

Stony Brook University



OFFICIAL COPY

The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.

© All Rights Reserved by Author.

**TRACKING A VARIABLE NUMBER OF
TARGETS BASED ON RSS MEASUREMENTS
IN WIRELESS SENSOR NETWORKS**

A Dissertation Presented

by

Jaechan Lim

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

August 2007

Stony Brook University

The Graduate School

Jaechan Lim

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Petar M. Djurić, Dissertation Advisor
Professor, Department of Electrical & Computer Engineering

Thomas Robertazzi, Chairperson of Defence
Professor, Department of Electrical & Computer Engineering

Mónica F. Bugallo,
Assistant Professor, Department of Electrical & Computer Engineering

Ki H. Chon,
Professor, Department of Biomedical Engineering

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

**TRACKING A VARIABLE NUMBER OF TARGETS BASED ON RSS
MEASUREMENTS IN WIRELESS SENSOR NETWORKS**

by

Jaechan Lim

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

2007

The main problem studied in this dissertation is the tracking of targets by particle filtering in wireless sensor networks, where the sensors measure received signal strength (RSS). The number of targets is assumed unknown and it may vary with time. The initial locations of the targets are also unknown as are the time-varying reference powers of the targets. The task of tracking multiple targets under the made assumptions is a very challenging task for several reasons. First, the initial localization

can be complicated. Second the received signal is a superposition of possibly several sources (targets) whose number is unknown. Third, the reference powers of the targets can also be unknown. Fourth, model selection has to be implemented at every time instant in order to choose the number of targets present in the network.

In the proposed tracking, we apply the least squares (LS) method for quick initialization of the procedure. This includes initial localization of the targets and estimation of their reference powers. Basically, we solve a nonlinear LS problem by using an iterative method. We also implement a criterion for deciding the number of targets in the sensor field. Once the number of targets and their initial parameters are determined, we apply particle filtering for performing the tracking. The algorithm allows for decrease or increase of the number of tracked targets and quick initial localization of the newly detected targets.

Contents

List of Figures	ix
List of Tables	xii
Acknowledgments	xiii
1 Introduction	1
1.1 Sensor networks	1
1.1.1 Time delay of arrival (TDOA)	3
1.1.2 Direction of arrival (DOA)	5
1.1.3 Signal energy sensor	7
1.1.4 RSS (Received signal strength)	7
1.1.5 Complete and binary measurements	8
1.2 Review of particle filtering	9
1.2.1 Monte Carlo (MC) integration	11
1.2.2 Sequential importance sampling	12
2 Signal processing problems in sensor networks	17
2.1 Detection	17

2.2	Localization/Tracking	18
2.3	Classification	18
3	Tracking of unknown number of targets	20
3.1	Known number of targets	22
3.2	Unknown number of targets	23
4	Preliminary results	24
4.1	RSS model	24
4.2	Problem Statement	25
4.2.1	State equations	25
4.3	Solution by particle filtering	26
4.3.1	Sampling importance resampling (SIR) filter	26
4.4	Experiments	26
5	Localization of a target by the ML approach	31
5.1	Summary	31
5.2	Introduction	32
5.3	RSS measurement model	33
5.4	Approximate least squares method	34
5.5	Maximum likelihood estimate	36
5.6	Simulations	36
5.7	Conclusions	40
6	When the reference powers are unknown	44
6.1	Summary	44

6.2	Introduction	45
6.3	Measurement model	45
6.4	The ratio LS method	45
6.5	The ML method	46
6.6	Simulations	47
6.7	Summary and Conclusion	48
7	Initial State Estimation	54
7.1	Summary	54
7.2	Introduction	55
7.3	The Model of the Problem	56
7.3.1	The Measurement and State Space Model in Wireless Sensor Networks	56
7.3.2	Varying Pattern of The Number of Targets	58
7.4	Initialization of a single target with a known reference power	59
7.4.1	Approximate least squares method	59
7.4.2	Initial localization of a new target in presence of multiple targets	59
7.5	Initializing a target with an unknown reference power	60
7.5.1	Performance of initial localization with an unknown reference power	64
7.5.2	Performance comparison	65
7.6	Simultaneous initial localization of multiple targets	67
7.7	Conclusions	71
8	Tracking a variable number of targets	75

8.1	Summary	75
8.2	Introduction	76
8.3	The system model	77
8.3.1	State space model and measurements	77
8.3.2	Varying patterns of the number of targets	77
8.4	Initializing a single target with known reference power	78
8.4.1	Least squares for multiple targets	78
8.5	Initializing a target with unknown reference power	78
8.6	Tracking by particle filtering	79
8.6.1	Sampling importance resampling (SIR) filter	79
8.6.2	SIR particle filter combined with initialization algorithm	80
8.7	Simulations	81
8.8	Conclusions	84
9	Conclusion	92
	Appendices	94
A	The Newton-Raphson method	94
B	Cramer-Rao Bound (CRB)	98
	Bibliography	104

List of Figures

1.1	An example of two sources with one pair of sensors showing angles of DOA.	6
1.2	A source emitting acoustic energy and sensors	8
1.3	Comparison between sensors that provide binary and complete information.	9
3.1	General target tracking situation. A : Initial situation, two targets, T_1, T_2 B : Newly appeared target T_3 , C : Number of targets in time.	21
4.1	Two targets tracked by randomly deployed sensors. The starting points of the two targets are (0,0) and (100,100).	28
4.2	The true and estimated trajectories of two targets.	28
4.3	Two targets tracked by randomly deployed sensors. The starting points of the two targets were at (0,0) and (100,100).	29
4.4	The true and estimated trajectories of two targets.	29
4.5	The root mean squared error of estimated trajectories. The peak of the error is when the two targets met each other.	30
5.1	Two-dimensional sensor field.	37

5.2	Estimated positions of the target when noise power is 0.01 dB	38
5.3	Estimated positions of the target when the noise power is 0.1 dB . . .	39
5.4	Estimated positions of the target when the noise power is 0.2 dB. . .	40
5.5	Mean of the error magnitude of the two methods.	41
5.6	Comparison between the CRB and the variances of the estimates obtained by the two methods - X coordinate.	42
5.7	Comparison between the CRB and the variances of the estimates obtained by the two methods - Y coordinate.	43
6.1	A two-dimensional sensor field.	48
6.2	Estimation results of the four methods when the noise power was 0.01 dB. There were 1000 runs.	50
6.3	Mean of error magnitude. There were 1000 runs.	51
6.4	The variance of the estimates obtained from 1000 runs - X coordinate.	52
6.5	The variance of the estimates obtained from 1000 runs - Y coordinate	53
7.1	Estimated location of a target in a two-dimensional sensor field. The noise power was 0.2 dB	65
7.2	The mean error magnitude and the variances of the estimates obtained with 1000 runs.	66
7.3	The mean error or the reference power obtained with 1000 runs. . . .	67
7.4	The variances of the estimates of the reference power obtained with 1000 runs.	68
7.5	Estimates of target location with known reference power.	69
7.6	Estimates of the target location with unknown reference power. . . .	69

7.7	The mean of the error magnitude of the location estimates obtained with 1000 runs.	70
7.8	A comparison with the CRB, X coordinate.	71
7.9	A comparison with the CRB, Y coordinate.	71
7.10	Simultaneous initial localization of two targets.	72
7.11	Simulation result of simultaneous initialization of two targets, 1000 runs.	74
8.1	Tracking two targets in a two-dimensional sensor filed.	82
8.2	Track of target 1 - X coordinate. The estimate is the mean of 1,000 runs.	83
8.3	Track of target 1 - Y coordinate. The estimate is the mean of 1,000 runs.	84
8.4	Track of target 2 - X coordinate. The estimate is the mean of 1,000 runs.	85
8.5	Track of target 2 - Y coordinate. The estimate is the mean of 1,000 runs.	87
8.6	Mean of the error for target 1 from 1,000 runs.	88
8.7	Mean of error for target 2 from 1,000 runs.	89
8.8	The reference power of target 1. The estimate is the mean of 1,000 runs.	90
8.9	The reference power of target 2. The estimate is the mean of 1,000 runs.	91

List of Tables

1.1	Sequential Importance Sampling Algorithm	14
5.1	Simulation results. MEM: mean error magnitude, CRB: Cramer-Rao bound, X: X coordinate, Y: Y coordinate, LS: least squares, ML: maximum likelihood.	39
6.1	Simulation results. Comparison of the mean of error magnitude of the four methods. (RLS stands for ratio LS.)	49
6.2	Comparison of the variances of the estimates and the CRBs [X, Y]: [X, Y] coordinate, the MKS unit of the CRB and variance is [m ²].	49
7.1	Initializing a new target in presence of multiple targets via the LS method (known reference power).	61
8.1	LS initialization of a new target for particle filtering	86
8.2	Mean error of the initialized parameters.	87

Acknowledgments

I would like to thank my academic advisor, Dr. Petar M. Djurić for not only his long-term financial support and academic advice but also for sharing his experience of life which I find priceless. I also thank all the people (laboratory mates, colleagues, and all other friends) with whom I have shared my time. The memories I have of them will last forever.

And, I thank my parents for their endless financial and moral support as well as their patience.

Chapter 1

Introduction

1.1 Sensor networks

In recent years, there has been a trend for using densely populated sensor networks for collecting and processing information for better monitoring of various phenomena and understanding of the physical world. Such deployment of a large number of sensors has been driven by the advance of technology that has made available increasingly cheap and yet powerful sensors. In practice, sensors are interconnected to form a sensor network in order to communicate and possibly exchange information among themselves. An important class of sensor networks are the wireless sensor networks. They have a very strong potential and are already becoming a dominant technology. They provide great opportunities for observing the physical world and can be used in a wide range of applications including applications in health, transportation, military, and home.

There are several kinds of sensors that are capable of sensing different type of

information such as acoustic energy, seismic energy, infrared signals and so on. The selection of the type of sensors that are used in a sensor network depends on the application. Typically, a sensor network may use several kinds of sensors deployed in the region of interest.

An important application of sensor networks is the tracking of moving targets. The problem involves the estimation of the position of one or more moving targets within the network field. There are three important categories of methods that are used for tracking (and localization, in general). The categories are defined according to the type of physical parameters read by the sensors.

The first group of methods is based on the time delay of arrival (TDOA) of the signal generated by the tracked target. According to these methods, the signal that is emitted from a source is received with different time delays by the sensors, which are proportional to the sensor-source ranges, also called range differences (RDs). The TDOA-based methods require very high accuracy of the time delay measurements in order to obtain reasonable estimates of the target locations or estimates of their trajectories and kinematic parameters (if they move). The source location is estimated from the intersection of a set of hyperboloids defined by the RD measurements and the known sensor locations. Some signal processing techniques that are used for the solution of the emitter location problem include the iterative least-squares (ILS) [1] method, the maximum likelihood (ML) estimation [2] technique, and the sequential Monte Carlo (particle filtering) algorithm.

The second group of methods is based on direction of arrival (DOA). In principle, it is the most conventional direction-finding technique. The assumption made in using this method is that the source energy is concentrated at discrete angles that

are referred to as the source directions of arrival. No scattering of energy is assumed. However, in sonar, radar, and communications, the point source assumption may be incorrect because of signal scattering that causes angular spreading of the source energy. In that case, a distributed source model is more suitable than the point source model. DOA-based methods use the difference of the phase of several source signals to estimate the direction of arrival of the signal. DOA measurements are relatively expensive because they usually require antenna arrays at each sensor node.

The last group of methods is based on the received signal strength (RSS). These methods use measurements of the sensors that represent received energy or power emitted by the target(s). The used sensors that collect such measurements are simpler and less expensive than the ones that measure TDOA or DOA.

In this report, we develop methods that use RSS measurements. In particular, we develop particle filtering algorithms for target tracking, which includes estimation of their trajectories and their kinematics. Before, we address sensors that measure RSS, we briefly review the TDOA and DOA sensors.

1.1.1 Time delay of arrival (TDOA)

One of the main drawbacks of TDOA-based methods is that they are relatively complicated because of the requirement of arrays of antennas to obtain the TDOAs. In addition, they can suffer from pulse distortions which result from multipath propagation. Nevertheless, TDOA-based methods are used in a variety of fields and various algorithms have been developed for the estimation of TDOAs between two signals.

Consider the following general discrete-time model [3]:

$$\begin{aligned} y_{0,t} &= x_t + v_{0,t} \\ y_{1,t} &= x_{t-D} + v_{1,t} \end{aligned} \quad (1.1)$$

where $y_{0,t}$ and $y_{1,t}$ are received signals by two sensors, for example microphones, x_t is the signal of interest that is a reference signal with no delay available at the first channel and x_{t-D} is D sequence time delayed signal at the second channel. The symbols $v_{0,t}$ and $v_{1,t}$ denote noises in each of the channels and they can possibly be correlated. The goal is to estimate D from a segment of observed data by the two sensors without prior knowledge about the source signal x_t or the noises. In a more general scenario, we can imagine that we have more than two sensors that collect information.

Among various algorithms, the most common method for estimation of D is the generalized cross correlation technique defined as follows [4, 5]:

$$\hat{D} = \arg \max_D \int_w W(e^{jw}) Y_0(e^{jw}) Y_1^*(e^{jw}) e^{-jwD} dw \quad (1.2)$$

where $Y_0(e^{jw})$ and $Y_1(e^{jw})$ are the discrete-time Fourier transforms of the signals $y_{0,t}$ and $y_{1,t}$, respectively, and $W(e^{jw})$ is a cross correlation weighting function. Among various weighting function suggested in the literature, the most common function results in the PHAT (PHase Transform) method. According to this method, the estimate of the delay D is obtained as follows:

$$\hat{D} = \arg \max_D \int_w \cos(wD - (\angle Y_0(e^{jw}) - \angle Y_1(e^{jw}))) dw. \quad (1.3)$$

1.1.2 Direction of arrival (DOA)

Methods based on DOA locate a source target by determining the signal's direction of arrival when a signal is detected. These methods can be applied to complex problems where, for example, more than one source exists, or when the number of sources is unknown or to even more complicated problems which involve moving sensors.

Mathematical description of the DOA problem

When applying the DOA method [6], the signals are assumed to be narrowband signals and, therefore, can be approximated as plane waves of fixed frequency. To facilitate our discussion, let us make some assumptions and define variables. Assume that there are K source targets and M pairs of sensors. We take $M=2$ in this example. The spacing between each pair of sensor is δ and the sources are far away from the sensors. The signals are narrowband and have constant frequency. The angle between the plane wave of source k and the sensor pair connecting line is equal to θ_k , for $k = 1, \dots, K$.

Let $s_{k,t}$ be the signal emitted from the k th source at time t and \mathbf{s}_t is the vector consisting of the K source signals. The vector $\mathbf{x}_{1,t}$ is composed of the received signals by the first pair of sensors at time t , and $\mathbf{x}_{2,t}$ denotes the corresponding received signal at the second pair of sensor.

The DOA is defined as the angle between each plane wave and the line connecting each sensor pair as in Fig. 1.1. It is denoted as $\theta_k, k = 1, \dots, K$. There will be K angles for each sensor pair. Given the above definitions, the measurement equation

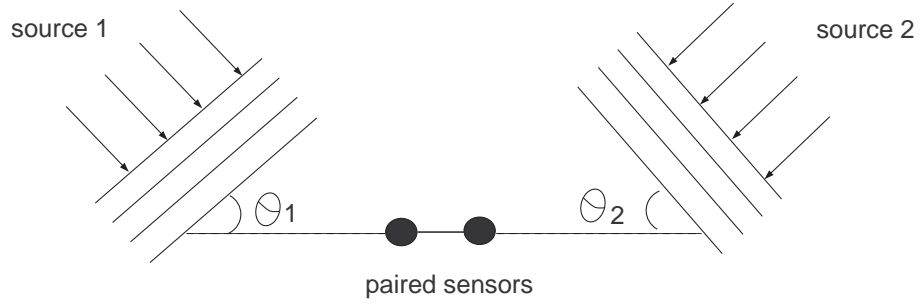


Figure 1.1: An example of two sources with one pair of sensors showing angles of DOA.

model can be stated as follows:

$$\mathbf{x}_{1,t} = \mathbf{A}\mathbf{s}_t + \epsilon_{1,t} \quad (1.4)$$

$$\mathbf{x}_{2,t} = \mathbf{A}\mathbf{\Gamma}\mathbf{s}_t + \epsilon_{2,t} \quad (1.5)$$

where $\mathbf{\Gamma}$ is a $K \times K$ diagonal matrix containing information about the phase delays between the signals received by the corresponding pairs of sensors, and \mathbf{A} is an unknown $M \times K$ matrix. The k th element of $\mathbf{\Gamma}$ is expressed as,

$$\Gamma_k = e^{j\omega\delta \sin\theta_k/c}, \quad k = 1, \dots, K \quad (1.6)$$

where c is the speed of the signal which is equal to $300,000\text{km/s}$ in free space. The goal here is to estimate $\mathbf{\Gamma}$, where \mathbf{x} , δ and the signal frequency ω are known and \mathbf{A} , \mathbf{s}_t , $\epsilon_{1,t}$, $\epsilon_{2,t}$ and K are unknown. The problem can be more complicated when the source is moving, which makes $\mathbf{\Gamma}$ a time-varying function. There are many algorithms available in the literature to find $\mathbf{\Gamma}$ for many different scenarios [7, 8].

1.1.3 Signal energy sensor

The received signal energy is one of the most common measurements obtained by sensors in practice. The received energy by a sensor is expressed as follows [9]:

$$y_{n,t} = g_n \sum_{k=1}^K \frac{s_{k,t}}{|\mathbf{r}_n - \mathbf{l}_{k,t}|^2} + v_{n,t}, \quad n = 1, 2, \dots, N \quad (1.7)$$

where $\mathbf{l}_{k,t}$ is the location of the k -th source at time instant t , n is a sensor index, t denotes time, K is the number of sources (targets), y is the sensed energy, g is the gain of each sensor, $s_{K,t}$ is the energy emitted by the k -th source at time instant t and measured at $1m$ distance from the source, \mathbf{r}_n is the location of the n -th sensor, $v_{n,t}$ is noise, and N is the number of sensors in the sensor field.

The expression of (1.7) is based on the signal energy decay model where the signal energy decays inversely proportional to the square of the distance between the source and the sensor under the conditions that the signal propagates in free and homogeneous space. Note that, unlike in the DOA problem, the signal energy propagates omni-directionally and not as a plane wave as assumed in Fig. 1.2.

