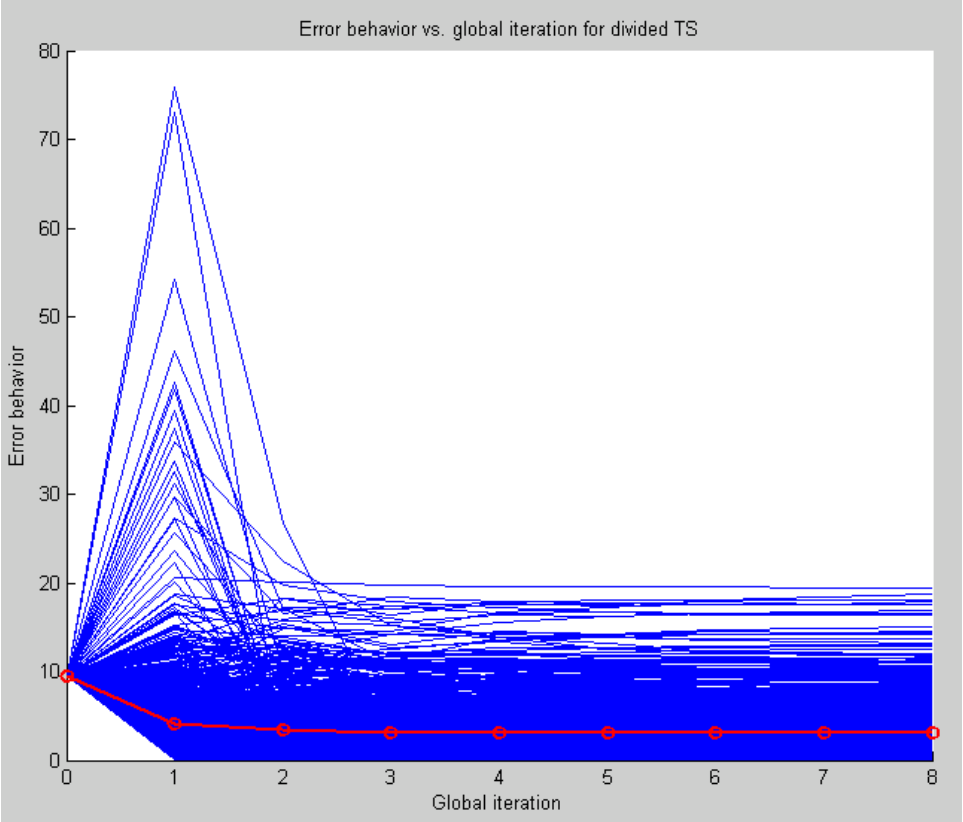



Figure 5.21 The error and MSE vs. global iteration for divided TS.

The error and MSE vs. number of mobile for divided TS is plotted in Figure 5.21. The iteration can improve the MSE performance. In most of time, it has stable situation. Sometime, the 1$^{st}$ iteration has the worst localization accuracy. However, it exists that cooperation can not improve localization. But all of them are converged.

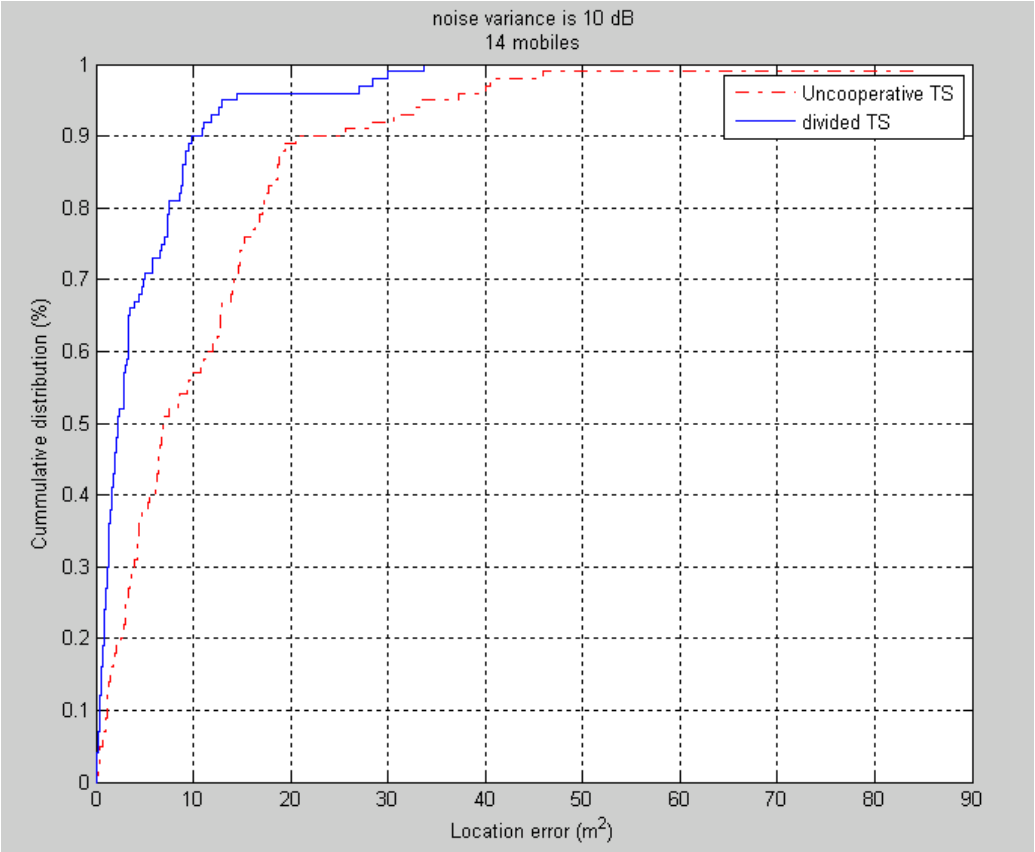



Figure 5.22 The CDF comparison of divided TS and uncooperative TS.

The CDF comparison of joint and divided TS and uncooperative TS is plotted in Figure 5.22. The localization accuracy of divided TS is better than uncooperative TS.