1.1.4 RSS (Received signal strength)

The *Received Signal Strength* model is discussed, for example, in [10]. There, the received power is expressed in dBm. Thus, the received power signal at sensor n at time instant t is expressed as follows:

$$y_{n,t} = \sum_k \left[P_0 - 10\alpha \log_{10} \left(\frac{|\mathbf{r}_n - \mathbf{l}_{k,t}|}{d_0} \right) \right] + v_{n,t}, \quad n = 1, \dots, N \quad (1.8)$$

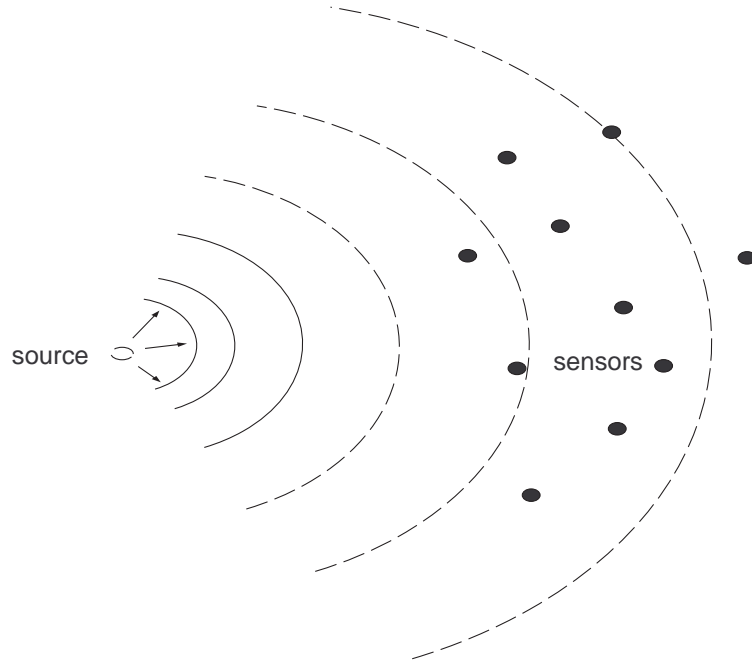


Figure 1.2: A source emitting acoustic energy and sensors .

where P_0 is the power received by the sensor at a reference distance d_0 , \mathbf{r}_n is the position of the n -th sensor, $\mathbf{l}_{k,t}$ is the location of target k at time instant t , α is an attenuation parameter that depends on the transmission medium, $v_{n,t} \sim \mathcal{N}(0, \sigma_v^2)$ is a Gaussian noise process with known variance σ_v^2 , N is the number of sensors, and K is the number of targets.

1.1.5 Complete and binary measurements

Complete measurements, also called range free measurements, mean that the sensors theoretically sense the existence of a target irrespective of its location. Binary measurements only provide information if a signal due to targets is present or not. In other words, when the signal power (energy) is above a sensing threshold, the sensor

outputs a 1, otherwise it sends a 0 as in Fig. 1.3.

Binary sensors

Suppose $y_{n,t}$ is the power received by the n -th sensor. Each sensor processes its measurements and transmits either 1 or 0 to a central unit following the following rule: The measurement is compared with a threshold, and

- if it is greater than the threshold, transmit 1,
- if it is less than the threshold, transmit a 0.

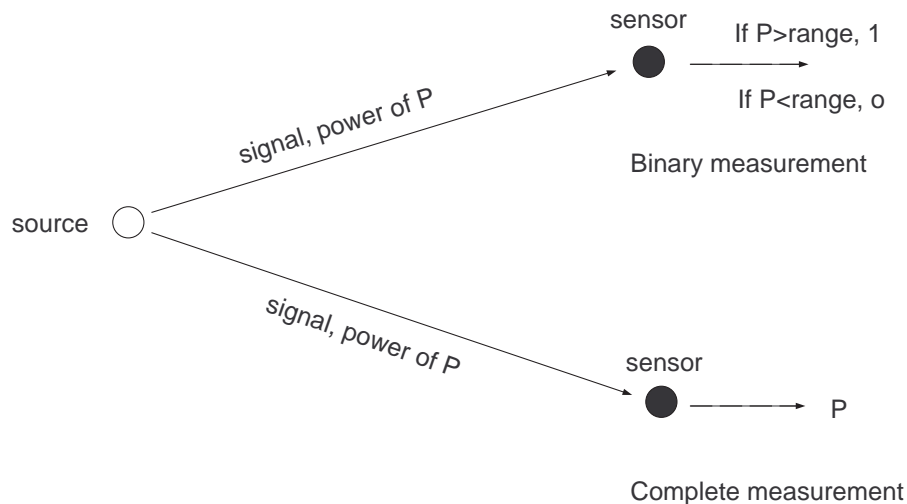


Figure 1.3: Comparison between sensors that provide binary and complete information.

1.2 Review of particle filtering

Particle filtering is a sequential Monte Carlo method based on the Bayesian methodology. It is primarily used for the estimation of the states of dynamic systems.

It is especially useful when the dynamic systems are nonlinear and/or non-Gaussian.

Many problems in statistical signal processing may be formulated in a state space form where the state transition equation expresses a hidden Markov process and the observation equation represents the observation as a function of the current state. Mathematically, these two equations can be stated as follows [11]:

$$\begin{aligned}\mathbf{x}_t &= f(\mathbf{x}_{t-1}) + \mathbf{u}_t \\ \mathbf{y}_t &= h(\mathbf{x}_t) + \mathbf{v}_t\end{aligned}\tag{1.9}$$

where \mathbf{x}_t is an unobserved hidden Markov process, \mathbf{y}_t is the observed measurement, $f(\cdot)$ and $h(\cdot)$ are generally non-linear functions, and \mathbf{u}_t and \mathbf{v}_t are random noise vectors with known distributions.

An example of a dynamic system is the Bearings only Tracking [12]. A moving object is being tracked in a two and three dimensional spaces. The hidden state \mathbf{x}_t comprises of the position and velocity of the object. The observations are bearings or angles of the object in a given coordinate system. The measurements are embedded in additive noise and are expressed by

$$y_t = \arctan(x_{v,t}/x_{h,t}) + v_t\tag{1.10}$$

where $x_{v,t}$ and $x_{h,t}$ are the vertical and horizontal coordinates in a two dimensional space. With this formulation, we would like to estimate the trajectory and velocities of the moving object given the bearings.

1.2.1 Monte Carlo (MC) integration

According to the Bayesian paradigm, the information about unknowns is contained in the posterior probability density function of the unknowns $\mathbf{x}_{0:t}$ given by $p(\mathbf{x}_{0:t}|y_{0:t})$, where $\{\mathbf{x}_t; t \in \mathbb{N}\}$ is the state sequence assumed to be an unobserved hidden process, and $\{y_t; t \in \mathbb{N}\}$ are observations which are conditionally independent given the process $\{\mathbf{x}_t; t \in \mathbb{N}\}$. Note that $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, \dots, \mathbf{x}_t\}$ and $y_{0:t} \triangleq \{y_0, \dots, y_t\}$ are the state sequence and the observations up to time t , respectively. Suppose there are N independent identically distributed (*i.i.d.*) streams $\{\mathbf{x}_{0:t}^{(i)}; i = 1, \dots, N\}$ with a distribution $p(\mathbf{x}_{0:t}|y_{0:t})$. Then, a Monte Carlo estimate of this distribution is given by [13]:

$$\hat{P}(d\mathbf{x}_{0:t}|y_{0:t}) = \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t}).^1 \quad (1.11)$$

If $I(f_t)$ as the expected value of a function $f_t(\mathbf{x}_{0:t})$ with respect to the posterior $p(\mathbf{x}_{0:t}|y_{0:t})$, its Monte Carlo estimate is given by

$$\hat{I}_N(f_t) = \int f_t(\mathbf{x}_{0:t}) \hat{P}(d\mathbf{x}_{0:t}|y_{0:t}) = \frac{1}{N} \sum_{i=1}^N f_t(\mathbf{x}_{0:t}^{(i)}). \quad (1.12)$$

Using the strong law of large numbers (SLLN), we can show that $\hat{I}_N(f_t) \xrightarrow{a.s.} I_N(f_t)$, where $\xrightarrow{a.s.}$ denotes *almost sure* convergence as $N \rightarrow \infty$ [14]. Thus, from (1.12) we can see that when obtain random *i.i.d.* samples from the posterior distribution, we can estimate $I_N(f_t)$ easily without difficulties. The convergence is dependent on N . One difficulty in this Monte Carlo method is the lack of efficient methods for generating *i.i.d.* samples from the posterior distribution, which is often multivariate,

¹The symbol $\delta_{\mathbf{x}}(d\mathbf{x}_{0:t})$ denotes the Kronecker delta function, $\delta_{\mathbf{x}}(d\mathbf{x}_{0:t}) = 1$ if $\mathbf{x} \in d\mathbf{x}_{0:t}$, otherwise $\delta_{\mathbf{x}}(d\mathbf{x}_{0:t}) = 0$.

non-standard and known only up to a proportionality constant.

1.2.2 Sequential importance sampling

Suppose that N particles $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^N$ can be drawn from a distribution $\pi(\mathbf{x}_{0:t}|y_{0:t})$, which we refer to as *importance function*. Then we can use the importance sampling identity to write

$$p(\mathbf{x}_{0:t}|y_{0:t}) = w(\mathbf{x}_{0:t})\pi(\mathbf{x}_{0:t}|y_{0:t}) \quad (1.13)$$

where the importance weight $w(\mathbf{x}_{0:t})$ is defined as

$$w(\mathbf{x}_{0:t}) = \frac{p(\mathbf{x}_{0:t}|y_{0:t})}{\pi(\mathbf{x}_{0:t}|y_{0:t})} \quad (1.14)$$

and where it is assumed that the distributions can be computed completely (including their normalizing constants). An estimate of $p(\mathbf{x}_{0:t}|y_{0:t})$ can be obtained by

$$\hat{p}_N(d\mathbf{x}_{0:t}|y_{0:t}) = \sum_{i=1}^N \tilde{w}_{0:t}^{(i)} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t}) \quad (1.15)$$

where the $\tilde{w}_{0:t}^{(i)}$'s are normalized weights obtained by

$$\tilde{w}_{0:t}^{(i)} = \frac{w(\mathbf{x}_{0:t}^{(i)})}{\sum_{i=1}^N w(\mathbf{x}_{0:t}^{(i)})}. \quad (1.16)$$

Similarly, the filtering density $p(\mathbf{x}_t|y_{0:t})$ can be approximated by

$$\hat{p}_N(d\mathbf{x}_t|y_{0:t}) = \sum_{i=1}^N \tilde{w}_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}(d\mathbf{x}_t). \quad (1.17)$$

Actually, (1.15) is an equivalent of the point mass approximation of $p(\mathbf{x}_{0:t}|y_{0:t})$ and in the case of perfect sampling, $\pi(\mathbf{x}_{0:t}^{(i)}|y_{0:t}) = p(\mathbf{x}_{0:t}^{(i)}|y_{0:t})$, and $\tilde{w}_{0:t}^{(i)} = 1/N$ for all i . If we want to obtain the estimate of $p(\mathbf{x}_{0:t}|y_{0:t})$ sequentially, we have to be able to propagate this estimate in time sequences without subsequently modifying the past time instance trajectories $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^N$. This means that $\pi(\mathbf{x}_{0:t}|y_{0:t})$ should admit $\pi(\mathbf{x}_{0:t-1}|y_{0:t-1})$ as a marginal distribution, or

$$\begin{aligned}\pi(\mathbf{x}_{0:t}|y_{0:t}) &= \pi(\mathbf{x}_{0:t-1}|y_{0:t-1}) \times \pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, y_{0:t}) \\ &= \pi(\mathbf{x}_0|y_0) \prod_{t=1}^n \pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, y_{0:t}).\end{aligned}\tag{1.18}$$

Such importance function allows for recursive evaluation of the importance weights in time when successive observations y_t become available. The importance weights w_t can be evaluated recursively as

$$w_t = w_{t-1} \times \frac{p(y_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{\pi(\mathbf{x}_t|\mathbf{x}_{t-1}, y_t)}.\tag{1.19}$$

One can summarize the above concept and obtain the sequential importance sampling filter given in Table 1.1. Many variants of this algorithm have appeared in the literature which are actually special cases of the general algorithm. Some of these algorithms were proposed in the early 1970 [15, 16].

There is an unlimited number of choices for the importance distribution. The only constraint is that its support should include that of $p(\mathbf{x}_{0:t}|y_{0:t})$. However, in practice one obtains quite poor performance of the algorithm when $\pi(\mathbf{x}_{0:t}|y_{0:t})$ is not chosen properly. Therefore, the selection of the importance function is of great importance in sequential importance sampling.

Table 1.1: Sequential Importance Sampling Algorithm

For time instances $t = 0, 1, 2, \dots$

1. For sample $i = 1, \dots, N$, sample $\mathbf{x}_t^{(i)} \sim \pi(\mathbf{x}_t | \mathbf{x}_{0:t-1}^{(i)}, y_{0:t})$ and define $\mathbf{x}_{0:t}^{(i)} \triangleq (\mathbf{x}_{0:t-1}^{(i)}, \mathbf{x}_t^{(i)})$.
2. For $i = 1, \dots, N$, evaluate the importance weights up to a normalizing constant:

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t | \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)} | \mathbf{x}_{0:t-1}^{(i)}, y_{0:t})}, \quad w_0^{(i)} = \frac{1}{N}. \quad (1.20)$$

3. For $i = 1, \dots, N$, normalize the importance weights:

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^N w_t^{(i)}}. \quad (1.21)$$

Degeneracy of the algorithm

The optimal choice of importance function (proposal function) is of course the posterior distribution of $p(\mathbf{x}_{0:t} | y_{0:t})$ in which case all the weights $\{w_t^{(i)}\}_{i=1}^N$ are equal. However, for all non-trivial cases, it is not possible to draw samples directly from it.

When an importance function is of the form (1.18), after few iterations sample degeneracy occurs. That is, all but only few of the particles will have weights different from zero. The *variance of the importance weights* only increases over time. Thus, propagating such samples is simply a waste of computational resources as their contribution to the final estimate is almost zero.

There are two techniques to alleviate the degeneration problem. One is to have a good choice of an importance function and the second one is to apply a procedure called resampling. Resampling can simply be explained as a procedure that allows

replicating of samples with significant weights while discarding those with almost zero weights. The resampling algorithm [17] can be implemented as follows:

Resampling Algorithm

- $[\{\mathbf{x}_t^{j*}, w_t^i, i^j\}_{j=1}^P] = \text{resample}[\{\mathbf{x}_t^j, w_t^i\}_{j=1}^P]$, where P is the number of particles.
- Initialize a cumulative distribution function (CDF): $c_1 = 0$
- For $i = 2 : P$
 - Construct a CDF according to $c_i = c_{i-1} + w_t^i$
- End For
- Start at the bottom of the CDF : $i = 1$
- Draw a starting point : $u_1 \sim \mathbb{U}[0, P^{-1}]$
- For $j = 1 : P$
 - Move along the CDF : $u_j = u_1 + P^{-1}(j - 1)$
 - While $u_j > c_i$
 - * $i = i + 1$
 - End While
 - Assign sample : $\mathbf{x}_t^{j*} = \mathbf{x}_t^i$
 - Assign weight : $w_t^j = P^{-1}$
 - Assign parent : $i^j = i$
- End For

The choice of the importance function (proposal density)

To decrease the severity of the degeneracy problem, the importance function should be chosen so that it minimizes the variance of the importance weights conditional upon the sampled trajectory $\mathbf{x}_{0:t-1}$ and the observations $y_{0:t}$. It is shown that the importance function that minimizes the variance of the weights is given by [18]

$$\pi(\mathbf{x}_t | \mathbf{x}_{0:t-1}^{(i)}, y_{0:t}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)}, y_t). \quad (1.22)$$

This proposal function is known as the optimal proposal density.

Optimal importance function

It seems that the proposal density $p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)}, y_t)$ was used for the first time in [19, 20]. The importance weight corresponding to this proposal density is given by $w_t^{(i)} = w_{t-1}^{(i)} p(y_t | \mathbf{x}_{t-1}^{(i)})$. Although this density is optimal in the sense that the variance of the resulting sample weights is minimum, there are important drawbacks in using it. First of all, it is usually difficult to draw samples from this density and second, in order to compute the importance weights of the samples, it requires the evaluation of the integral,

$$p(y_t | \mathbf{x}_{t-1}^{(i)}) = \int p(y_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)}) d\mathbf{x}_t \quad (1.23)$$

which in most cases is not trivial.

Chapter 2

Signal processing problems in sensor networks

The challenge of sensor networks is to achieve reliable performance by dynamic configuration and the collaboration of these sensor nodes that have limited sensing and communication capabilities. Some of the tasks of these sensor networks may include detection, localization, tracking and classification of objects which are of interest to diverse military, industrial and civilian applications [21].

2.1 Detection

The detection problem in sensor networks is basically a hypotheses testing problem [22]. The null hypotheses usually represents noise only and the alternative stands for signal in noise. Let us consider a Bayesian detection problem where $y_{1,t}, y_{2,t}, \dots, y_{N,t}$ is a sequence of observations and each corresponding to a specific sensor. The sensors are required to send summaries $m_{t,n}(y_{t,n})$ of their own

observations to the fusion center. The fusion center receives noisy versions of them, $u_{n,t}$, where

$$u_{n,t} = m_{n,t}(y_{n,t}) + w_{n,t} \quad (2.1)$$

with $w_{n,t}$ representing noise. The hypothesis testing problem amounts to deciding if H_0 or H_1 is true, based on the received data $u_{n,t}$ at the fusion center [23].

2.2 Localization/Tracking

In sensor networks, localization is one of the key tasks in many applications, where a number of sensor devices are used for instance, to locate one or more, static or moving targets. The goal of localization is to locate the target in the domain based on the observed data with as good accuracy as possible [24]. If the targets are of interest and are moving, then the localization problems will be a tracking problem [25]. There are many algorithms developed to solve these problems. The tracking problem is of main interest in this thesis.

2.3 Classification

Classification of objects moving through the sensor field is an important application that requires collaborative signal processing (CSP) among nodes. Given the limited resources of nodes, a key constraint is to exchange the least amount of information between them and still achieve desired performance. The information exchanged between nodes may be based on correlated or independent measurements. The former requires *data fusion* for optimal performance - exchange of (low

dimensional) feature vectors that yield sufficient information for discrimination. For the latter, *decision fusion* is sufficient - exchange of local likelihood values (scalars) computed from individual measurements. Generally, the measurements are a mixture of correlated and independent components so that a combination of data and decision fusion between nodes is necessary.

For example, if we consider that there are M classes of targets and assume that \mathbf{y} denotes a vector of measurements, which under hypothesis H_j , $j = 1, \dots, M$, is modeled as [26]

$$H_j \quad : \quad \mathbf{y} = \mathbf{s}_j + \mathbf{w}, \quad j = 1, \dots, M \quad (2.2)$$

where \mathbf{s}_j denotes a signal corresponding to the j -th class of targets, and \mathbf{w} denotes additive noise with some known distribution, we can define a classifier C that maps the vector \mathbf{y} to one of the target classes. If we further assume that all the classes are equally likely, we can use the maximum likelihood (ML) classifier (which is a special case of the maximum a posteriori (MAP) classifier) as the optimal classifier which has the form

$$C(\mathbf{y}) = \arg \max_{j \in \{1, \dots, M\}} p_j(\mathbf{y}) \quad (2.3)$$

where $p_j(\mathbf{y})$ denotes the likelihood function for j -th class.

Chapter 3

Tracking of unknown number of targets

Target tracking is a crucial problem in sensor networks. Before target tracking can begin, one has to (a) detect the targets in the sensor field and (b) estimate their initial locations (localization). Another related problem is classification of the targets. A brief description of these problems was provided in Chapter 2.

Consider the following scenarios of target tracking in a sensor network field [27]:

1. Multiple targets may be present in the sensor field.
2. The targets may appear and disappear with time. In other words, the number of targets is a time varying function.
3. The targets may maneuver.

For example, in Fig. 3.1, two targets are being tracked initially, but there is a newly appeared target later during tracking. Two types of situations are discussed in the

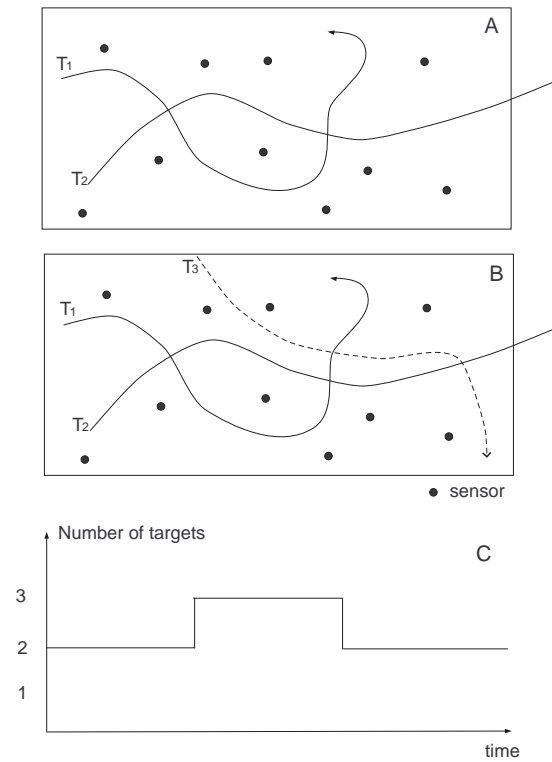


Figure 3.1: General target tracking situation. A : Initial situation, two targets, T_1, T_2
B : Newly appeared target T_3 , C : Number of targets in time.

following sections; one where the number of targets is known and another, when the number of targets is unknown.

3.1 Known number of targets

If we know the number of source targets, then detection is not necessary. Tracking of multiple targets is a lot more difficult and complicated than tracking a single target. The conventional *joint probabilities data association filter* algorithm [28] is well known for solving multi-target tracking problems.

Joint Probabilities Data Association Filter (JPDAF)

Data association is important in multiple target tracking when the sensor usually measure signals from one source. Typical algorithms to deal with such problems are the joint probabilities data association filter (JPDAF) proposed in [28]. The basis of JPDAF is the calculus of joint probabilities between the measurements and the tracks. The algorithm assigns weights for reasonable measurements to update the tracks. Kalman filtering is applied for estimation of the states of the individual objects. While the Kalman filter has been shown to provide highly efficient state estimates, it is restricted to Gaussian distributions of the noise processes in the state-space equations. In the case of a single target, this filter is known as probabilities data association filter (PDAF). As we have discussed, more recently, particle filters have been introduced to estimate non-Gaussian, non-linear dynamic processes [18]. The major advantage of this technique lies in the ability to represent multi-modal state densities. One way to apply particle filters to the problem of tracking multiple targets is to estimate the combined state space. Unfortunately, the complexity of this

approach grows exponentially with the number of objects to be tracked. One can take advantage of particle filters and the efficiency of existing approaches for multiple target tracking. There are methods where particle filters are used to track the states of the source targets and JPDAFs are applied to assign the measurements to the individual source targets [29].

3.2 Unknown number of targets

If the number of targets is known and constant over time, the state-space (spanned by the individual state-spaces of all targets) has a known and constant dimensionality. However, if the number of targets is unknown or varies over time, the number of targets, K is a random variable and a part of the state space. Since the dimensionality of the state space varies with K , it is not trivial to compare two states corresponding to different values of the random variable K . One way to address this problem is to estimate K separately from the rest of the state space, and then given this, estimate the other state variables [30]. Another approach [31] is to evaluate the likelihood in a space of constant dimensionality, thus avoiding the problem of comparing spaces of different dimensionality. However, the problem can also be addressed by employing finite set statistics (FISST) [32, 33], which is an extension of Bayesian analysis to incorporate comparisons of between state spaces of different dimensionality. Thus, a distribution over K can be estimated with the rest of the state space. FISST has been used extensively for tracking problems, mainly implemented as a set of Kalman filters.

Chapter 4

Preliminary results

In this chapter we show some preliminary results of target tracking by particle filtering. The prior density is chosen as the importance density and resampling is applied at every time step. The algorithm is applied to track up to two source targets. It is assumed that the number of targets has been correctly detected and that the initial locations of targets are known.

4.1 RSS model

Here we use the RSS model discussed in Section 1.1.4. For convenience, we rewrite the measurement equation, i.e.,

$$y_{n,t} = \sum_{k=1}^K \left[P_0 - 10\alpha \log_{10} \left(\frac{|\mathbf{r}_n - \mathbf{l}_{t,k}|}{d_0} \right) \right] + v_{n,t}, \quad n = 1, \dots, N$$

where $v_{n,t}$ is zero-mean Gaussian noise with variance σ_v^2 . This expresses the received signal power by the n -th sensor at time t . So, the observation vector will be as

follows:

$$\mathbf{y}_t = [y_{1,t} \ y_{2,t} \ \cdots \ y_{N,t}]^\top. \quad (4.1)$$

Note that here we assumed that the emitted signal powers of all the targets are the same.

4.2 Problem Statement

Two targets are moving with random accelerations from the starting points (0,0) and (100,100). Our objective is to track the two targets by particle filtering. The sensors are distributed randomly over the sensor field.

4.2.1 State equations

The state equations are written as follows:

$$\boldsymbol{\alpha}_{k,t} = \boldsymbol{\alpha}_{k,t-1} + \mathbf{w}_{k,t} \quad (4.2)$$

$$\mathbf{u}_{k,t} = \mathbf{u}_{k,t-1} + \boldsymbol{\alpha}_{k,t-1}T \quad (4.3)$$

$$\mathbf{l}_{k,t} = \mathbf{l}_{k,t-1} + \mathbf{u}_{k,t-1}T + \frac{1}{2}\boldsymbol{\alpha}_{k,t-1}T^2 \quad (4.4)$$

where $\boldsymbol{\alpha}_{k,t}$ is the acceleration of the k -th target at time t and is a *hidden state*. The velocity of the k -th target is denoted by $\mathbf{u}_{k,t}$, and its location by $\mathbf{l}_{k,t}$. Note that the location and the velocity are functions of $\boldsymbol{\alpha}_{k,t}$ according to the law of classical mechanics. The noise $\mathbf{w}_{k,t}$ is distributed uniformly between W_{max} and W_{min} , and T is the sampling time interval.

4.3 Solution by particle filtering

4.3.1 Sampling importance resampling (SIR) filter

We use the prior $p(\boldsymbol{\alpha}_t | \boldsymbol{\alpha}_{t-1}^{(i)})$ as the proposal density so that we generate particles according to (4.2), and we do resampling at every time step. Therefore, the non-normalized weights are computed by

$$w_t^{(i)} \propto p(\mathbf{y}_t | \boldsymbol{\alpha}_t^{(i)}). \quad (4.5)$$

If we define,

$$\mathbf{g}_t^{(i)} = [g_{1,t}^{(i)}, \dots, g_{N,t}^{(i)}]$$

where

$$g_{n,t}^{(i)} = P_0 - 10\alpha \log_{10} \left(\frac{|\mathbf{r}_n - \mathbf{l}_{t,k}^{(i)}|}{d_0} \right)$$

the weights will be proportional to

$$p(\mathbf{y}_t | \boldsymbol{\alpha}_t^{(i)}) = (2\pi\sigma_v^2)^{-\frac{N}{2}} \exp \left[-\frac{1}{2\sigma_v^2} (\mathbf{y}_t - \mathbf{g}_t^{(i)})^\top (\mathbf{y}_t - \mathbf{g}_t^{(i)}) \right]. \quad (4.6)$$

4.4 Experiments

All the units are expressed in the MKS system. The simulations were done with 40 time sequences, 1,500 particles, and 500 sensors. The sensors were randomly distributed within the range of the target tracks. Namely, we first generated the trajectories and then we randomly drew the locations of the sensors from a

uniform distribution within the range of the trajectories. The other parameters were $P_0 = 4.0488(dB)$, $\alpha = 2, 3$ and $d_0 = 1m$.

Targets with trajectories that do not cross

Two experiments were conducted. In the first, two targets never crossed from the initial time instant to the end of the tracking. Fig. 4.1 shows two trajectories of targets and the locations of the sensors, and Fig. 4.2 shows the estimated trajectories. As can be seen, the tracking was accurate.

Problems arise when the two targets meet each other as in Fig. 4.3. It is clear that according to (4.5), if the targets have the same power and they come very close to each other, one may not be able to discriminate which one is which. If the targets have discriminating features, for example different emitting powers and/or different state transition equations, confusion may not happen. For example, in our simulation, each target had different kind of $\mathbf{w}_{k,t}$. The state noise of one of the targets was uniformly distributed and of the other, it was normal. Fig. 4.4 shows an example when the targets were tracked well even though they had crossing trajectories. Fig. 4.5 shows the mean squared error of the estimated trajectories. Note that the peak point of the error was at the moment when the two targets crossed each other.

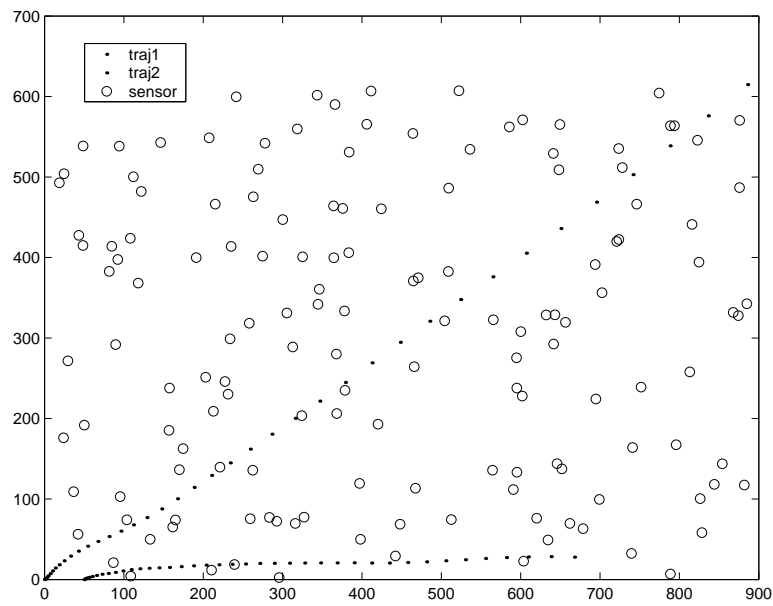


Figure 4.1: Two targets tracked by randomly deployed sensors. The starting points of the two targets are $(0,0)$ and $(100,100)$.

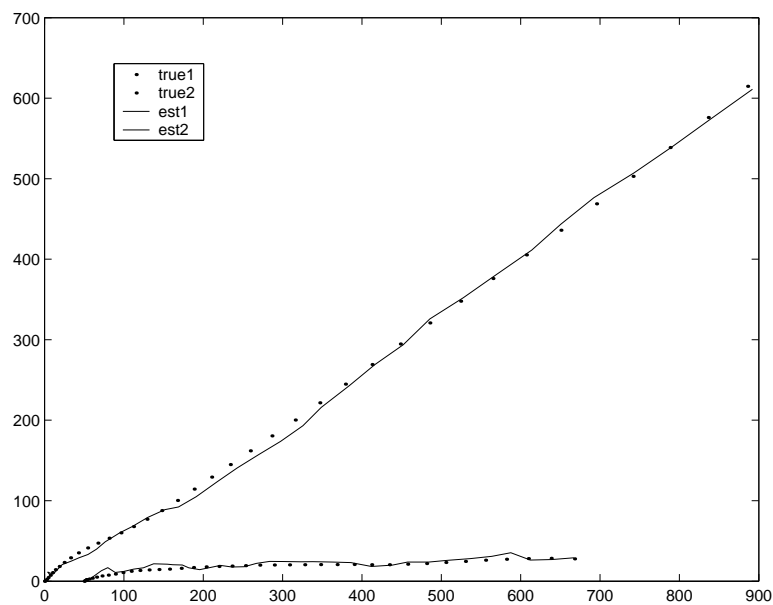


Figure 4.2: The true and estimated trajectories of two targets.

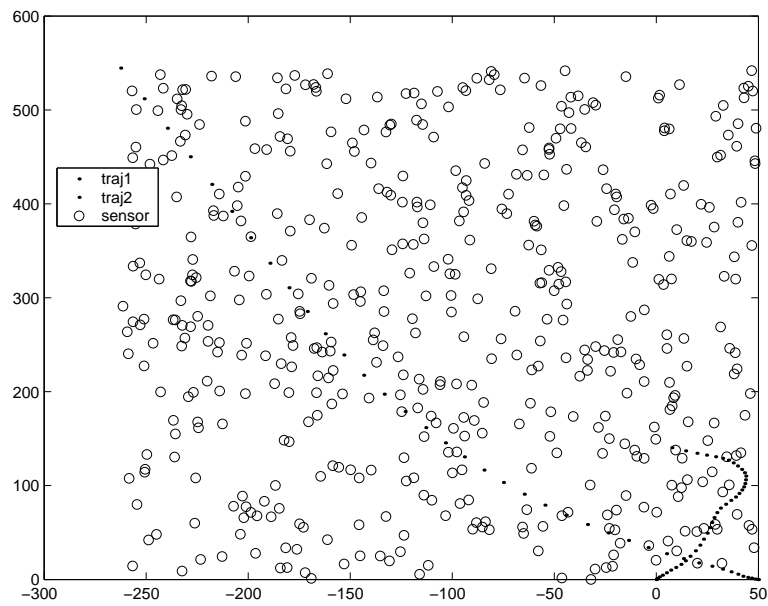


Figure 4.3: Two targets tracked by randomly deployed sensors. The starting points of the two targets were at $(0,0)$ and $(100,100)$.

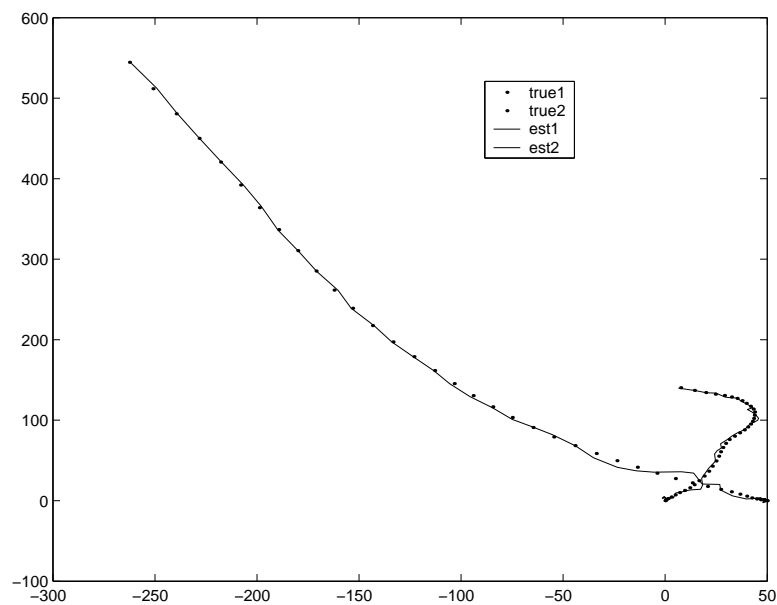


Figure 4.4: The true and estimated trajectories of two targets.

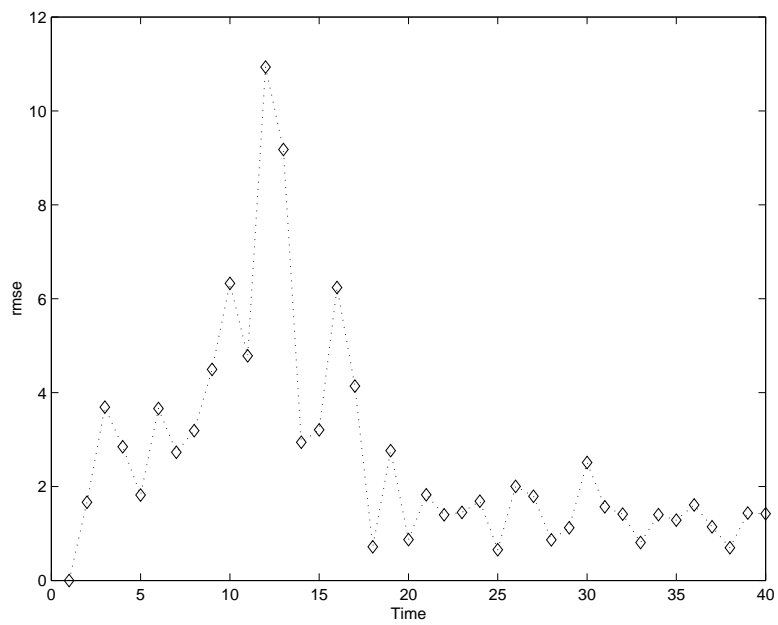


Figure 4.5: The root mean squared error of estimated trajectories. The peak of the error is when the two targets met each other.

Chapter 5

Localization of a target by the ML approach

5.1 Summary

In this chapter, a localization technique based on a multiple number of received signal strength (RSS) measurements is presented. The “least squares (LS)” method can be applied in determining the node locations in ad-hoc sensor networks or a cellular phone localization where the RSS measurement model is employed. We apply an iterative ML method to the non-linear LS problem where the measurement equations cannot be solved directly by the ML method. We adopt the solution of the LS method as an initial guess of the numerical ML method. We compare the performance of the iterative ML method to the Cramer-Rao bound (CRB). The ML approach outperforms the LS method.