### 5.4.3 Distance-Augmented Algorithm

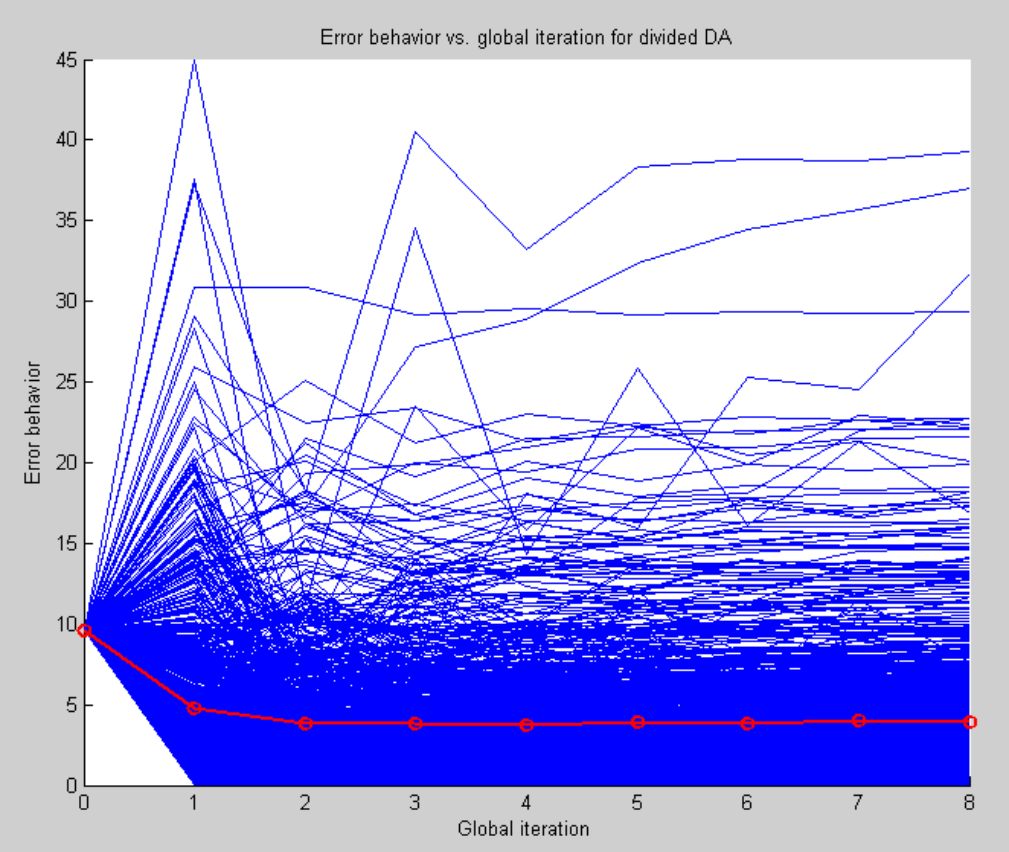



Figure 5.23 The error and MSE vs. global iteration for divided DA.

Figure 5.23 shows the error and MSE vs. number of mobile for divided DA. As divided TS algorithm, in most of time, it is stable. But it could be diverged.

.

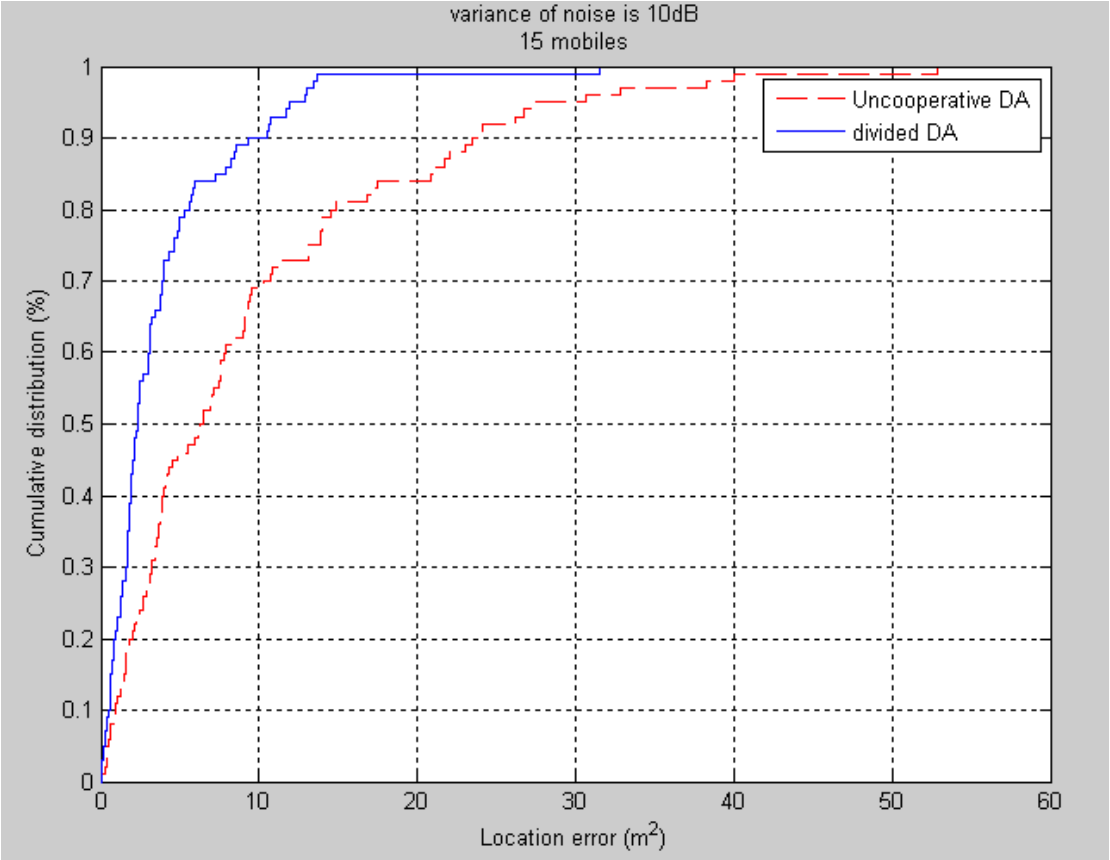



Figure 5.24 The CDF comparison of divided DA and uncooperative DA.

The CDF comparison of joint and divided DA and uncooperative DA is plotted in Figure 5.24. As before, the localization accuracy of divided DA is better than uncooperative DA.