5.2 Introduction

In this chapter, we propose a localization technique based on a combination of well known existing methods, i.e., the least squares (LS) and maximum likelihood (ML) method. In certain specific problems regarding localization of targets, network-based localization solution using received signal strength (RSS) is very important. For example, one has to determine the location of a node in ad-hoc sensor networks [34] or a cellular phone by networked base stations [35], and so forth. Note that in the U.S., the Federal Communications Commission (FCC) mandated wireless service providers to locate mobile phones of emergency 911 dialers (the enhanced 911, E-911) in 1996, and revised the mandate in 1999. Several methods for this problem have been proposed and discussed in recent years, such as methods based on time of arrival (TOA) [36], time difference of arrival (TDOA) [37], angle of arrival (AOA) [38], and so on. All the methods estimate the location of an object (a node or a cellular handset) relating the “object” and the “networked sensors or base stations”. The approximated LS method (linearized LS method) is a good choice for providing a solution for this problem because it does not require information about the noise distribution, and it is relatively simple and easy to apply.

In this chapter, we provide a more accurate solution for localization based on RSS measurements. We note here that the methods which do not use RSS measurements require expensive processing to estimate the necessary parameters such as “time delay” or “angle” at the step before estimating the distance or location of a source target. By contrast, the RSS model is relatively quite simple. We assume a one-to-one mapping between a target source and a sensor node or base station in the RSS model in order to apply the LS method. This is achievable by a grid spacing of the

field [39].

The RSS measurement model shows a direct relation between the location of the target and the measurement. However, we need to use a numerical ML method because it is not possible to compute the ML estimate of the location directly. An important part of the iterative ML method is finding a good initial guess of the estimate. We can acquire a good initial guess by the approximate LS method (see Chapter 5.4). We show the accuracy of the result of combining these two popular methods by comparing them with the Cramer-Rao bound (CRB) (see Chapter 5.6). We also show that our approach outperforms the approximate LS method in target localization [40].

5.3 RSS measurement model

We consider situations where N sensors make RSS measurements of a target whose location is unknown. (The problem is equivalent when we have to locate mobile phones from RSS measurements made by base stations). The received signal strength is described by a non-linear model according to [10]:

$$y_n = 10 \log_{10} \left(\frac{\Psi d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}|^\alpha} \right) + v_n, \quad n = 1, 2, \dots, N \quad (5.1)$$

where \mathbf{l} is the location of the target, n is the index of the sensor (the sensor's location is known), Ψ is the received power at a reference distance d_0 , \mathbf{r}_n is the location of the n -th sensor, α is an attenuation factor ($\alpha \geq 1$), v_n is background zero-mean Gaussian noise, and N is the total number of sensors in the sensor field.

By maximizing the likelihood function (which is obtained from this equation), we

get the ML estimate. It is clear that a closed form solution for the location \mathbf{l} can not be obtained. Even if we try to apply the LS method, the equation has to be modified so that we can use the linear LS method (see Chapter 5.4). Note that the optimized non-linear LS method is equivalent to the ML method if v_n is zero mean and white Gaussian [41]. We can obtain the ML solution by using an iterative method such as the Newton-Raphson method. For its initialization, we adopt the the solution found by the approximate LS method.

From the received measurement, we can estimate or compute the distance between the n -th sensor and the target as follows (without loss of generality, we assume d_0 is 1m and α is 2):

$$|\mathbf{r}_n - \mathbf{l}|^2 = \Psi 10^{-\left(\frac{y_n}{10}\right)}. \quad (5.2)$$

The strength of the RSS measurement drops very quickly as the distance between the sensor and the target increases. Because of that, we do not use information from sensors that are located relatively far from the target. In fact, we only use the three measurements with largest RSS.

5.4 Approximate least squares method

The LS method has been frequently applied for localization [34, 40, 42]. We adopt it for providing the initial guess of the ML iterative method. Next, we present the details of the approximate LS method in localizing a target sensor node using RSS measurements.

We estimate three distances from the noisy measurement according to (5.2), and

we denote them by r_A , r_B , and r_C , respectively. If $A(a_1, a_2)$, $B(b_1, b_2)$, and $C(c_1, c_2)$ are the known locations of the three sensors that provided the RSS measurements, the modified linear LS method of estimating the target location is based on [43]

$$\mathbf{H}\mathbf{x} = \mathbf{d} + \mathbf{e} \quad (5.3)$$

where

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} 2(a_1 - c_1) & 2(a_2 - c_2) \\ 2(b_1 - c_1) & 2(b_2 - c_2) \end{bmatrix} \\ \mathbf{d} &= \begin{bmatrix} a_1^2 - c_1^2 + a_2^2 - c_2^2 + r_C^2 - r_A^2 \\ b_1^2 - c_1^2 + b_2^2 - c_2^2 + r_C^2 - r_B^2 \end{bmatrix} \\ \mathbf{x} &= \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned}$$

and \mathbf{e} is some zero mean error.

The LS estimate is given by

$$\hat{\mathbf{x}} = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{d}.$$

When the distances are estimated from the measurements, the LS method finds as a solution a point which gives the least sum of squared differences between the function of data and the function of estimated point. When \mathbf{H} is not a full rank matrix, there is not a unique solution. In that case, we should use more measurements. In our RSS measurement model, if a sensor is far from the target sensor, the received

power is low and the information about the distance between the sensor and the target may not even be in the measurement. Therefore we use only the three best measurements, which means we use the three strongest measurements received for obtaining the initial guess for the iterative ML method.

5.5 Maximum likelihood estimate

The maximum likelihood estimator (MLE) is the most popular and practical estimator. In nonlinear problems, it may be very difficult to find the MLE from the ML function (sometimes it may be difficult to find the ML function). In those situations, we use a numerical method for obtaining the ML estimates, e.g., the Newton-Raphson or scoring method or the expectation-maximization (EM) algorithm.

From (5.1), we find the ML estimate by minimizing the cost function, $\sum \left[y_n - 10 \log_{10} \left(\frac{C}{D_n} \right) \right]^2$ where $C = \Psi d_0^\alpha$ and $D_n = |\mathbf{r}_n - \mathbf{l}|^\alpha$. The cost function is the same as the one of the LS method with zero mean white Gaussian noise [44] (note that the LS method is not the same as the “approximate LS method” that was explained in the previous section). We cannot find the analytical solution which maximizes the likelihood function, and therefore, we have to use a numerical approach. To that end we adopt the Newton-Raphson method (see Appendix A [45]).

5.6 Simulations

In this section, we show the results of the simulations where we apply the approximate LS method at the first step, and then apply the ML method using

the solution of the LS method. We compare the two methods with the Cramer-Rao bound (CRB). The CRB is derived in Appendix B.

When we apply an iterative method, we have to be very careful because the method may diverge. This may happen when the second derivative of the log-likelihood function is small [44]. Therefore, in the simulations, we set a threshold for the second derivative so that when the inverse of the derivative is larger than the threshold, the iterations stop. Then we use the LS estimate.

In the simulations, we ran 10 iterations. The two-dimensional sensor field is depicted in Fig. 5.1. The reference power was selected as 10,000 [J/s], which is

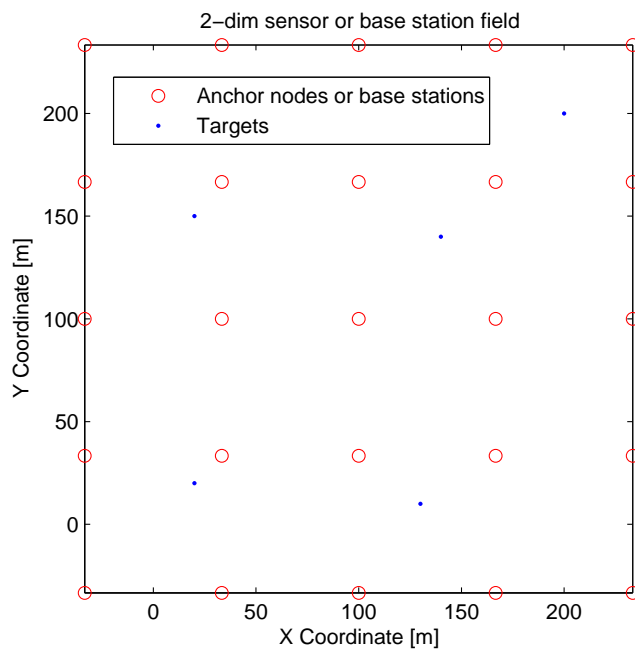


Figure 5.1: Two-dimensional sensor field.

known in practical situations (the reference power of the mobile phone or anchor node is known). In the simulations, a number of different locations of the targets

were estimated. The source targets were located at (20, 20), (130, 10), (140, 140), (20, 150), and (200, 200), respectively. There were 5×5 sensors, and the number of simulation runs was 1000. The background noises had three different values (0.01, 0.1, and 0.2 [dB]), which in practice can be known in advance.

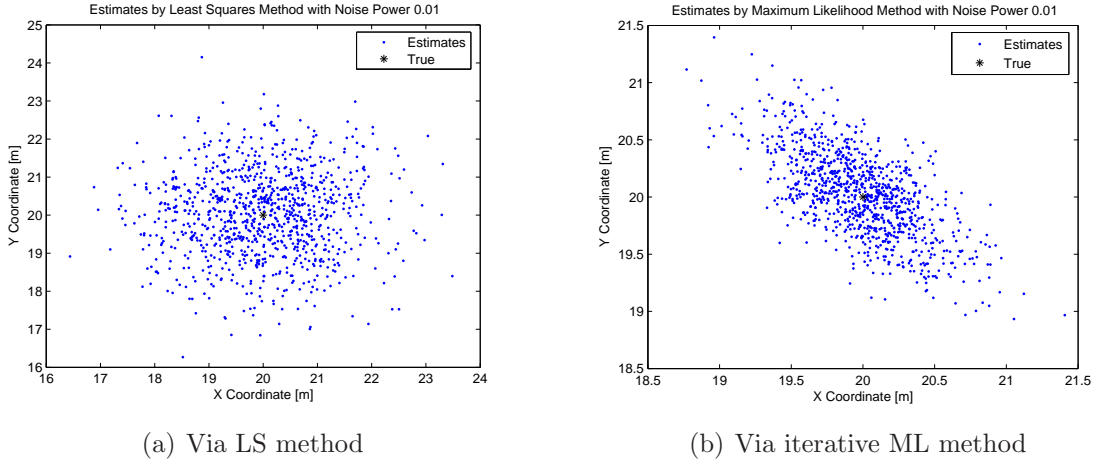


Figure 5.2: Estimated positions of the target when noise power is 0.01 dB

If we compare each pair of results when the target sensor is located at (20, 20), Figures 5.2, 5.3, and 5.4, show that the ML method had better estimates. Note that some ML iterative estimates have relatively larger errors, which most likely is due to the divergence of ML method (as we mentioned, in those cases, we stopped the iteration and used the LS estimate).

As the noise power increases, the results show that, even though the ML iterative method provides better performance, the worst estimates of both methods do not seem to be very different. The simulation results are summarized in Figures 5.5 to 5.7, and Table 5.1 where the values are averaged over all the locations of the targets. Figure 5.5 shows the mean of error magnitude of the two methods. The iterative ML

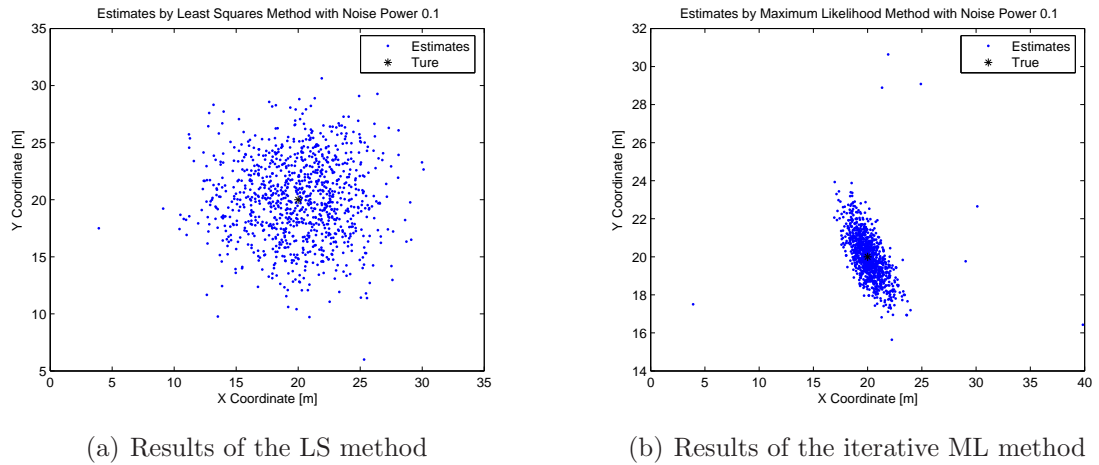


Figure 5.3: Estimated positions of the target when the noise power is 0.1 dB

methods clearly outperforms the LS method.

In Figures 5.6 and 5.7, the performances of the two methods are compared with the CRB. When the noise power is low (0.01 dB), the ML iterative method performs very close to the CRB. The exact values can be seen in Table 5.1. As the noise power increases, the performances of the methods deteriorate quickly.

Noise Power [dB]	MEM-LS [m]	MEM-ML [m]	CRB-X [m ²]	CRB-Y [m ²]	Var-LS-X [m ²]	Var-LS-Y [m ²]	Var-ML-X [m ²]	Var-ML-Y [m ²]	SNR [dB]
0.01	1.38	0.44	0.16	0.16	1.19	1.25	0.13	0.14	38.38
0.1	4.52	1.45	1.58	1.58	12.79	12.88	2.12	1.66	28.38
0.2	14.19	10.57	16.77	16.76	138.21	131.39	111.14	99.70	18.29

Table 5.1: Simulation results. MEM: mean error magnitude, CRB: Cramer-Rao bound, X: X coordinate, Y: Y coordinate, LS: least squares, ML: maximum likelihood.

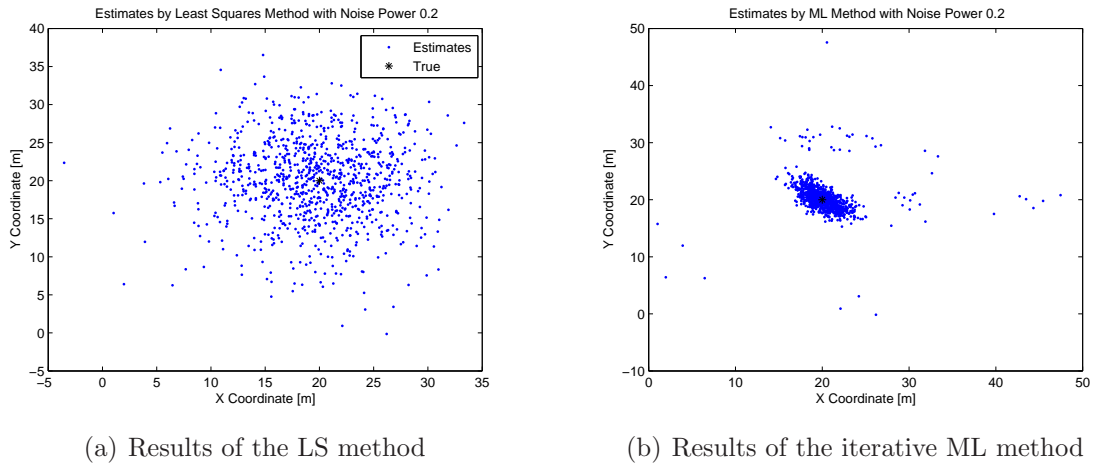


Figure 5.4: Estimated positions of the target when the noise power is 0.2 dB.

5.7 Conclusions

In this chapter, we proposed a combined LS and iterative ML technique for locating a target based on RSS measurements. The method can be applied in many scenarios such as locating mobile stations in wireless cellular systems or sensor nodes in ad-hoc wireless sensor networks. The method is simple and quick. The additional computational cost of the numerical ML method over the LS method is modest. We compared the result of the LS method-only (which is currently a prevalent method for localization) and the iterative ML method with the CRB.

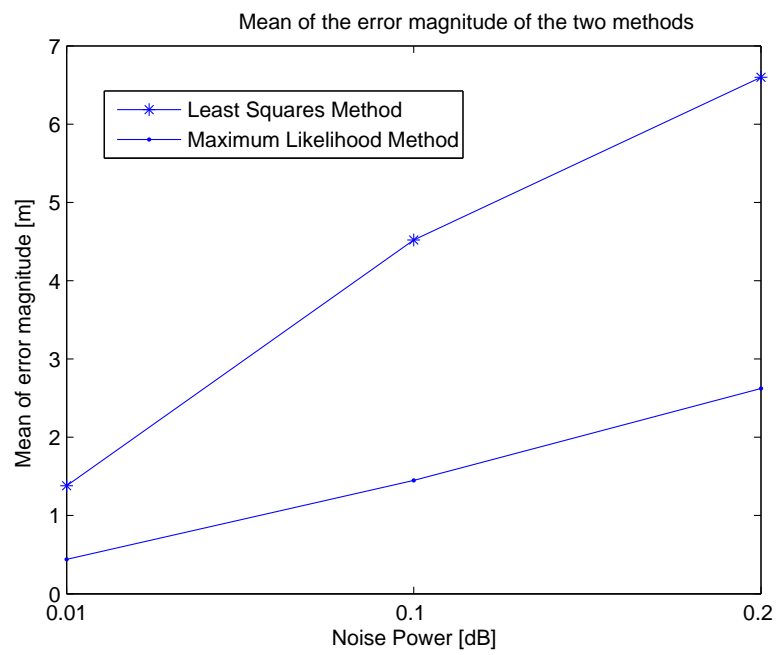


Figure 5.5: Mean of the error magnitude of the two methods.