### 5.4.4 Hyperbolic-Canceled Algorithm

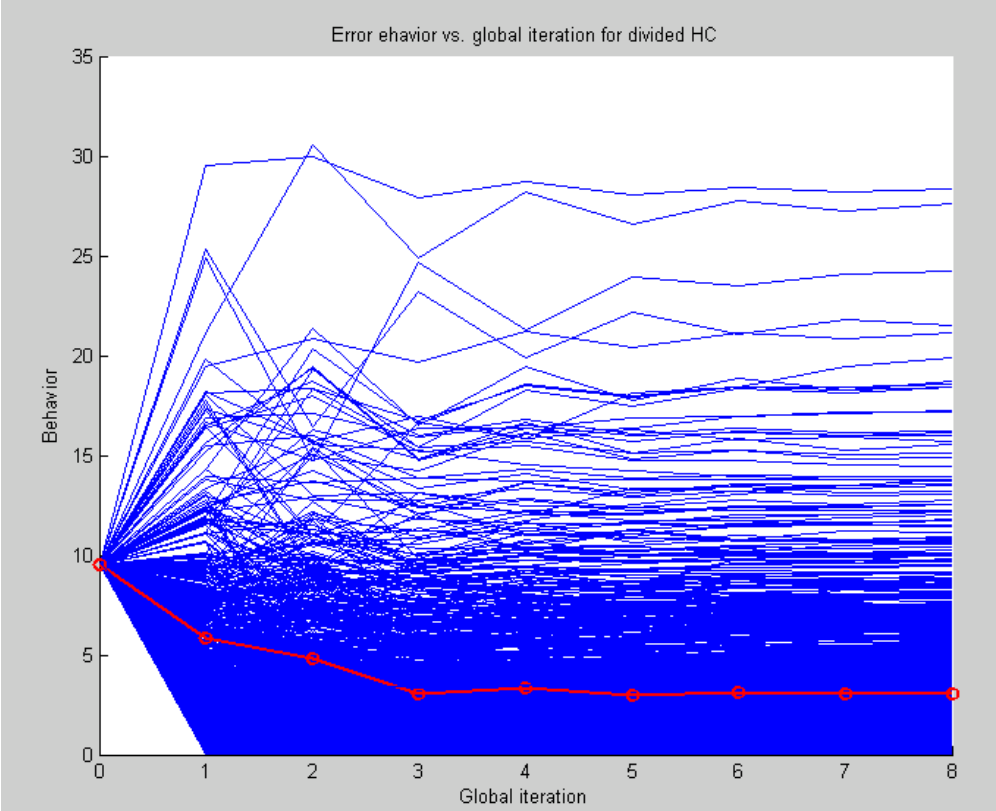



Figure 5.25 The error and MSE vs. global iteration for divided HC.

Figure 5.25 shows the error and MSE vs. number of mobile for divided HC. As divided TS, in most of time, it has acceptable localization result.

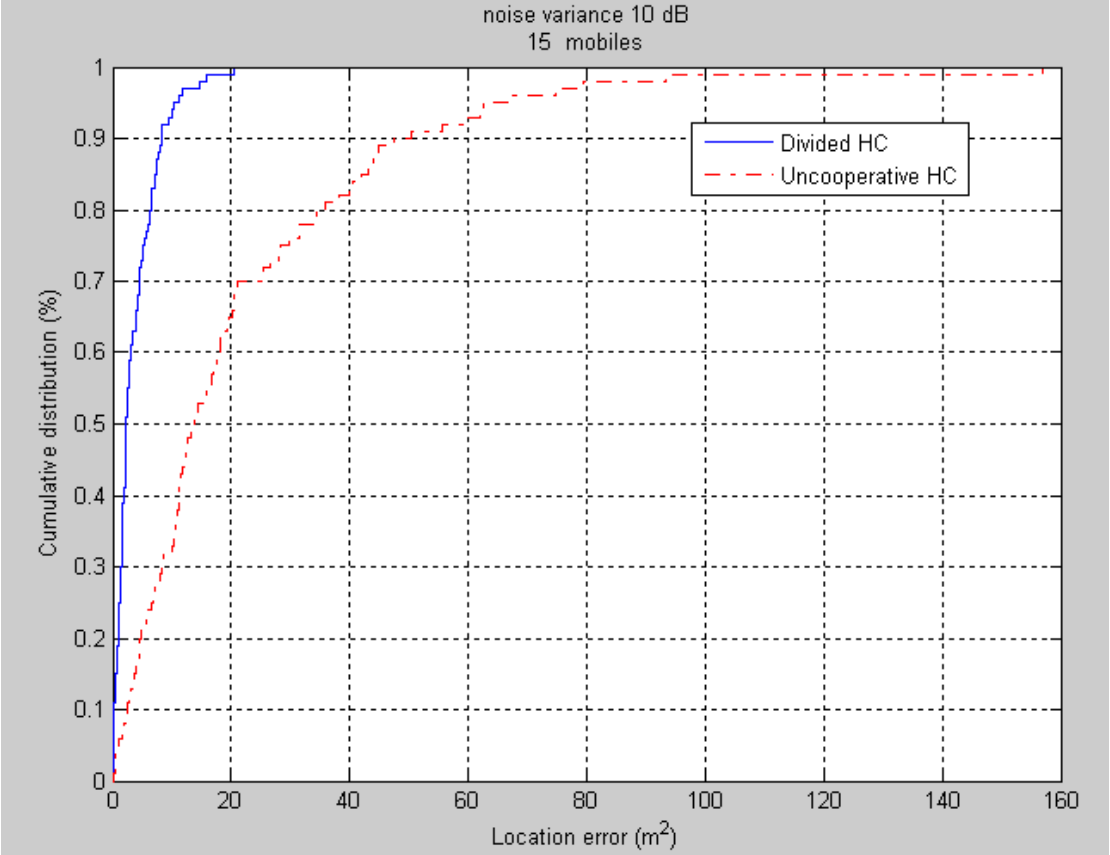



Figure 5.26 The CDF comparison of divided HC and uncooperative HC.

The CDF comparison of joint and divided HC and uncooperative HC is plotted in Figure 5.26. As before, the localization accuracy of divided HC is better than uncooperative HC.

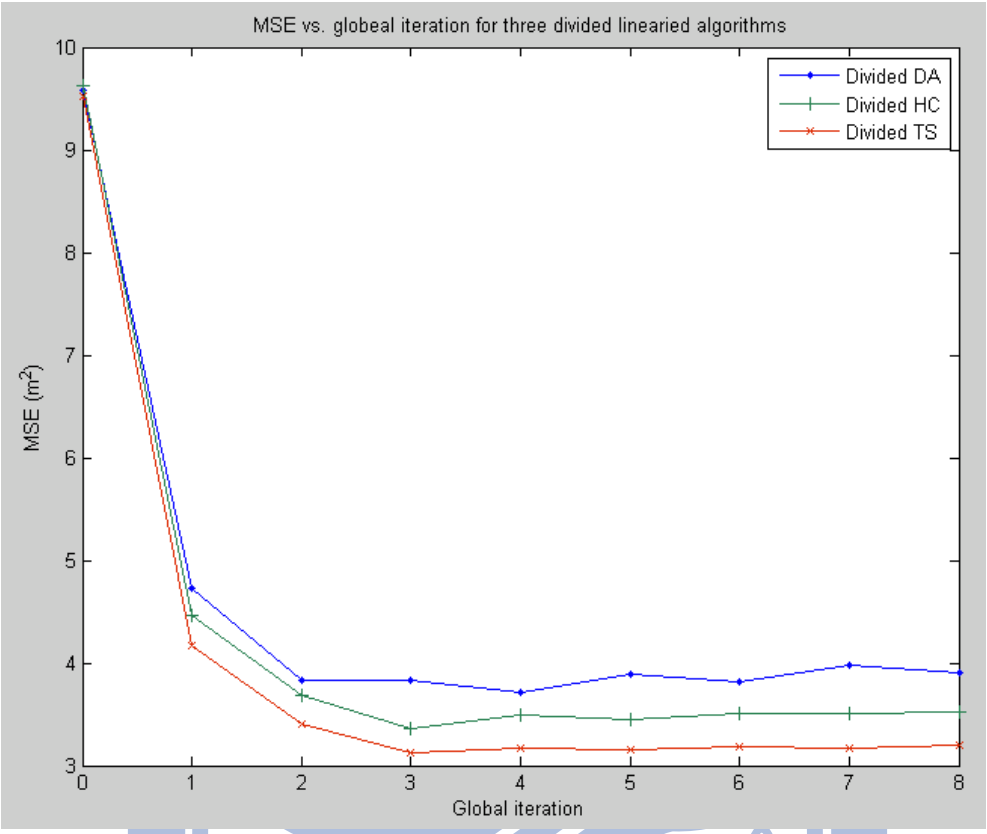



Figure 5.27 The MSE vs. global iteration for three divided linearized algorithms.