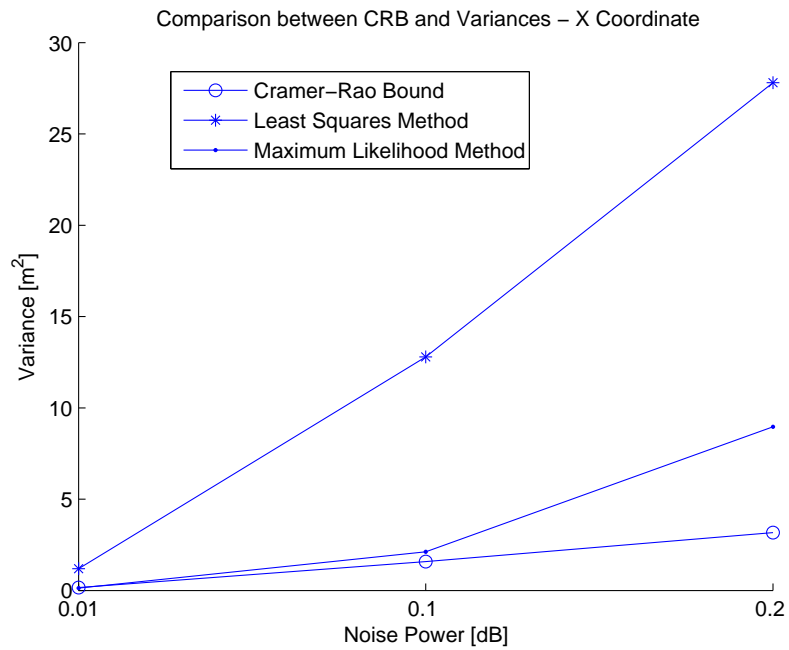


Figure 5.6: Comparison between the CRB and the variances of the estimates obtained by the two methods - X coordinate.

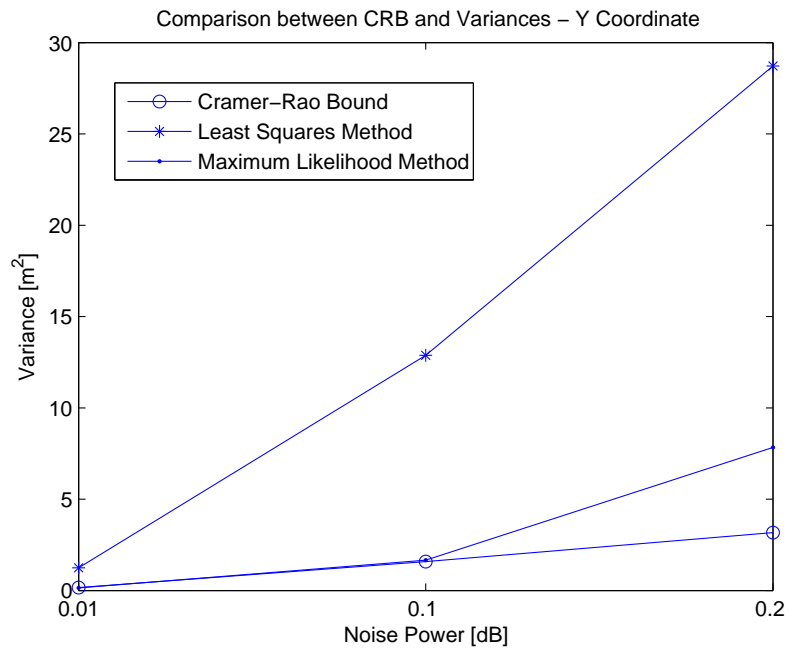


Figure 5.7: Comparison between the CRB and the variances of the estimates obtained by the two methods - Y coordinate.

Chapter 6

When the reference powers are unknown

6.1 Summary

We present a method for localization of a source based on RSS measurements where the reference signal power of the source is unknown. We compute the ML estimate by using an iterative approach which takes the LS estimate as its initial guess (as was the case in the previous chapter). The proposed method can be applied in the situations that have already been discussed. We show and compare the performances of the methods with CRBs. The results suggest that the iterative ML method that uses the “ratio LS” estimate as initial guess yields the best performance.

6.2 Introduction

The importance of the problem of localizations was discussed in the previous chapter, so we will not repeat it here. In the same vain as there, we present an iterative ML method which uses an LS estimate of the location as its starting estimate. The main difference between this method and the one from Chapter 5 is that here we relax the assumption of known reference power.

6.3 Measurement model

The measurement model is the same as before. We rewrite it here for convenience [46]:

$$y_n = 10 \log_{10} \left(\frac{\Psi d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}|^\alpha} \right) + v_n, \quad n = 1, 2, \dots, N \quad (6.1)$$

where the symbols have the same meaning as in Chapter 5. Recall that we do not use measurements from the sensors that are relatively far from the target because they do not receive strong enough signal that can provide good information about the location of the source.

6.4 The ratio LS method

We apply the LS method by taking advantage of the multiple measurements [34, 40, 42]. From (6.1), when we assume $\alpha = 2$ and $d_0 = 1$ [m], we can estimate the squared distance between a sensor and the target location by

$$|\mathbf{r}_n - \mathbf{l}|^2 \triangleq \Upsilon_n^2 = \Psi \left(10^{\frac{y_n}{10}} \right)^{-1}. \quad (6.2)$$

If we use the measurement of three sensors, the distances can be denoted by Υ_1 , Υ_2 , and Υ_3 respectively. If we denote the locations of the three sensors by (a_1, b_1) , (a_2, b_2) , and (a_3, b_3) respectively, we can find the LS estimate from (5.3) in Chapter 5.4. Note that the solution by (5.3) requires that Ψ is known,

We are able to estimate the location of the target even when the reference power is unknown. We can do it by canceling out the common “reference power factor” from the measurements of the multiple sensors, which results in a new set of equations. From (6.2), we have $\left| \frac{\Upsilon_i}{\Upsilon_j} \right| \triangleq \gamma_{ij} = \sqrt{10^{\left(\frac{y_j - y_i}{10}\right)}}$. If

$$\zeta_{ij}^2 = |\mathbf{l} - \tau_{ij}|^2$$

where

$$\tau_{ij} = \frac{\mathbf{r}_i - \gamma_{ij}^2 \mathbf{r}_j}{1 - \gamma_{ij}^2}, \quad \zeta_{ij} = \frac{\gamma_{ij} |\mathbf{r}_i - \mathbf{r}_j|}{1 - \gamma_{ij}^2}$$

it is easy to see that we can apply the LS method straightforwardly. We use $i = 1$ and $j = 2, 3, 4$, implying that now we need at least four measurements. We use the four strongest received powers.

6.5 The ML method

From (6.1), we realize that we have to use a numerical method to maximize the likelihood function because it is impossible to find the ML estimate analytically. The derivation of the ML estimator is given in Appendix A. The initial guess of the algorithm is obtained by the ratio LS method.

6.6 Simulations

We compare the performances of target location of four methods. The comparisons include the CRB which is derived in Appendix B. The sensor field is depicted in Figure 6.1.

When applying the iterative ML method, we used as initial estimates the ones obtained by the LS and the ratio LS methods. The reference power was selected as 10,000 [J/s], the source target was located at (200, 200), and the number of networked sensor nodes was 5×5 . There were 1000 simulation runs, and the background noises were chosen to have 4 different levels (0.01, 0.1, 0.5, and 1 [dB]).

During the simulation, we used a threshold for the second derivative so that when the inverse of the derivative was larger than the threshold, the iterations were stopped and for estimates we used the initial estimates. We chose 10 as the maximum number of iterations.

The estimation results of the four methods when the noise power was 0.01 dB are shown in Figure 6.2. The ratio LS method and the iterative ML method with initial guess by the regular LS method showed approximately similar “mean error magnitude” as shown in Figure 6.3. However, the ratio LS method well outperformed the iterative ML method with LS initial value when we compared them by the variances of the estimates, as shown in Figures 6.4 and 6.5.

The iterative ML method with ratio LS initial guess clearly has lower variance than the CRB as shown in Figure 6.4. It produces better performance than the computationally expensive iterative ML method with LS initial guess. The numerical result of the simulations are presented in Tables 6.1 and 6.2.

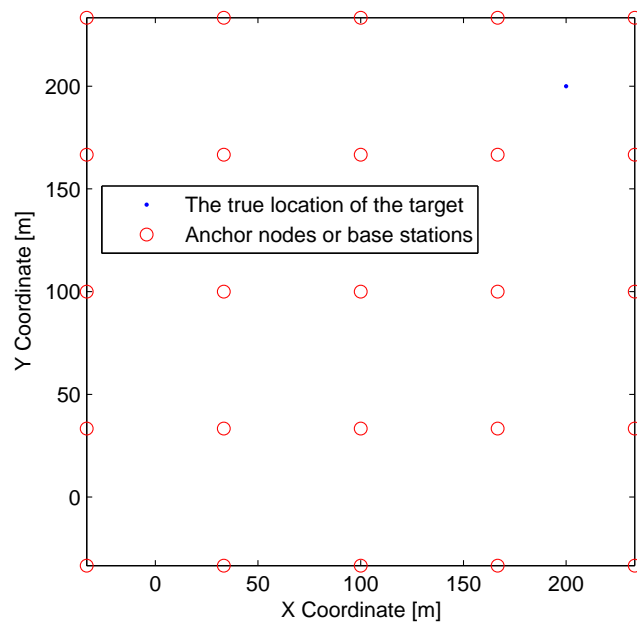


Figure 6.1: A two-dimensional sensor field.

6.7 Summary and Conclusion

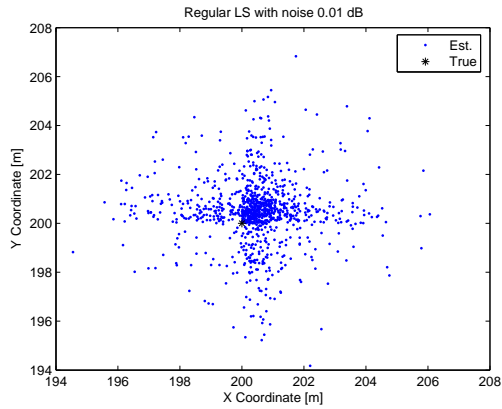
We used the ratio LS method as the initial guess of the iterative ML method to locate a target based on RSS measurements. The iterative method that takes this initial guess shows the best performance of the four methods. The proposed method can be practically applied to problems where the reference powers of the signals are unknown.

Noise Power [dB]	LS [m]	RLS [m]	ML-LS [m]	ML-RLS [m]	SNR [dB]
0.01	1.66	0.52	0.50	0.49	36.30
0.1	5.59	1.73	1.50	1.50	26.30
0.5	11.06	4.42	4.23	3.44	19.31
1	14.20	6.75	6.81	4.98	16.30

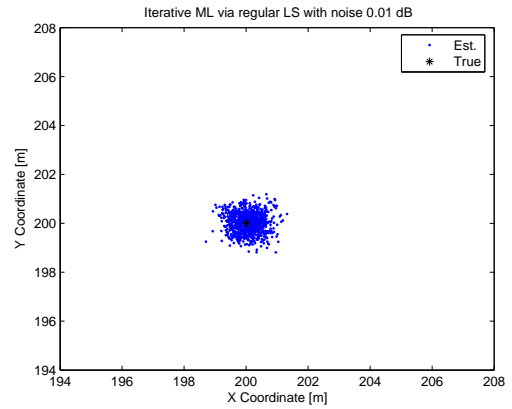
Table 6.1: Simulation results. Comparison of the mean of error magnitude of the four methods. (RLS stands for ratio LS.)

Noise [dB]	CRB-X	CRB-Y	LS-X	LS-Y	RLS-X	RLS-Y	ML-LS-X	ML-LS-Y	ML-RLS-X	ML-RLS-Y	SNR [dB]
0.01	0.22	0.22	1.89	1.73	0.19	0.18	0.16	0.16	0.16	0.16	36.30
0.1	2.21	2.21	18.73	22.94	2.00	2.06	1.39	1.53	1.43	1.54	26.30
0.5	11.05	11.05	79.32	81.24	14.55	13.27	33.75	27.24	8.21	7.32	19.31
1	22.09	22.09	127.78	129.19	35.15	32.60	69.89	78.84	16.94	14.73	16.30

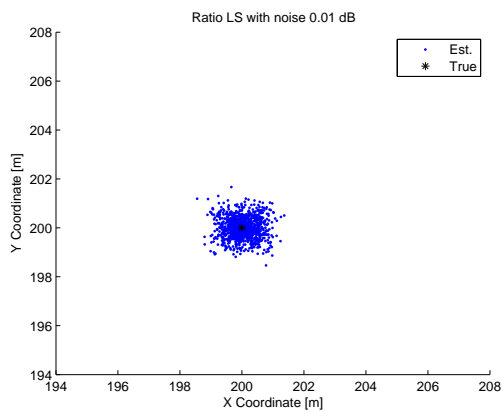
Table 6.2: Comparison of the variances of the estimates and the CRBs $[X, Y]$: $[X, Y]$ coordinate, the MKS unit of the CRB and variance is $[m^2]$.



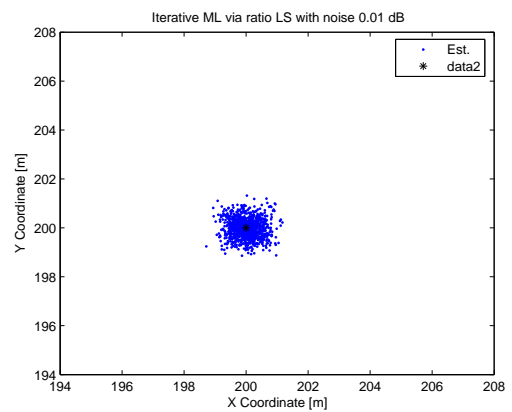
(a) Regular LS method.



(b) Iterative ML method initialized by a regular LS estimate.



(c) Ratio LS method.



(d) Iterative ML method initialized by the ratio LS estimate.

Figure 6.2: Estimation results of the four methods when the noise power was 0.01 dB. There were 1000 runs.

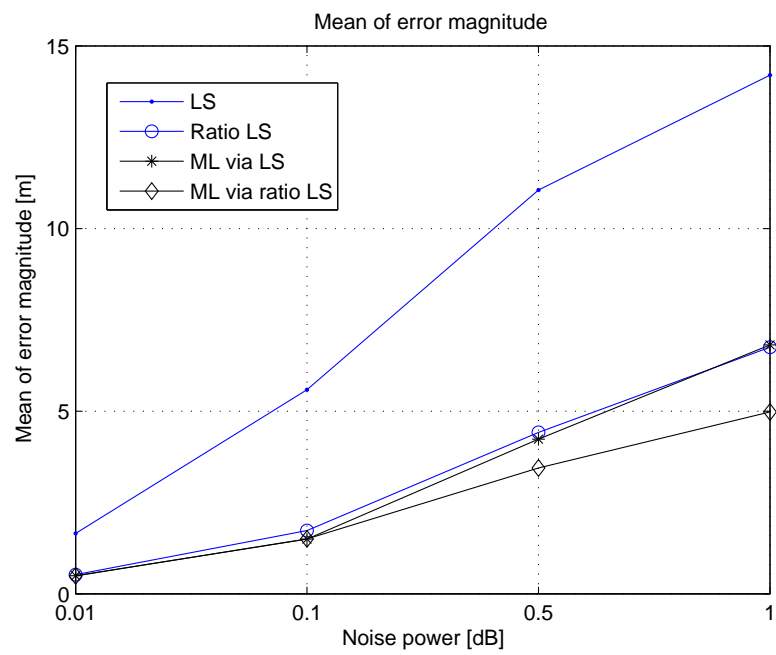


Figure 6.3: Mean of error magnitude. There were 1000 runs.

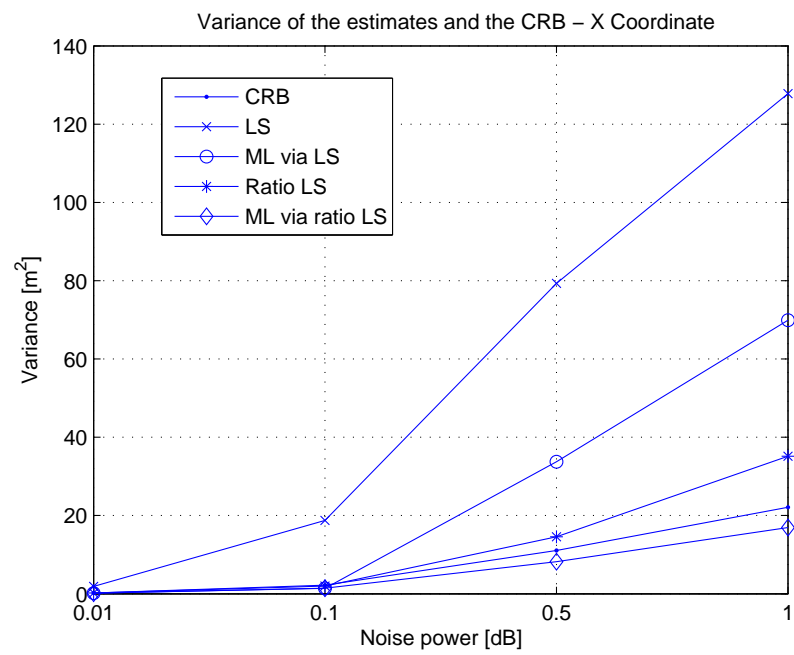


Figure 6.4: The variance of the estimates obtained from 1000 runs - X coordinate.

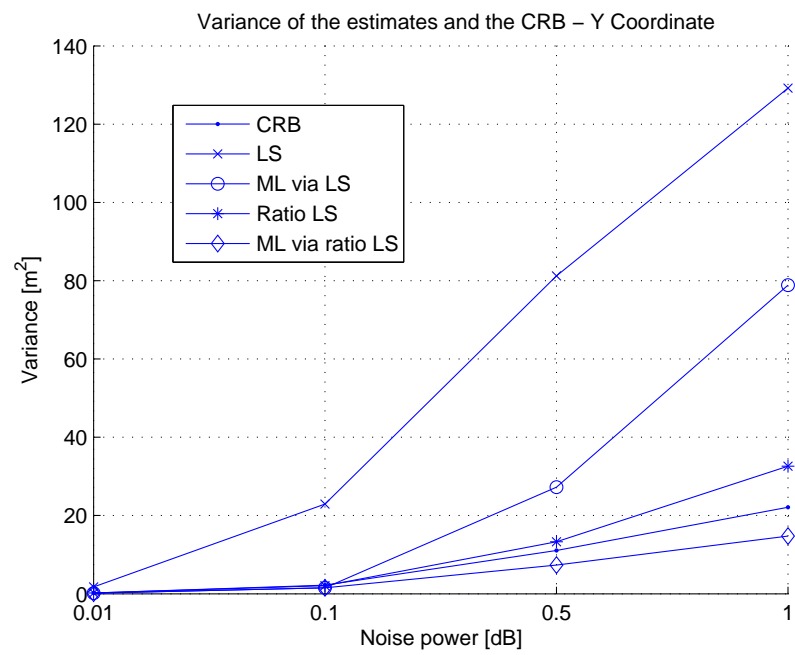


Figure 6.5: The variance of the estimates obtained from 1000 runs - Y coordinate

Chapter 7

Initial State Estimation

7.1 Summary

We present a technique for initial state estimation in a multi-target tracking scenario where the number of targets in the sensor field is unknown. Since, generally, a one-to-one mapping between RSS measurements and a target location is not possible, initializing the state of a target (location, velocity, reference power) is certainly a challenging problem. We assume that an initial estimate needs to be provided for only a single target. We apply an approximate LS method for the initial localization of a target, that is very quick and simple. Even if we may apply an optimum non-linear LS method (or equivalently, the ML method), we prefer the approximate LS method for its simplicity. Since the initial estimate is needed for tracking purposes, any tracking method (for example, particle filtering), with a reasonably good starting value will be able to improve on the accuracy of the initial estimate. This chapter focuses on the initialization of the targets in the tracking system where *particle filtering* is assumed

to be applied for tracking. Therefore, we mainly focus on estimating the location and reference power of a newly appearing target in the sensor field. We also introduce a simultaneous initialization of multiple targets.