The MSE vs. global iteration for three divided linearized algorithms are plotted in Figure 5.27. we can see TS has the best MSE performance and DA's performance is the worst one.

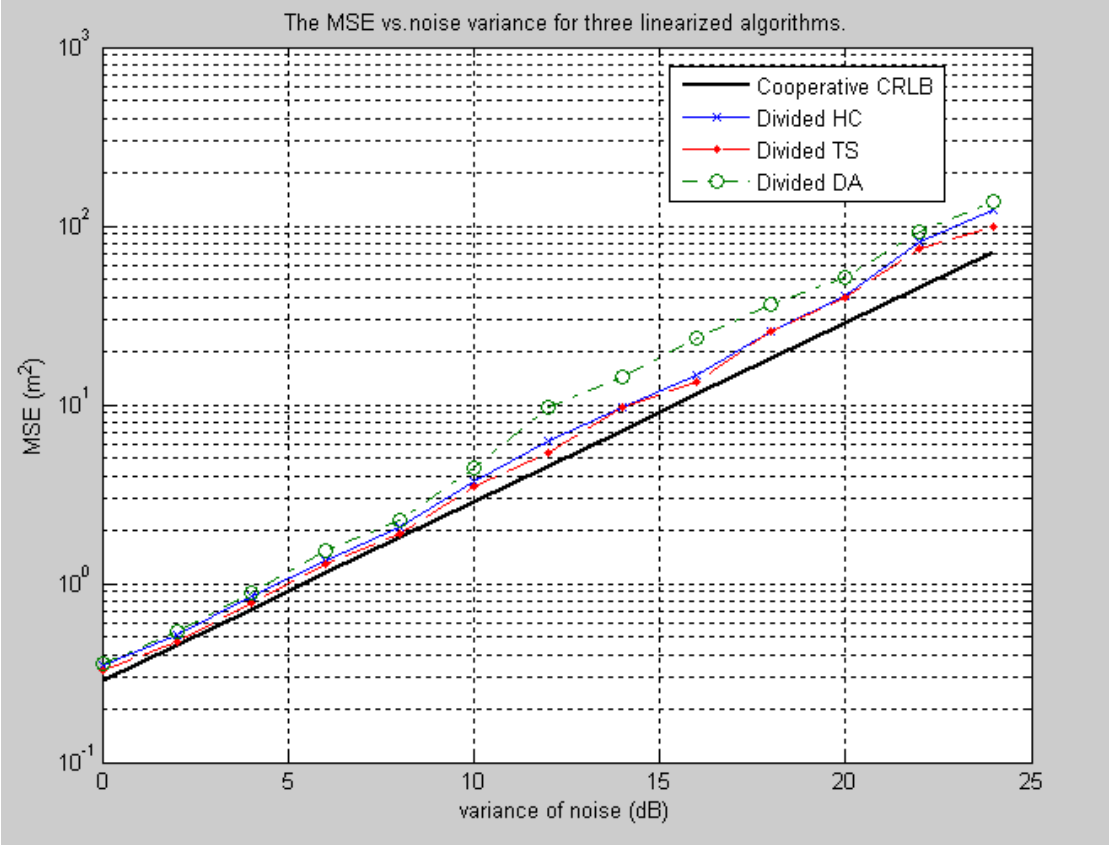



Figure 5.28 The MSE vs. noise variance for three linearized divided algorithms.

The MSE vs. noise variance for three linearized divided algorithms are plotted in Figure 5.28. We can see that the divided TS has the best MSE performance and divided DA has the worse one. We can expect that from the MSE performance of uncooperative linearized algorithms in Figures 5.3 to 5.7.

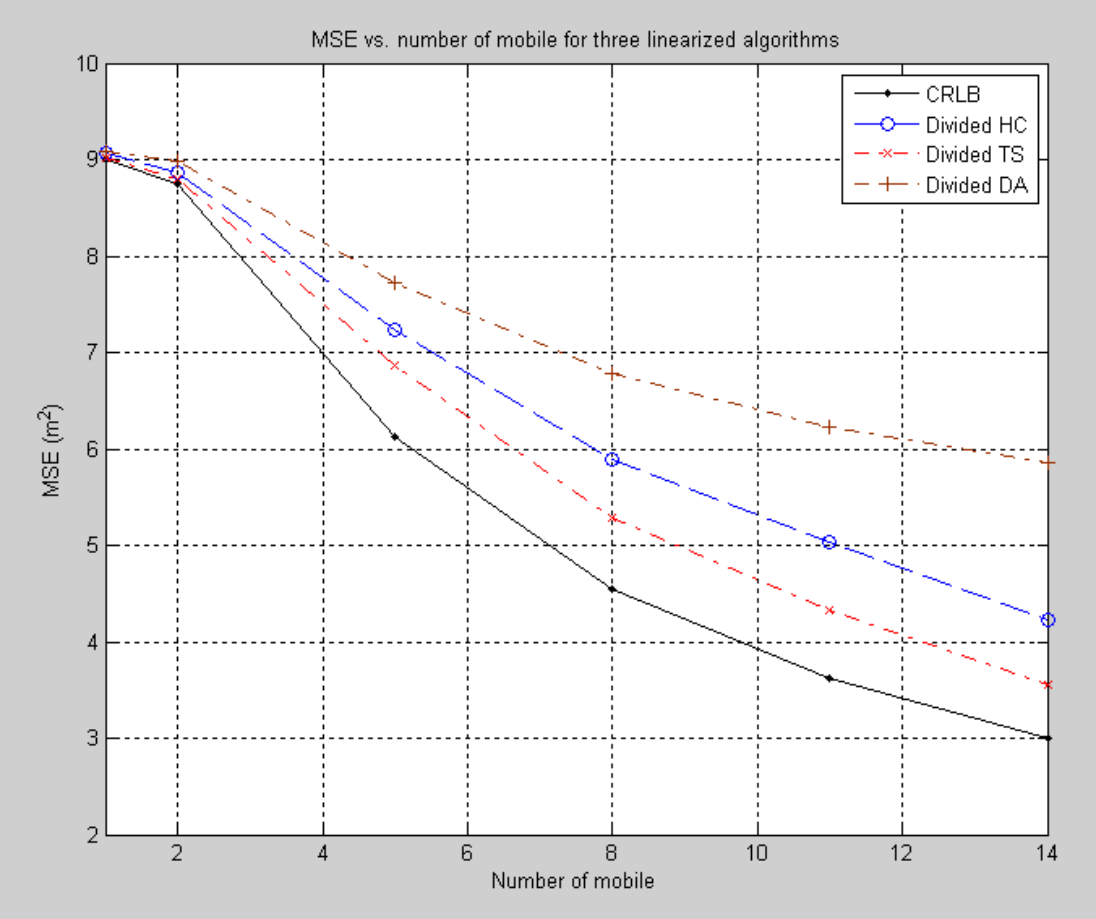



Figure 5.29 The MSE vs. number of mobile for three divided linearized algorithms.