7.2 Introduction

We mentioned that the objective is to use a quick and simple method for initializing the state of the target that needs to be estimated sequentially. In general, the approximated LS method (linearized LS) shows quite a good performance [40], and we adopt it here. It is used in mobile phone locating systems [35, 42] and for sensor node localizations in wireless ad-hoc sensor networks [34].

Previously, we have proposed a general tracking algorithm by particle filtering in wireless sensor networks where we assumed that the initial location of the target and its reference power are known [47]. Here we mainly present a detection and initial localization technique for targets with unknown reference powers.

When the reference power of the target is unknown, first we directly estimate the distance between a predicted new target and each sensor from the measurement equations. Then, we manipulate the equations so that the common reference power cancels out [48], and finally we apply the LS method.

If we have targets being tracked from the past time steps, we use predicted measurements for the continuing targets in order to estimate the distance from the possible new target to each sensor (see Table 7.1 for more details). The predicted measurements of the existing targets are generated by the particle filter according to the state space equation. Again, we do not use all the measurements from the

sensors because the strength of the received signal drops quickly as the distance from the target and the sensor increases (as can be seen from the adopted measurement model (7.1)). Therefore, we use only the four strongest sensor measurements when the reference power is unknown, and only three measurements when the reference power is known. After quickly initializing the location of the new possible target, we estimate the reference power given the location estimate (see Chapter 7.5). When initializing a single target without any other existing targets, we can apply the LS method directly and achieve a good estimate of the reference power. However, when we have existing targets, a modified approach is applied. We also show how to localize the initial location of multiple targets simultaneously in Chapter 7.6.

7.3 The Model of the Problem

7.3.1 The Measurement and State Space Model in Wireless Sensor Networks

We consider a two-dimensional sensor field with possibly randomly distributed RSS sensors. Here we repeat the RSS model [10]:

$$y_{n,t} = 10 \log_{10} \left(\sum_{k=1}^K \frac{\Psi_k d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_{k,t}|^\alpha} \right) + v_{n,t}, \quad n = 1, 2, \dots, N \quad (7.1)$$

where $\mathbf{l}_{k,t}$ is the location of the k -th target at time instant t , n is a sensor index, t denotes time, K is the number of targets, Ψ_k is the reference power of the k -th target measured at the reference distance d_0 , \mathbf{r}_n is the location of the n -th sensor, α is the attenuation factor ($\alpha \geq 1$), $v_{n,t}$ is background zero-mean Gaussian noise, and

N is the total number of sensors used in the field.

Sensors that are located closely to the source target receive strong signals while the strength decreases very quickly as the distance increases. Therefore, we do not use information from the sensors that are located relatively far from the target because they do not have very useful information about the source location. We use the best three or four sensors, that is, we use the three strongest received signals when the reference power is known and the four best measurements when the reference power is unknown. When we deal with the tracking problem, we have to be able to estimate the locations and reference powers of the targets in addition to tracking the targets, which is a challenging problem.

We model a single moving target system by a state space model as follows:

$$\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathbf{G}_x \mathbf{x}_{t-1} + \mathbf{G}_u \mathbf{u}_t \quad (7.2)$$

where $\mathbf{x}_t = [x_{1,t} \ x_{2,t} \ \dot{x}_{1,t} \ \dot{x}_{2,t}]^\top$ is the state vector which indicates the position, and the velocity of a target respectively in a two-dimensional Cartesian coordinate system, \mathbf{G}_x and \mathbf{G}_u are known matrices according to the classical mechanics defined by

$$\mathbf{G}_x = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{G}_u = \begin{pmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{pmatrix}, \quad \mathbf{u}_t = [\ddot{x}_{1,t} \ \ddot{x}_{2,t}]^\top$$

where \mathbf{u}_t is a Gaussian noise-like acceleration turbulence, and T_s is the sampling time

(s). Thus, the targets maneuver with a random acceleration based on the classical dynamics (a discrete time sampled version). Only a part of the state enters the measurement equation (the location). Nonetheless, we can estimate the complete state of the target.

7.3.2 Varying Pattern of The Number of Targets

We suppose that particle filtering is used for tracking a variable and multiple number of targets in accordance with the initialization step in this chapter. In that case, at any time instant, particles are propagated according to the dynamic state space model. We assume a certain pattern by which the number of target varies with time. In particular, we suppose that the number of targets varies in consecutive two time steps according to one of three patterns as in [47] as follows:

1. the number of targets remains the same as in the previous time step with the same identities, or
2. the number of targets increases by a newly appearing target, or
3. the number of targets decreases by one.

This assumption is realistic when the sampling time interval is small. Depending on the number of targets at the current time step, there is a certain number of hypothesized models for the next time step. We assume that each model has the same independent transition probability from the previous time step. We select the model whose particles have the largest total of particle weights.

The initialization of a new target is performed obviously only if the new model includes a new target.

7.4 Initialization of a single target with a known reference power

We initialize the location of a target using the measurements from multiple sensors. We apply the LS method to estimate the distance between the target and each sensor. We note that in the context of sequential signal processing, the localization has to be performed with a method that is fast and simple. There is no need for more accurate and computationally expensive method because the particle filtering only needs an approximate estimate. It will improve the estimate of the target's location with time.

7.4.1 Approximate least squares method

We consider a target in a two-dimensional wireless sensor network, where the RSS measurement sensor model is employed. From (7.1) and with $\alpha = 2$ and $d_0 = 1$ [m], we have (omitting the time index):

$$y_n = 10 \log_{10} \left(\frac{\Psi}{|\mathbf{r}_n - \mathbf{l}|^2} \right) + v_n, \quad n = 1, 2, \dots, N. \quad (7.3)$$

The initial location can be estimated as in Chapter 5.4.

7.4.2 Initial localization of a new target in presence of multiple targets

One of the most difficult scenario in target tracking is when the number of targets varies with time. The LS method has to be modified so that it can be applied. Suppose

the varying pattern of the number of targets follows as in Section 7.3.2. If there are already targets being tracked, and an additional target appears, we have first to cancel (remove) the contribution to the measurements of the continuing targets. We call the method residue cancelation least squares (RCLS). In that case, since we use the predicted locations of the continuing targets, the performance degrades when compared to that of tracking a single target. After we remove the contribution to the measured power of the continuing targets, we estimate the distances between a new target and each sensor. Then we select the three best sensors based on the residual powers and proceed in the usual way of estimating the location of the new target. The whole procedure is summarized in Table 7.1. If there are no continuing targets, then the RCLS is reduced to the regular LS method as in Chapter 7.4.1.

7.5 Initializing a target with an unknown reference power

If the reference power of the target is unknown, the problem is more challenging. We start the process in a similar way as when the power of the target is known. We then cancel out the power as follows:

$$\left| \frac{\Upsilon_i}{\Upsilon_j} \right| \triangleq \gamma_{ij} = \sqrt{10^{\left(\frac{y_j - y_i}{10}\right)}} \quad (7.5)$$

where the notation is the same as in the previous chapter. We can also write

$$|\mathbf{l} - \tau_{ij}|^2 = \zeta_{ij}^2 \quad (7.6)$$

Table 7.1: Initializing a new target in presence of multiple targets via the LS method (known reference power).

-
- Suppose that the number of sensors in the field is N and the number of continuing targets is K . At any time instant, we receive the measurements $y_{1:N}$, where

$$\begin{aligned}
 y_n &= 10 \log_{10} \left(\sum_{k=1}^{K+1} \frac{\Psi_k d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_k|^\alpha} \right) + v_n \\
 &\triangleq 10 \log_{10} \left[\left(\sum_{k=1}^K \frac{\Psi_k d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_k|^\alpha} \right) + \frac{\Psi_{K+1} d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_{K+1}|^\alpha} \right] \\
 &\triangleq 10 \log_{10} \left[\Gamma_n + \frac{\Psi_{K+1} d_0^\alpha}{\Upsilon_n^\alpha} \right]
 \end{aligned} \tag{7.4}$$

where

$$\begin{aligned}
 \Upsilon_n &= \left[\frac{\Psi_{K+1} d_0^\alpha}{10^{(y_n/10)} - \Gamma_n} \right]^{1/\alpha} \\
 \Gamma_n &= \sum_{k=1}^K \frac{\Psi_k d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_k|^\alpha}.
 \end{aligned}$$

The symbol Γ_n denotes the estimated or predicted part of the measurement due to the continuing targets, $\Upsilon_n = |\mathbf{r}_n - \mathbf{l}_{K+1}|$ is the estimated distance between the new target and each sensor, Ψ_{K+1} is the reference power of the new target, and $\mathbf{l}_{1:K}$ represents the predicted locations of the propagating targets from the previous time step.

- From $\Upsilon_{1:N}$, find the minimum of Υ_{min} .
 - The LS algorithm uses the information of $\{\mathbf{r}_{min}, \Upsilon_{min}\}$ by estimating the initial location of the newly appeared target by using the neighboring sensors of the best sensor (\mathbf{r}_{min}). Find the best two neighboring sensors that have shorter distances than the rest of the neighbors (we have to make sure that these three sensors do not lie on a straight line).
-

where

$$\tau_{ij} = \frac{\mathbf{r}_i - \gamma_{ij}^2 \mathbf{r}_j}{1 - \gamma_{ij}^2}, \quad \zeta_{ij} = \frac{\gamma_{ij} |\mathbf{r}_i - \mathbf{r}_j|}{1 - \gamma_{ij}^2}. \quad (7.7)$$

We use $i = 1$ and $j = 2, 3, 4$ (this means that we use the four strongest received signal powers; a different combination of sensor measurements can also be employed). The LS method is then applied to the equations obtained by forming the ratios of the power canceled equations. Once the target location \mathbf{l} is estimated, we can directly apply a method for estimating the reference power of the new target.

If we rewrite the measurement equation,

$$y_n = 10 \log_{10} \left(\frac{\Psi}{\Upsilon_n^2} \right) + v_n \quad (7.8)$$

then,

$$y_n + 10 \log_{10} (\Upsilon_n^2) = 10 \log_{10} \Psi + v_n. \quad (7.9)$$

If we set

$$y_n + 10 \log_{10} (\Upsilon_n^2) = \tilde{y}_n, \text{ and } 10 \log_{10} \Psi = \theta$$

and provided v_n is Gaussian, the ML estimates of θ and Ψ are respectively given by

$$\hat{\theta}_{ML} = \frac{1}{N} \sum \tilde{y}_n, \text{ and } \hat{\Psi}_{ML} = 10^{\left[\frac{1}{N} \sum \frac{\tilde{y}_n}{10} \right]}. \quad (7.10)$$

If a new target is initialized in the presence of multiple targets, the solution has to be slightly modified. We rewrite the measurement equation as follows:

$$y_n = 10 \log_{10} \left(\frac{\Psi}{\Upsilon^2} + \Gamma_n \right) + v_n \quad (7.11)$$

where Γ_n is the estimated (predicted) contribution to the measured power of the existing targets. Similarly to the case of a single target initialization, from (7.11), we can perform the initialization as follows. First we find

$$|\mathbf{r}_n - \mathbf{l}|^2 \triangleq \Upsilon_n^2 = \Psi \left(10^{\frac{y_n}{10}} - \Gamma_n \right)^{-1}. \quad (7.12)$$

After canceling out the power, we obtain

$$\left| \frac{\Upsilon_i}{\Upsilon_j} \right| \triangleq \gamma_{ij} = \sqrt{\frac{10^{\left(\frac{y_j}{10} - \Gamma_j\right)}}{10^{\left(\frac{y_i}{10} - \Gamma_i\right)}}}. \quad (7.13)$$

The rest of the steps are the same as in (7.6) and (7.7), where only γ_{ij} is differently formulated. In the multiple target case, for estimating the reference power of the new target we can not apply the ML method directly because we can not solve the ML function directly. This is due to the Γ_n factor. It can be easily shown that the maximum log-likelihood function has the following formula:

$$10^{\frac{\sum y_n}{10}} = \prod \left(\frac{\Psi}{\Upsilon^2} + \Gamma_n \right) \quad (7.14)$$

Therefore, we apply the LS method to estimate the power of the new target.

In summary, the only difference between target initialization with unknown and known reference powers is in the equations on which we apply the LS method and the need to use at least one more sensor measurement in the former scenario. Otherwise, we follow the same steps for the two cases. We show the performance of the initial localization method when the reference power is unknown in the following subsection.

7.5.1 Performance of initial localization with an unknown reference power

In this section, we present simulations where a target was located at $(200, 200)$, and it had four different reference powers (10, 100, 1000, and 10000 W), and the noise power was fixed as 0.2 dB. The results were obtained with 1,000 runs. The sensor field is depicted in Figure 7.1 with estimates of the target location when the reference power was 10,000 W, and the measurement was received at each sensor according to (7.1) .

The estimation results with different powers are similar. When the reference power is unknown, we use the four best sensors in applying the LS method. In the simulations, the four right top sensors were used for most of the time. We show the performance of the result in Figure 7.2 by displaying the mean error magnitudes and the variances of the estimates. Generally, the performance did not depend on the reference power except when the reference power was 10 W where the errors and the variances were the least. Therefore, the signal strength does not seem to affect the location estimate in this case. However, as shown in Figures 7.3 and 7.4, the signal strength strongly affects the performance of estimating the reference power of the target.

Interestingly, the performance is inversely proportional to the signal strength. This is caused by the error while canceling out the reference power from the measurements. This error does not affect the performance of the location estimates, but it affects the estimates of the reference power which are based on the location estimates. The reason can be easily deduced from the equations used in deriving the method.

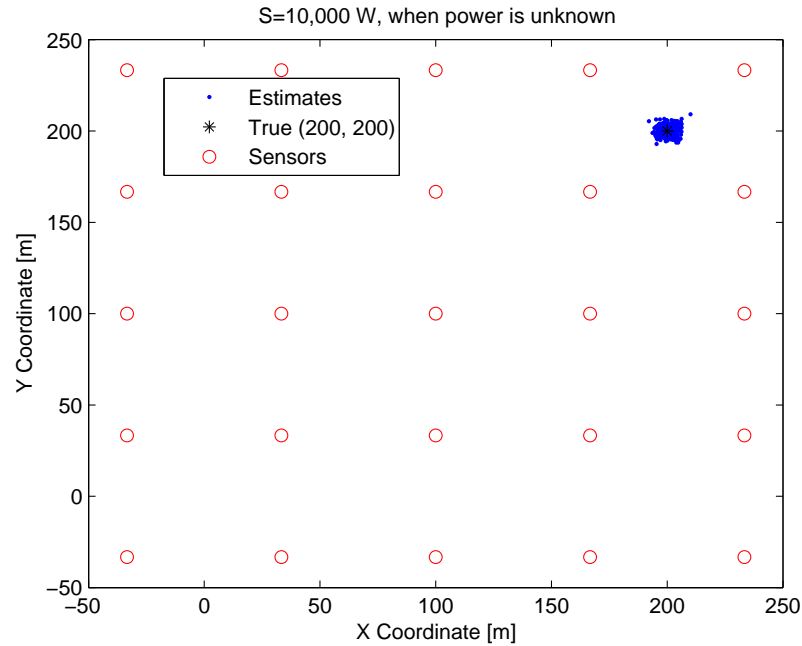


Figure 7.1: Estimated location of a target in a two-dimensional sensor field. The noise power was 0.2 dB

7.5.2 Performance comparison

In this section we compare the performances of the proposed methods for initial localization of one target when the reference power is known and when it is unknown. The simulations were performed with a fixed reference power of 10,000 W, and three different values of noise powers (0.01, 0.1, and 1 dB, respectively). The rest of the simulation conditions were the same as in the previous section when the reference powers were unknown.