The MSE vs. number of mobile for three divided linearized algorithms are plotted in Figure 5.29. We can see that the more mobiles, the better MSE performance. However, divided TS has the best performance and divided DA has the worst. No matter witch on algorithm, the more mobiles, and the farter from cooperative CRLB.

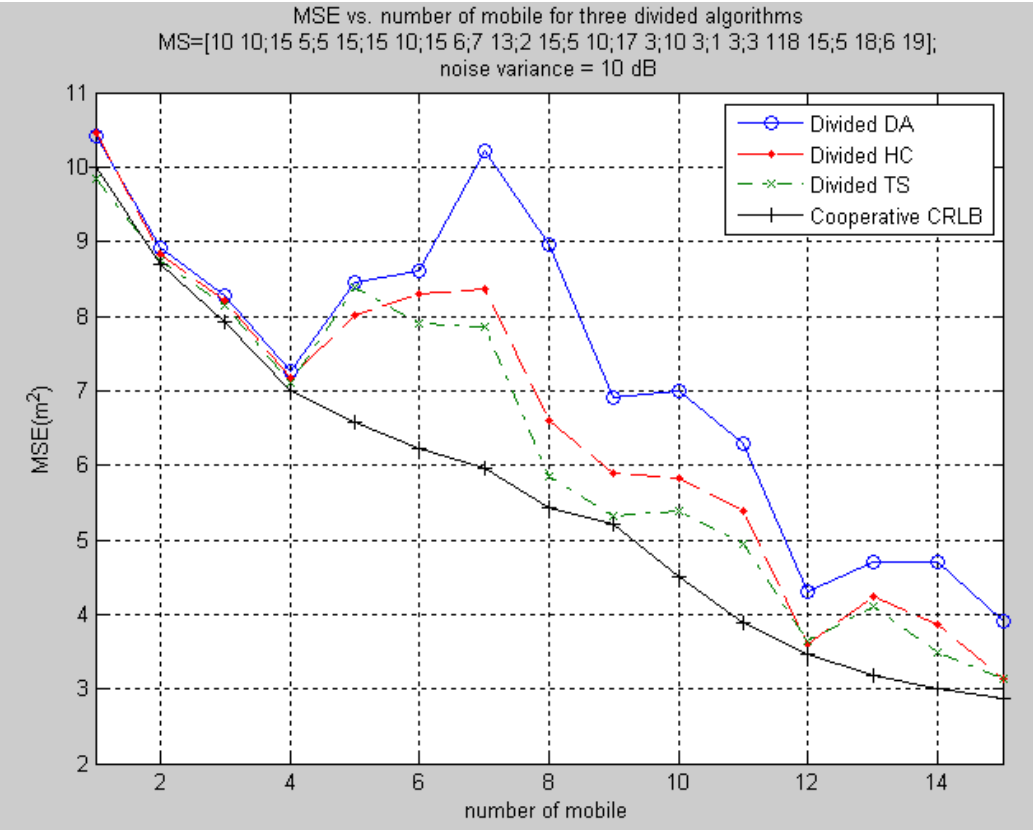



Figure 5.30 The MSE vs. number of mobile for three divided linearized algorithms with fixed 14 mobiles.

The MSE vs. number of mobile for three divided linearized algorithms with fixed 14 mobiles are plotted in figure 5.30. we can see that if we fix 14 mobiles at (15, 5)m, (5, 15)m, (10, 15)m, (15, 6)m, (7, 13)m, (2, 15)m, (5, 10)m, (17, 3)m, (10, 3)m, (1, 3)m, (3, 1)m, (18, 15)m, (5, 18)m and (6, 19)m, it can not ensure that more mobiles can provide localization benefit.

## 5.5 Theoretical Analysis of Mean-Square-Error

Previous description, we know the cooperative CRLB from full cooperation FIM (4.7) is very difficult and complex when the number of mobile increases. Therefore, we first derive AEFIM (4.54). We can see that if there are more mobiles, the MSE performance will be improved. Then, we derive AC-CRLB (4.64) by AEFIM. We can discover that the cooperative angles and number of mobile can influence the MSE performance. Computer simulation will compare the true cooperative CRLB and approximation (cooperative CRLB from AEFIM and AC-CRLB), and show factors how to effect the MSE performance in Section 5.5.1. However, we also derive the converged MSE for divided TS in special case $M=2$. Therefore, we utilize computer simulation to show the result in Section 5.5.2.