The estimated target locations of 1000 runs when the noise power was 1 dB are depicted in Figures 7.5 and 7.6. The figures clearly show that the result of the method with an unknown power is better than the method with a known power. We can see

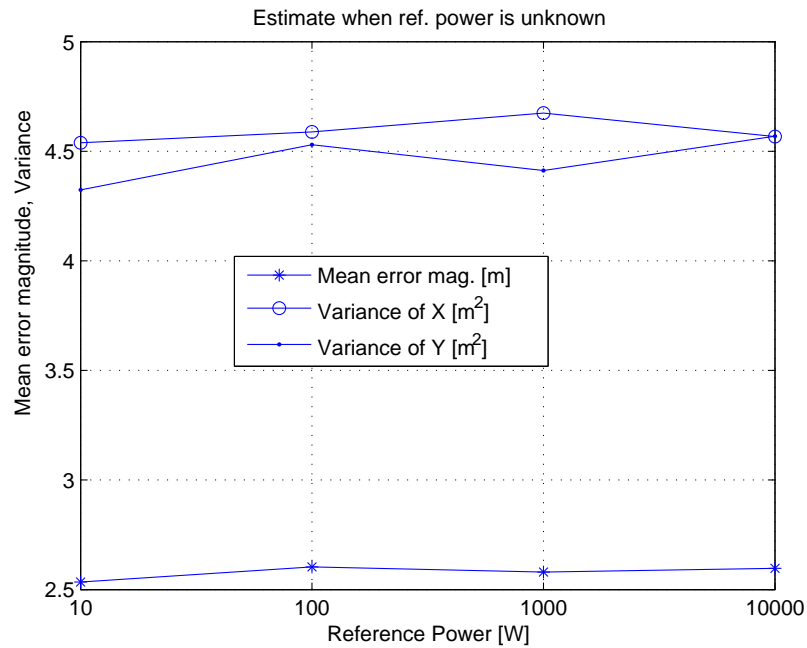


Figure 7.2: The mean error magnitude and the variances of the estimates obtained with 1000 runs.

more details of the performance comparisons from the following figures. Figure 7.7 shows that the LS method with an unknown reference power well outperforms the LS method with a known reference power, and Figures 7.8 and 7.9 show the comparison of the variances of the estimates with the CRBs. The variance of the estimates of the LS method with an unknown reference power approaches the CRB when the noise power is 0.01 and 0.1 dB quite closely. This result means that we do not have many advantages of using the regular LS method with a known reference power over the LS method with an unknown reference power. That is because even if we know the reference power of the target, we still would better use the LS method with an unknown reference power for better localization performance. The only disadvantage of applying the LS method with an unknown reference power is that we have one more

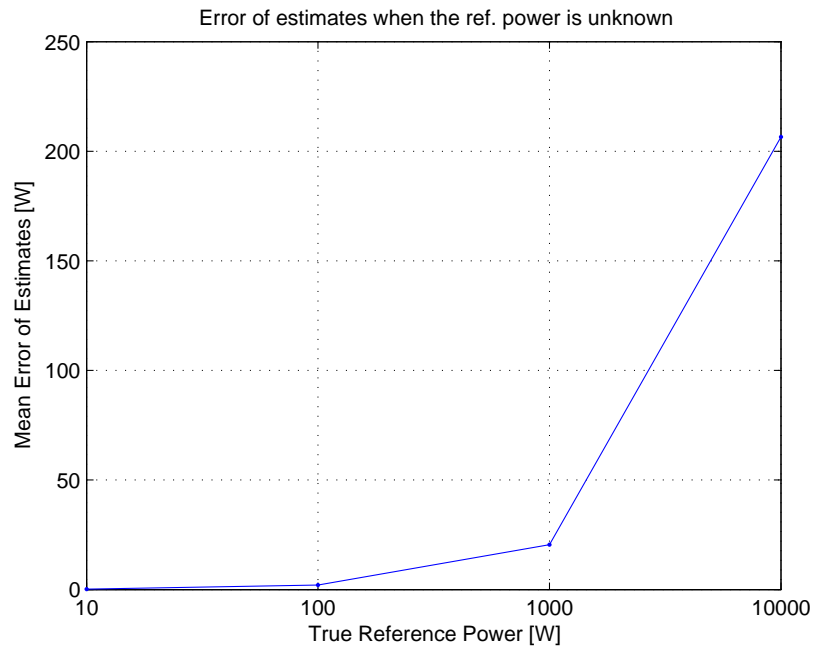


Figure 7.3: The mean error of the reference power obtained with 1000 runs.

step of canceling out the common power, and we need one more sensor measurement. Therefore, when we track multiple targets with a known reference power, we may apply the LS method for initial localization after canceling out the common power factor in the measurement. However, we need some more work to get additional insight for understanding this counterintuitive result.

7.6 Simultaneous initial localization of multiple targets

So far, we have presented initialization results of a single target in cases of a known and unknown reference power of the target. In this section we present

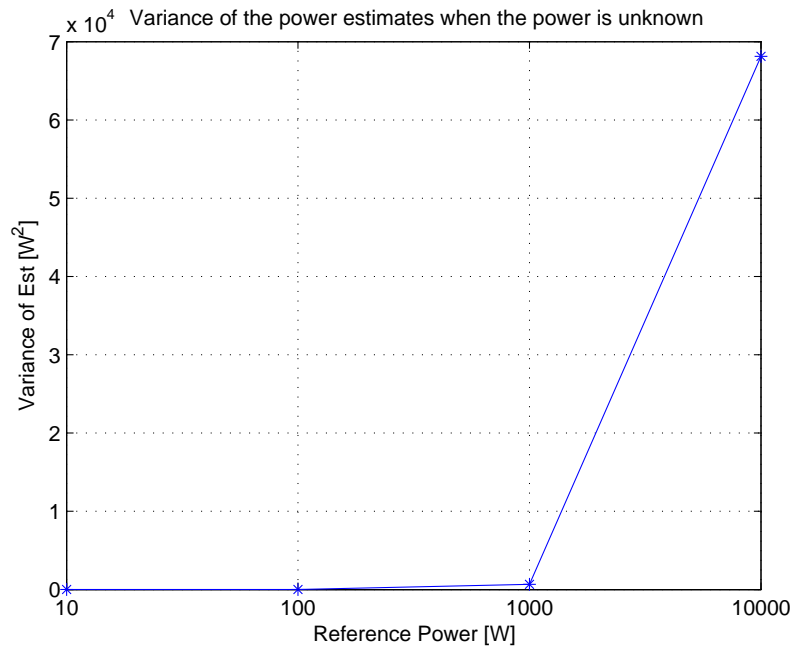


Figure 7.4: The variances of the estimates of the reference power obtained with 1000 runs.

a technique for simultaneous initialization of two targets. We take advantage of the property of the RSS measurement that the strength of the signal is inversely proportional to the log-squared distance between a sensor and a target, which means the strength of the power decreases rapidly as the distance between the sensor and the target increases. In that case, we can initialize the first target as if there was only a single target in the sensor field of interest. However, in that case, the best sensors (the best sensors mean sensors that receive the strongest signals from the target) are not neighboring to each other any more. Therefore, once we find the best sensor, we determine two more neighboring sensors and apply the LS method to initialize the first target. After the first target is localized, the procedure is the same as initializing a target in the presence of multiple targets whose details are explained in Chapter

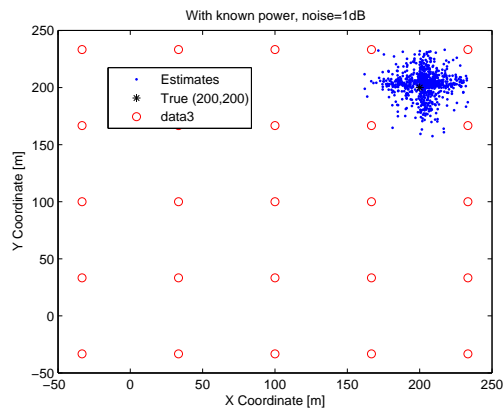


Figure 7.5: Estimates of target location with known reference power.

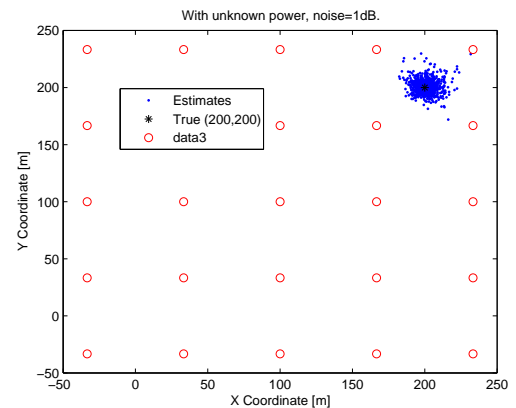


Figure 7.6: Estimates of the target location with unknown reference power.

7.4.2. The process may be iterated to produce better performance.

If the two targets are far away from each other, the method performs better than when they are in each other's proximity. The criterion of nearby or distant targets has to be studied carefully. However, we can come up with a criterion as follows. We consider that two targets are fairly close enough if the best sensors and neighboring sensors of the two targets are the same when the iterative method is performed. Two targets are distant when the three best sensors for each target are different when the iterative method is performed.

For instance, in Figure 7.10(a), the best sensors for each target are among sensors around each target. The three measurements for each LS method are completely different. In Figure 7.10(b), the best sensors that are used for the LS method are three sensors of the four sensors surrounding both targets. And these three sensors that receive the best measurements do not change as the iterative method is performed. Simulation were performed with a reference power of 10,000 W, three different noises

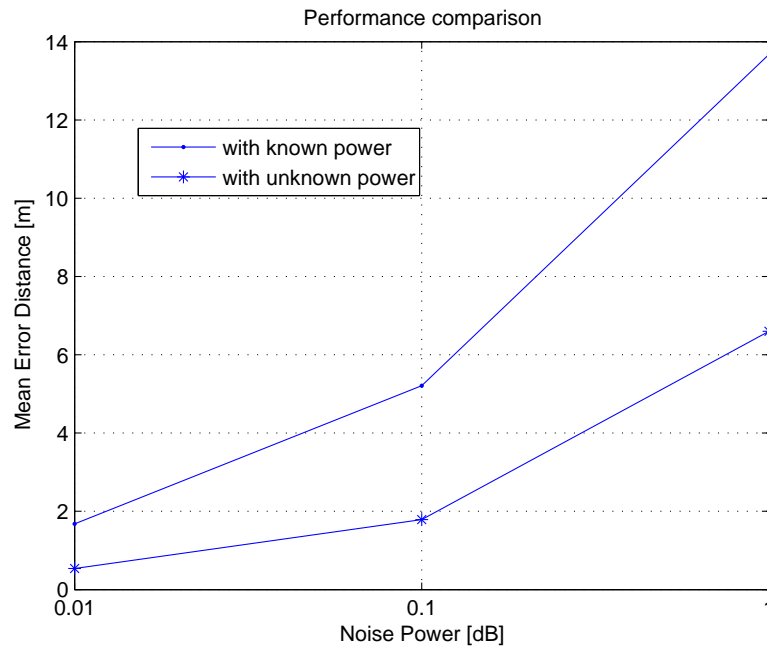


Figure 7.7: The mean of the error magnitude of the location estimates obtained with 1000 runs.

(0.01, 0.1, and 1 dB), and 7 different iteration numbers (0,1,2,3,4,10, and 100).

In this section, we focus on introducing the technique of initial localization of two targets at the same time rather than its performance. Therefore, we apply the LS method with a known reference power of the target, and we assume the reference power of both targets to be known. We may improve the performance if we apply the LS method after canceling out the common reference power even if the reference powers are known, which was shown in Section 7.5.2. We leave the joint initial localization and estimation of the reference power of multiple targets for future work.

In Figure 7.11, we compare the performances of the methods when the two targets are well separated and when they are close to each other. Generally, the results show that the initialization is far better when the two targets are distant. The effectiveness

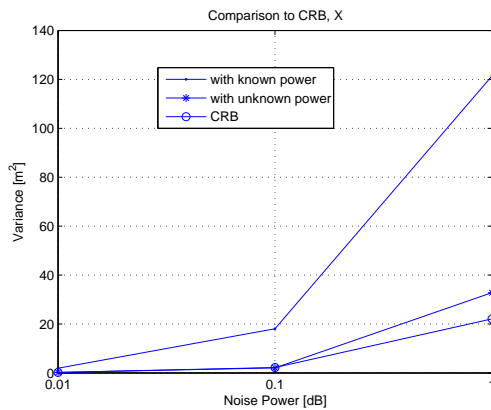


Figure 7.8: A comparison with the CRB, X coordinate.

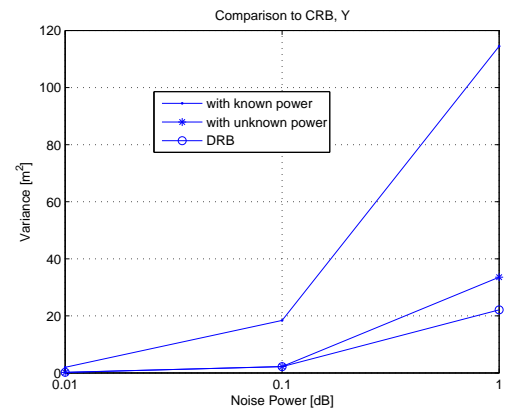
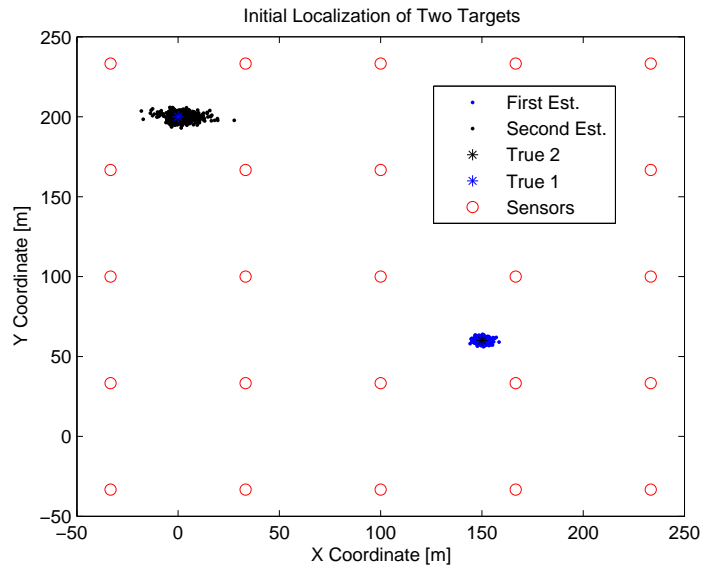


Figure 7.9: A comparison with the CRB, Y coordinate.

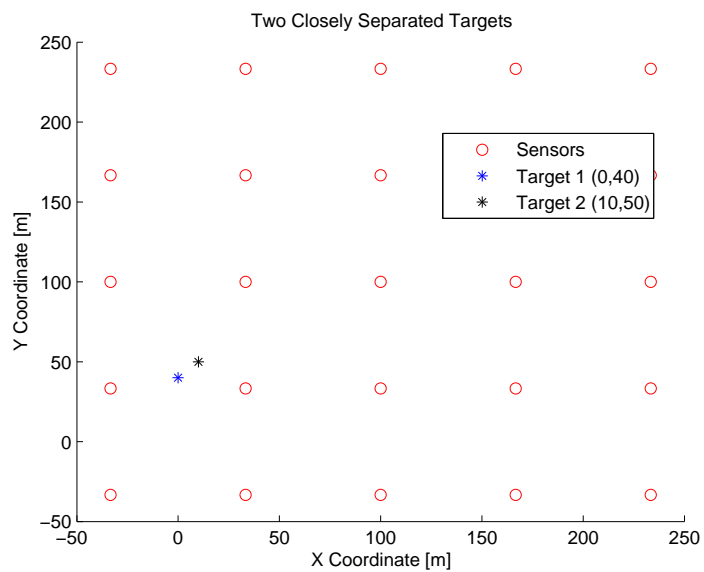
of the iterative method is only conspicuous in the mean error magnitude of the first target and when the two targets are distant, as shown in Figure 7.11(a). The estimates of the second target are not affected by the iterative method. The iterative method also decreases the two variances of the estimated locations of the targets as shown in Figure 7.11(c). However, the more iterations we use does not imply that we will have better performance. Most of the results show that the performance does not improve beyond the first iteration except for the mean error magnitude of the estimated target location when the targets are distant, as shown in Figure 7.11(b). Even though more iterations yield improved results as measured by the mean error magnitude, they do not decrease the variance of the estimates (see Figure 7.11(d)).

7.7 Conclusions

In this chapter, we introduced an initialization technique based on RSS measurements in wireless sensor networks when there are multiple targets. The



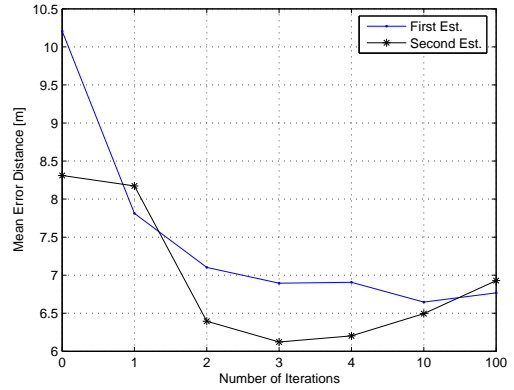
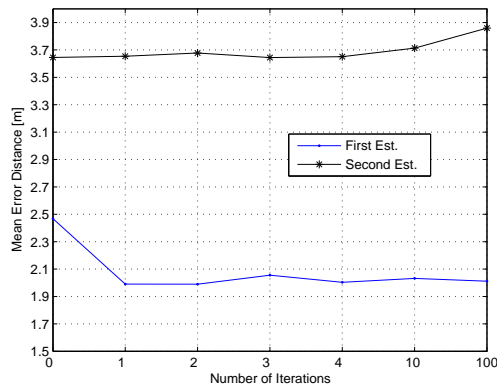
(a) Initial localization of two distantly separated targets.



(b) Initial localization of two closely separated targets.

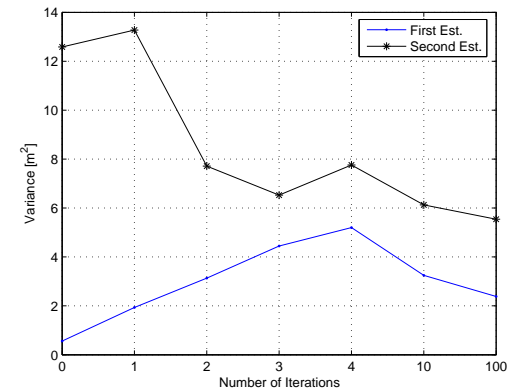
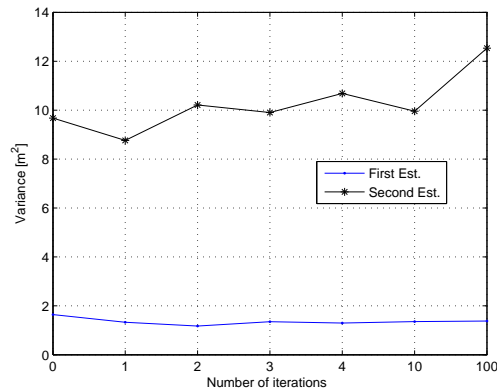
Figure 7.10: Simultaneous initial localization of two targets.

number of targets may vary, and so may the unknown reference powers of the targets. The method first assumes that there is one target, and it estimates its position. Then it proceeds to estimate the location of the second target but now using the information about the first target and so on. The process is repeated iteratively. We provided simulation results that show the performance of the proposed method when the reference powers of the targets are known. The case of unknown powers is left for future work.



(a) Mean error magnitudes when two targets are distant.

(b) Mean error magnitudes when two targets are close.



(c) Variance of the estimates when two targets are distant.

(d) Variance of the estimates when two targets are close.

Figure 7.11: Simulation result of simultaneous initialization of two targets, 1000 runs.

Chapter 8

Tracking a variable number of targets

8.1 Summary

We present target tracking in wireless sensor networks where the tracking is based on RSS measurements where the tracking is performed by particle filtering and where the number of targets is unknown and varies with time, the reference powers of the targets, too, are unknown and vary with time, and the initial locations of the target are also unknown. We combine the initialization method introduced earlier with particle filtering which tracks the states of the targets. Since it is assumed that the reference powers of the targets are unknown, we apply the LS method that first localizes the targets after canceling out the reference powers, and then estimates the reference power with estimated locations of the targets.

8.2 Introduction

In this chapter, we detect and track targets based on RSS measurements where the number of targets is unknown and varies with time. The reference powers of the targets are also unknown and vary with time. Previously, we proposed a general algorithm for tracking of multiple targets by particle filtering and based on RSS measurements [47]. Here we combine this approach with an initialization method based on the LS principle. We note that rather than very accurate initializations we need quick initializations.