### 5.5.1 Approximation of CRLB

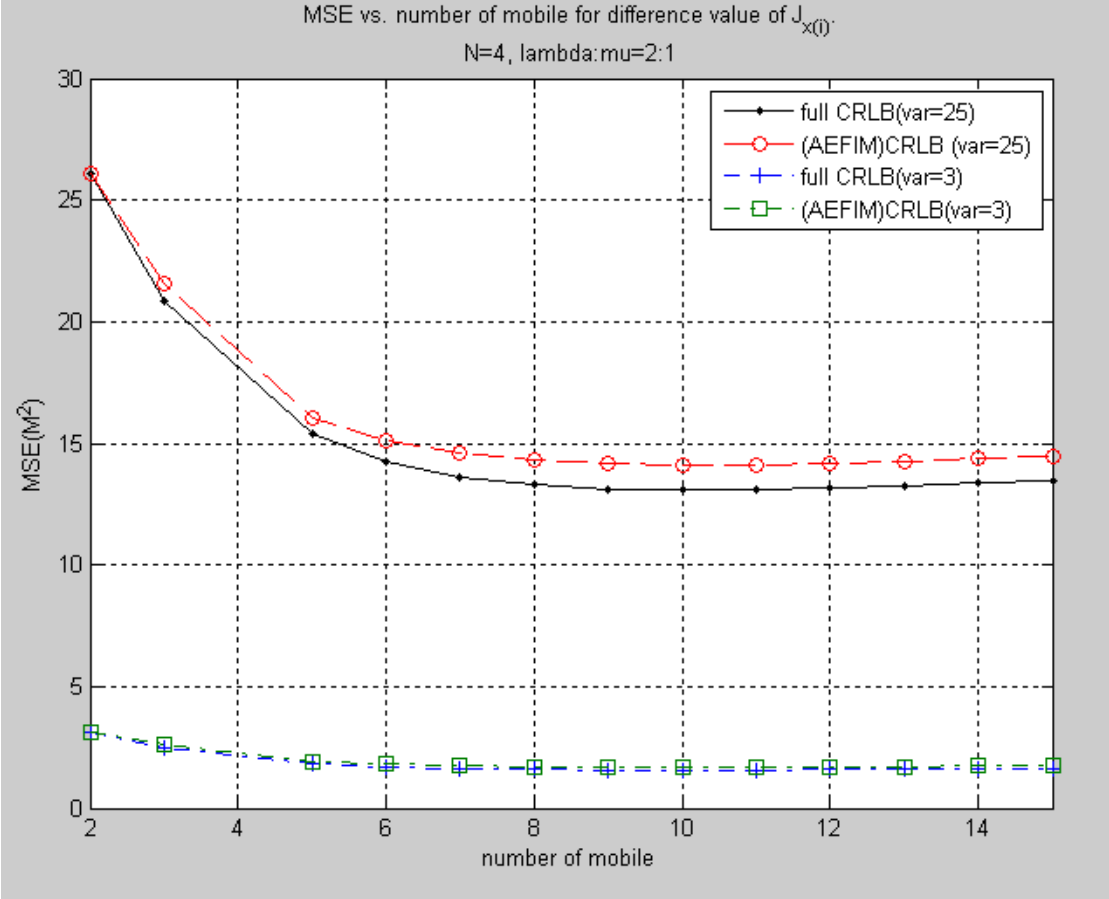



Figure 5.31 The MSE vs. number of mobile for difference value of $\mathbf{J}_{\mathbf{x}_i}$.

The MSE vs. number of mobile for difference value of $\mathbf{J}_{\mathbf{x}_i}$ are plotted in Figure 5.31. it sets omni-direction cooperative mobiles. We can see that if the $\mathbf{J}_{\mathbf{x}_i}$ is larger, the CRLB of AEFIM (4.54) is closer to cooperative CRLB (4.8). The reason is that Block matrix inversion lemma 1 is proposed under larger $\mathbf{J}_{\mathbf{x}_i}$. Even if $\mathbf{J}_{\mathbf{x}_i}$ is smaller, the CRLB of AEFIM is not far from cooperative CRLB.

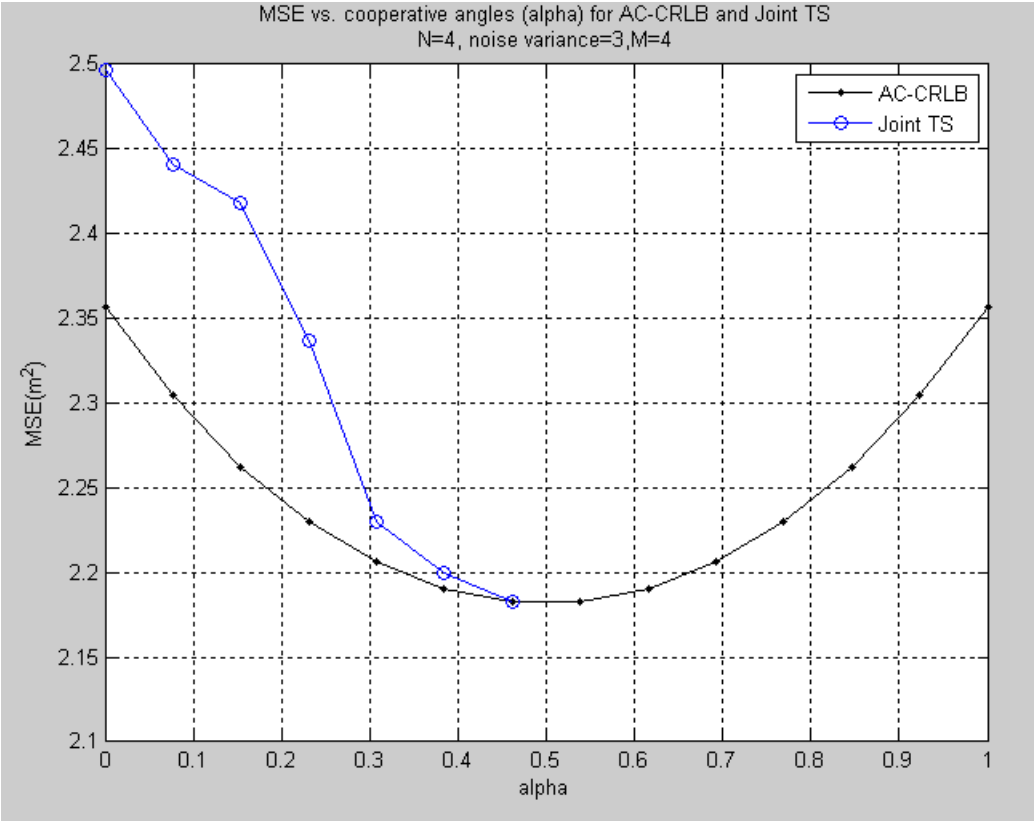



Figure 5.32 The MSE vs. cooperative angles (alpha) for AC-CRLB and joint TS.

The MSE vs. cooperative angles (alpha) for AC-CRLB and joint TS are plotted in Figure 5.32. We set *M=4*, $\chi = 0.5$ in (4.64). The eigenvalue $\alpha$ in (4.64) means a trend of the cooperative angles. If the value of alpha has minimum zero and maximum $(M-1)/\sigma^2$, that means these cooperative angles tend to one direction. Therefore, AC-CRLB will has the maximum value. However, if $\alpha = (M-1)/2\sigma^2$, AC-CRLB has the best MSE performance. The MSE performance of joint TS can show that effect of cooperation angles. We can see the omni-direction is better than beam.

### 5.5.2 Divided Taylor-series Expansion for Twins Mobiles

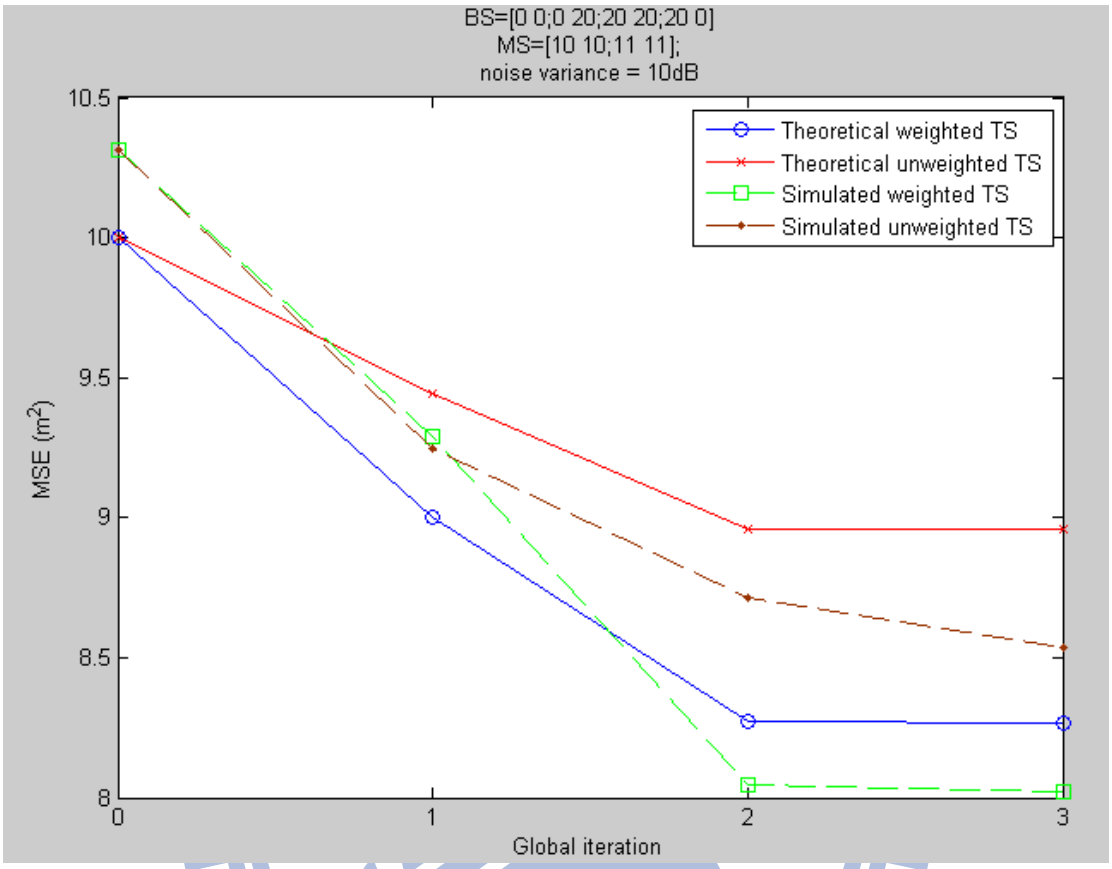



Figure 5.33 The MSE vs. global iteration of divided TS for two mobiles.

Figure 5.33 shows the MSE vs. global iteration of divided TS for two mobiles. We can see the that no matter theoretical MSE or simulated MSE, the performance of weighted TS is better than unweithted TS. Even if the simulated results are not really close to theoretical MSE, but the trend are very similar.

# Chapter 6
# Conclusions and Future Works

Joint TS algorithm and three linearized divided algorithms based on distance measurement have been devised for cooperative localization system. Alternative realizations with smaller computational complexity are also proposed. In addition, theoretical performance measure, namely, biases and mean square error of the sensor position estimates are derived and verified by computer simulation. It is shown that the proposed positioning methods have lower computation cost and available MSE performance. Besides, we investigate the fundamental limits of cooperative localization system with ranging capability. We then applied the notion of equivalent Fisher information matrix (EFIM) to derive and simplify the full cooperative FIM, approximation of EFIM (AEFIM). We can see the more cooperative mobiles, the better of MSE performance. Than, we further utilize AEFIM to simplify the cooperative Cramer-Roa Lower Bound (CRLB), named approximation of cooperative CRLB (AC-CRLB). From AC-CRLB, we know that the measurement error variance, numbers of sensors, numbers of mobiles and cooperative angles can affect the localization accuracy. We also derive the theoretical converged MSE of divided TS for two mobiles. The simulation has been shown that these approximations are reliable.

Therefore, the derivation of theoretical MSE performance for divided linearized algorithms is also an interesting research topic and a good starting point is to investigate that sensors placement and cooperative mobiles in wireless sensor networks.

# Bibliography

[1] P. Biswas, T-C Liang, K-C Toh, Y. Ye, and T-C Wang.; “Semi-definite programming approaches for sensor network localization with noisy distance measurements,” *Automation Science and Engineering, IEEE Transactions*, pp. 360-371, 2006.

[2] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M.Z. Win.; “Ranging with ultrawide bandwidth signals in multipath environments,” *Proceedings of the IEEE*, pp. 404-426, 2009.

[3] Patwari, N. Ash, J.N., Kyperountas, S.; Hero, A.O., III; Moses, R.L.; Correal, N.S.; “Locating the nodes: cooperative localization in wireless sensor networks,” *IEEE Signal Processing Magazine*, pp. 54-69, 2005.

[4] Gustafsson, F.; Gunnarsson, F.; “Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements ” *IEEE Signal Processing Magazine*, pp. 41-53, 2005.

[5] K.C. Ho, X.N. Lu, and L. Kovavisaruch.; “Source localization using TDOA and FDOA measurements in the presence of receiver location errors: analysis and solution,“ *Signal Processing IEEE Transactions*, pp. 684 - 696, 2007.

[6] A.N. Bishop, B. Fidan, B.D.O. Anderson, K. Dogancay; and P.N. Pathirana.; “Optimal range-difference-based localization considering geometrical constraints,” *IEEE Journal of Oceanic Engineering*, pp. 289-301, 2008.

[7] M. Souden, S. Affes, and J. Benesty.; “A two-stage approach to estimate the angles of arrival and the angular spreads of locally scattered sources,”*Signal Processing, IEEE Transactions*, pp. 1968-1983, 2008.

[8] X. Li.; “Collaborative localization with received-signal strength in wireless sensor networks,” *Vehicular Technology, IEEE Transactions*, pp. 3807-3817, 2007.

[9] H. Ren, and M. Meng.; “Power adaptive localization algorithm for wireless sensor networks using particle filter,” *Vehicular Technology, IEEE Transactions*, pp. 2498-2508, 2009.

[10] Hsin-Yuan Chen; Tung-Yi Chou.; “Hybrid TDOA/AOA mobile user location with artificial neural networks, ” *Networking, Sensing and Control, IEEE International Conference*, pp. 847-852, 2008.

[11] A. Broumandan, T. Lin, J. Nielsen, and G. Lachapelle.; “Practical results of hybrid AOA/TDOA geo-location estimation in CDMA wireless networks,” *IEEE 68th Vehicular Technology Conference*, pp. 1-5, 2008.

[12] Chan, F.; So, H.C.; Ma, W.-K.; “A novel subspace approach for cooperative localization in wireless sensor networks using range measurements,” *Signal Processing, IEEE Transactions*, pp. 260-269, 2009.

[13] Yiheng Zhang, Qimei Cui, Xiaofeng Tao.; “Cooperative group localization for 4G wireless networks,” *Vehicular Technology Conference*, pp. 1-5, 2009.

[14] Orooji, M.; Abolhassani, B.; “New method for wstimation of mobile location based on signal attenuation and Hata model signal prediction,” *Engineering in Medicine and Biology Society, IEEE-EMBS 27th Annual International Conference*, pp.6025-6028, 2005.

[15] Alsindi, N.A.; Pahlavan, K.; Alavi, B.; “An error propagation aware algorithm for precise cooperative indoor localization,” *Military Communications Conference, IEEE*, pp.1-7, 2006.

[16] I. Guvenc, C-C Chong, and F. Watanabe.; “NLOS identification and mitigation for UWB localization systems,” *Wireless Communications and Networking Conference*, pp. 1571-1576, 2007.

[17] Seow, C. Kiat, Tan, and S. Yim.; “Non-line-of-sight localization in multipath environments,” *Mobile Computing, IEEE Transactions*, pp. 647-660, 2008.

[18] G. Reina, A. Vargas, K. Nagatani, and K. Yoshida.; “Adaptive Kalman Filtering for GPS-based mobile robot localization,“ *Safety, Security and Rescue Robotics, IEEE International Workshop*, pp. 1-6, 2007.

[19] R. Zhan and J. Wan.; “Iterated unscented Kalman Filter for passive target tracking,” *Aerospace and Electronic Systems, IEEE Transactions*, pp. 1155-1163,

2007.

[20] Purvis, K.B.; Astrom, K.J.; Khammash, M.; “Estimation and optimal configurations for localization using cooperative UAVs,” *Control Systems Technology, IEEE Transactions*, pp. 947-958, 2008.

[21] Wei-Yu Chiu; Bor-Sen Chen.; “Mobile location estimation in urban areas using mixed manhattan/euclidean norm and convex optimization,” *Wireless Communications, IEEE Transactions*, pp. 414-423, 2009.

[22] Laurendeau, C.; Barbeau, M.; “Hyperbolic location estimation of malicious nodes in mobile WiFi/802.11 networks,” *Local Computer Networks, IEEE Conference*, pp. 600-607, 2008.

[23] Timothy Sauer.; “Numerical analysis, ” Boston: Pearson Addison Wesley, c2006.

[24] Gene H. Golub, Charles F. Van Loan.; “Matrix computations” 3rd, Johns Hopkins, 1996.

[25] H. V. Poor.; “An introduction to signal detection and estimation,” 2nd ed, New York: MA: Kluwer, 2000.

[26] Alsindi, N.A.; Pahlavan, K.; Alavi, B.; Xinrong Ei; ”A novel cooperative localization algorithm for indoor sensor networks, ” *Personal, Indoor and Mobile Radio Communications, IEEE*, pp.1-6, 2006.

[27] Doherty, L.; pister, K.S.J.; El Ghaoui, L.; “Convex position estimation in wireless sensor networks,” *Twentieth Annual Joint Conference, IEEE Computer and Communications Societies*, pp. 1655-1663, 2001.

[28] Coates, M.; “Distributed particle filters for sensor networks,” *Third International Symposium*, pp. 99-107, 2004.

[29] Alsindi, N.A.; Alavi, B.; Pahlavan, K.; “Measurement and modeling of ultra -wideband TOA-based ranging in indoor multipath environments,” *Vehicular Technology, IEEE Transactions*, pp. 1046-1058, 2009.

[30] Randolph L. Moses, Olph L. Moses, Dushyanth Krishnamurthy, Robert Patterson.; “A self-localization method for wireless sensor networks,” *Applied Signal Processing, EURASIP Journal*, 2002

[31] Yuan Shen, Wymeersch, H.; Win, M.Z.; “Fundamental limits of wideband cooperative localization via fisher information,” *Wireless Communications and Networking Conference, IEEE*, pp. 3951-3955, 2007.

[32] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, Yinyu Te.; “Semi-definite programming based algorithms for sensor network localization,” *Sensor Networks, ACM Transactions*, pp. 188-220, 2006.

[33] Shang, Y.; Rumi, W.; Zhang, Y.; Fromherz, M.; “Localization from connectivity in sensor networks,” *Parallel and Distributed Systems, IEEE Transactions*, pp. 961-974

[34] Drineas, P.; Javed, A.; Magdon-Ismail, M.; Pandurangant, G.; Virrankoski, R.; Savvides, A.; “Distance matrix reconstruction from incomplete distance information for sensor network localization,” *Sensor and Ad Hoc Communications and Networks, IEEE Communications Society* , pp. 536-544, 2006.

[35] W. H. Foy, “Position-location solutions by Taylor-series estimation,” *Aerospace and Electronic Systems, IEEE Trans*, pp. 187-194, 1976.

[36] H. Ren, and M. Meng, “Power adaptive localization algorithm for wireless sensor networks using particle filter,” *Vehicular Technology, IEEE Transactions* , pp. 2498-2508, 2009.

[37] U. Sarac, F.K. Harmanci, T. Akgul.; “Experimental analysis of detection and localization of multiple emitters in multipath environments,” *Propagation Magazine and IEEE Antennas*, pp. 61-70, 2008.

[38] P. Tarrio, A.M. Bernardos, J.A. Besada, and J.R. Casar.; “A new positioning technique for RSS-Based localization based on a weighted least squares estimator,” *Wireless Communication Systems, IEEE International Symposium*, pp. 633-637, 2008.

[39] A. Beck, P. Stoica, and J. Li.; “Exact and approximate solutions of source localization problems,” *Signal Processing, IEEE Transactions*, pp. 1770-1778, 2008.

[40] J. Yi, and M.R. Azimi-Sadjadi, “A robust source localization algorithm applied to acoustic sensor network,” *ICASSP 2007, IEEE International Conference*, pp. III-1233 - III-1236, 2007.
[41] Mayorga, Carlos Leonel Flores; della Rosa, Francescantonio; Wardana, Satya Ardhy; Simone, Gianluca; Raynal, Marie Claire Naima; Figueiras, Joao; Frattasi, Simone; “Cooperative positioning Techniques for mobile localization in 4G cellar networks,” *Digital Object Identifier, IEEE International Conference*, pp.39-44, 2007.

[42] Zhang Yiheng, Cui Qimei, Tao Xiaofeng.; “Cooperative Positioning for the Converged Networks” *Vehicular Technology Conference, IEEE*, pp.1-6, 2009.

[43] Ho, K.C.; Le Yang; “On the use of a calibration emitter for source localization in the presence of sensor position uncertainty,” *Signal Processing, IEEE Transactions*, pp. 5758 - 5772 , 2008.

[44] Yuan Shen; Wymeersch, H.; Win, M.Z.; “Fundamental limits of wideband cooperative localization via fisher information,” *Wireless Communications and Networking Conference, IEEE*, pp. 3951-3955, 2007.

[45] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M.Z. Win, “Ranging with ultrawide bandwidth signals in multipath environments,” *Proceedings of the IEEE*, pp. 404-426, 2009.

[46] Gillette, M.D.; Silverman, H.F.; “A linear closed-form algorithm for source localization from time-differences of arrival,” *Signal Processing Letters, IEEE*, pp. 1-4, 2008.