If the reference power of the target is unknown and it varies with time, the problem is by far more challenging. In fact, that scenario is the main subject of this chapter and the solution is proposed in Chapter 8.5. The initialization step then comprises two sub-steps. First, we directly estimate the distance between a possible new target and each sensor from the observation equations, and then expand the equations with respect to the squared estimated distance to cancel out the common reference power before applying the LS method [48]. When we estimate the distance between the possible new target and each sensors, we use the predicted RSS of the existing targets. The predicted RSS can be computed from the particles propagated from the previous time step by the particle filter. Since all the particles are supposed to generate offspring particles that have new targets, the initialization step is applied to every single particle.

Even though many sensors are deployed in the sensor field, we do not use all the measurements from the sensors because many of the sensors are far from the targets and their measurements do not contain information about the location of the targets. In fact, as before we only use the four strongest sensor measurements and apply

the LS method when the reference power is unknown, and only three measurements when the reference power is known. After quick estimation of the location of the new possible target, we estimate its reference power given the estimated location (see Chapter 8.5). When initializing a single target without any other existing targets, we can apply the ML method directly and achieve a good estimate of the reference power. However, when we have existing targets, we can not directly apply the ML method and instead, we use an LS method to estimate the reference power. We provide computer simulations in Section 8.7.

8.3 The system model

8.3.1 State space model and measurements

The model of the problem follows what we described in Section 7.3.1. The state space model of a single target and the measurement model are given by 7.2 and 7.1.

8.3.2 Varying patterns of the number of targets

The main tracking is based on the use of a particle filter which needs initial estimates of the unknowns. When each particle propagates according to the state space model, we assume the pattern by which the number of targets changes with time as in Section 7.3.2. This assumption is similar to the birth/death model which is adopted in the literature on multiple target tracking. However, according to our model, any particle that has the information of the target state will have a move for birth, death, or unchanged number of targets, whereas the birth/death events are random in most of the literature [49].

8.4 Initializing a single target with known reference power

We already know that the LS method is not the best performer in initialization (see [50]). However, when we use particle filtering, quick and approximate initialization is quite appropriate (the LS method has this feature) because the particle filtering will improve the accuracy of the location very quickly. The procedure for initial localization of a target with the LS method is the same as before (see Chapter 5.4 for details).

8.4.1 Least squares for multiple targets

The LS method has to be modified when the number of targets varies. Suppose the varying pattern of the number of target follows as in Section 8.3.2. If there are existing targets and an additional target appears, we cancel out the portion of the measurement due to the continuing targets. After we remove the portion of the continuing target measurements, we re-estimate the distances between the new target and all the sensors. Finally, we apply the LS method in the usual way. The whole procedure is summarized in Table 7.1 of Section 7.4.2.

8.5 Initializing a target with unknown reference power

The initialization method of a new target when the reference power of the target is unknown is explained in Section 7.5. The same steps of initialization are applied

in this section for tracking of multiple targets.

8.6 Tracking by particle filtering

We apply the LS method and/or ML estimator to find the initial estimates of the unknown states of a new target which are then used by the particle filtering algorithm for tracking [51, 17, 18]. When the reference power of the target is unknown, the states of the particles have one more element, and the initialization algorithm produces the location and the reference power of the new target.

When we apply particle filtering, we need to select the best model among the possible models by summing up the weights of each model. The model which has maximum *weight sum* will be selected as the surviving model (although a random mechanism, for selection can also be used) (see Section 8.3.2) [47]. Any particles that are generated by the *particle filtering* produce offsprings of a new target [47] even if they can be removed after the model selection step or resampling [18]. Therefore the initialization step is applied to every single particle at every time step of the tracking algorithm.

8.6.1 Sampling importance resampling (SIR) filter

There are many particle filtering algorithms [17] and our choice for tracking system is the sampling importance resampling (SIR). SIR is relatively easier to apply and is generally applicable to any model. If we denote the state function, observation function, state, observation, and the weight by \mathbf{f}_t , \mathbf{h}_t , \mathbf{x}_t , \mathbf{y}_t , and w_t^i (where t is the time index and i is the particle index) respectively, the importance density,

$q(\boldsymbol{\chi}_t|\boldsymbol{\chi}_{t-1}^i, \mathbf{y}_{1:t})$ is the prior density, $p(\boldsymbol{\chi}_t|\boldsymbol{\chi}_{t-1}^i)$, and because resampling is executed at every time step, the weight is computed as

$$w_t^i \propto p(\mathbf{y}_t|\boldsymbol{\chi}_t^i). \quad (8.1)$$

8.6.2 SIR particle filter combined with initialization algorithm

In tracking of varying number of targets, the state space equation must include the state of the number of targets. We denote it by K_t at time step t . As mentioned previously, the number of targets K_t has three patterns of evolution, e.g., $K_t = K_{t-1} + 1$, $K_t = K_{t-1}$, and $K_t = K_{t-1} - 1$ [47]. Because K_t is not a random variable in our model, every single particle produces all the possible descendants. The number of descendants depends on K_{t-1} . If $K_{t-1} = 0$, then there two offsprings: $K_t = 0$ and $K_t = 1$. If $K_{t-1} = n > 0$, then the descendants will have $n - 1$, n , and $n + 1$ targets. The total number of hypotheses is $n + 1$ because when a particle produces a descendant with $n - 1$ targets, it will have n different ways of switching from n to $n - 1$ targets because all the targets have equal possibility of disappearance. When $K_t = 0$, then the state space contains only the number of targets.

The posterior function of interest will be $p(K_{1:t}, \mathcal{S}_{1:t}, \mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ (where \mathcal{S} is a vector denoting the reference powers of the target and \mathbf{x} a vector of the dynamic state parameters), and the distribution is approximated by the probability mass function, $p(\boldsymbol{\chi}_{1:t}^m) = p(\{K_{1:t}^m, \mathbf{x}_{1:t}^m, \mathcal{S}_{1:t}^m\}) = w_{1:t}^m$, where $m = 1, 2, \dots, M$, and M is the total number of the particles. We compute the weight (when we use the SIR particle filter)

by

$$w_t^m \propto p(y_{1:N,t} | \mathbf{x}_t^m, \mathbf{K}_t^m, \mathcal{S}_t^m).$$

If we use only the three best sensors and assuming that the signals picked by the sensors are not correlated, then

$$w_t^m \propto \prod_{n=1}^3 p(y_{n,t} | \mathbf{x}_t^m, \mathbf{K}_t^m, \mathcal{S}_t^m).$$

We need to apply the initialization algorithm when we generate a particle that has newly appearing target (this is the case when $\mathbf{K}_t = \mathbf{K}_{t-1} + 1$). The algorithm generates an estimated location and reference power of the target. The summary of the initialization and its use with particle filtering is given in Table 8.1. The initialization algorithm is explained in Section 8.5. If we cancel the reference power to estimate the location of the new target, we do not use neighboring sensors anymore. Note that when the reference power is known, even though we use the three best measurements for estimation of the states, the initialization algorithm has to use the sensors that are neighboring to each other (see Table 7.1) because sometimes, the three best sensors may not be neighbors to each other which may cause failure of the LS method.

8.7 Simulations

In this section, we present a simulation based on the proposed approach. We assume that the reference power of the target is unknown and varies according to $\mathcal{S}_t = \mathcal{S}_{t-1} + \mathcal{N}(0, \sigma^2)$. The initial reference powers of the targets were 200 and 180

[J/s], respectively. We used 300 particles. A target appeared at $(0, 150)$ initially, and two time steps later, another target appeared at $(200, 0)$. The two targets were in the field until $t = 62$. The second target disappeared at $t = 63$ and the first target at $t = 80$. The acceleration-noise followed a zero mean Gaussian distribution with variance one. The background noise power was 0.01 [dB], the variance of the reference power was 0.001, and there were 5×5 sensors.

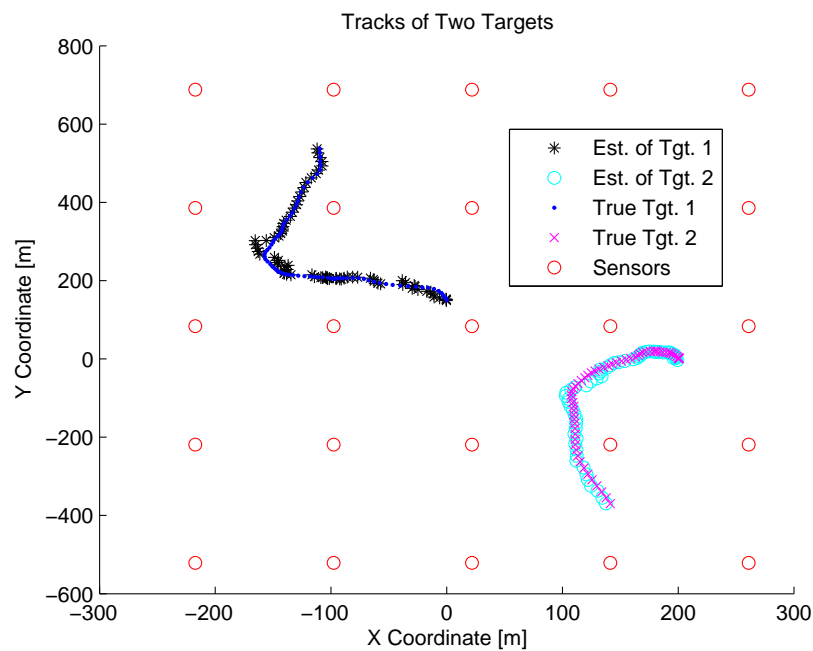


Figure 8.1: Tracking two targets in a two-dimensional sensor field.

A single simulation result is depicted in Figure 8.1. We ran 1,000 realizations. The results show that the algorithm, in general, performed well with accurate detection of a new target and disappearance of a target. When we initialized a new target, if it had too low reference power or if it was located outside the field of interest, it was removed even if it was initialized as a new target in the initialization

step.

Figures 8.2 to 8.5 show the results of 1000 runs. They compare the true tracks and the mean of the estimated tracks with respect to each coordinate of each target. Figures 8.6 and 8.7 show the mean of the estimation errors of each target, and Figures 8.8 and 8.9 present the results of 1000 runs comparing true power and the mean of the estimated power of each target. Note that the error for target 1 increases during the interval from $t = 3$ to $t = 62$ as shown in Figures 8.2, 8.3, and 8.6. The initialized parameters have very small mean errors as shown in Table 8.2.

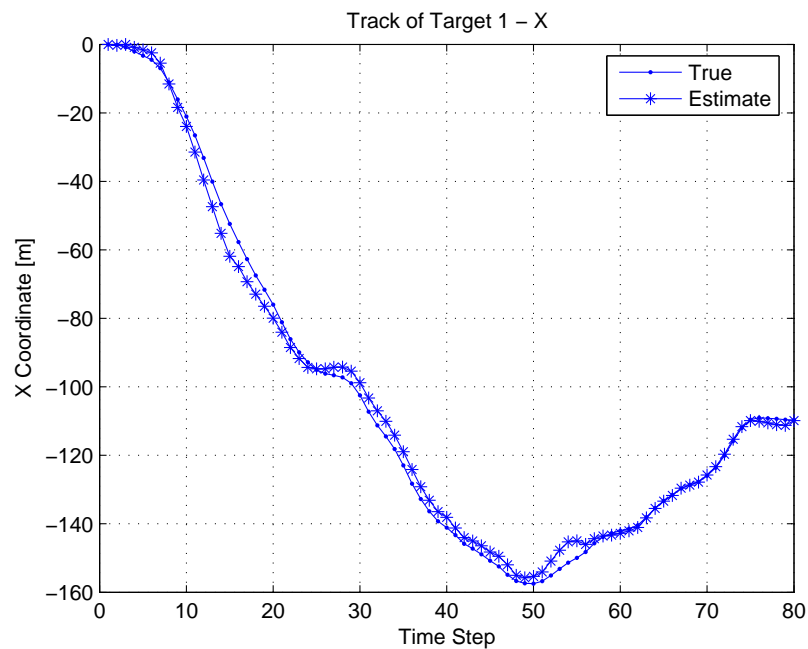


Figure 8.2: Track of target 1 - X coordinate. The estimate is the mean of 1,000 runs.

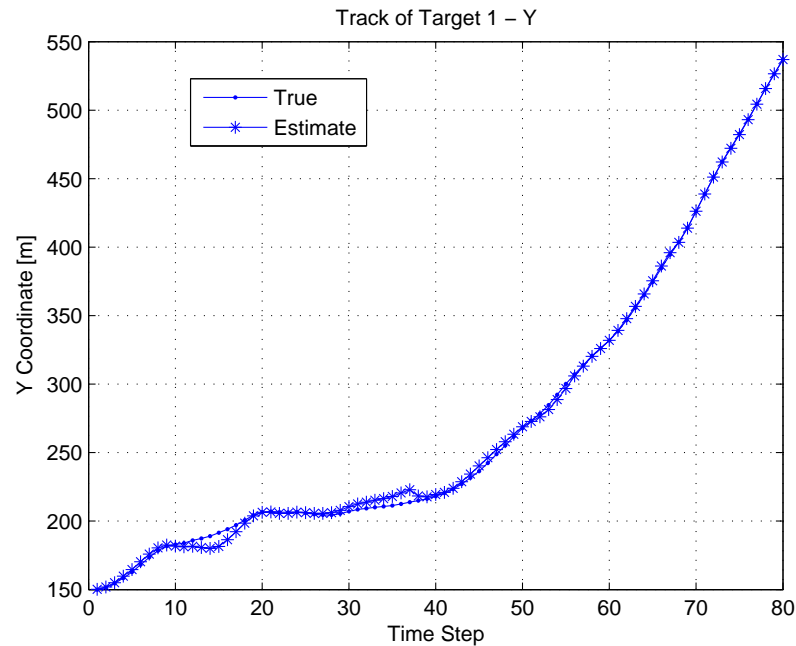


Figure 8.3: Track of target 1 - Y coordinate. The estimate is the mean of 1,000 runs.

8.8 Conclusions

In this chapter, we presented a general tracking algorithm based on particle filtering that uses RSS measurements where the multiple targets are unidentified, the number of targets varies, and the reference powers of the targets are unknown and vary with time. We introduced an initialization step of the tracking algorithm where the LS method is applied. We assumed only a single target at the beginning of the tracking. However, simultaneous initialization of two targets can also be accomplished [52]. In that case, the estimated reference power may not be accurate. In the future, we can improve on this method and study the simultaneous localization of more than two targets.

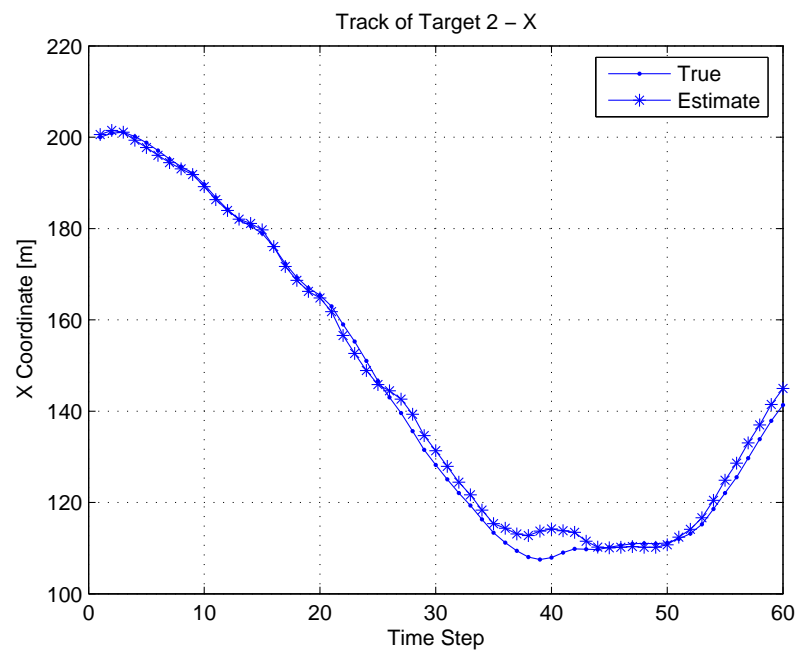


Figure 8.4: Track of target 2 - X coordinate. The estimate is the mean of 1,000 runs.

Table 8.1: LS initialization of a new target for particle filtering

At time t , from all the measurements $y_{s_{1:N}}$, find the three best measurements ($y_{(s_1, s_2, s_3, s_4), t}$) and the identities of the corresponding sensors (s_1, s_2, s_3, s_4). Suppose $K_{t-1}^m = 1$ (the method can easily be generalized for any value of K_{t-1}^m).

- $\eta = 0$,
- For $m = 1 : M$ (M is the number of particles.)
 - * Model 1: $K_t^{M+1+\eta} = 0$ (the particle $M + 1 + \eta$ is generated from particle m). The elements of the other states become empty. Compute the weight, $w_t^{M+1+\eta}$.
 - * Model 2: $K_t^m = 1$ (particle m is generated from particle m). Generate a new particle χ_t^m as follows:

$$\chi_t^m = \{\mathbf{K}_t^m, \mathcal{S}_t^m, \mathbf{x}_t^m\} = \{1, \mathcal{S}_t^m, \mathbf{x}_t^m\}, \{\mathbf{x}_t^m, \mathcal{S}_t^m\} \sim p(\mathbf{x}_t, \mathcal{S}_t | \mathbf{x}_{t-1}^m, \mathcal{S}_{t-1}^m)$$

according to the SIR particle filter. Compute the weight, w_t^m .

- * Model 3: $\mathbf{K}_t^{M+2+\eta} = 2$ (particle $M + 1 + \eta$ is generated from particle m). Suppose

$$\begin{aligned} y_{n,t} &= 10 \log_{10} \left(\sum_{k=1}^{\mathbf{K}_t^{M+2+\eta}} \frac{\Psi_k d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_{k,t}|^\alpha} \right) + v_{s_n,t} \\ &= 10 \log_{10} \left[\left(\sum_{k=1}^{\mathbf{K}_{t-1}^m} \frac{\Psi_k d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_{k,t}^p|^\alpha} \right) + \frac{\Psi_{\text{new}} d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_{\text{new},t}|^\alpha} \right] \\ &= 10 \log_{10} \left[\Gamma_n + \frac{\Psi_{\text{new}} d_0^\alpha}{\Upsilon_{s_n,t}^\alpha} \right] \quad \text{then, } \Upsilon_{s_n,t} = \left(\frac{\Psi_{\text{new}} d_0^\alpha}{10^{(y_{s_n,t}/10)} - \Gamma_n} \right)^{1/\alpha} \end{aligned} \quad (8.2)$$

where $\Gamma_n = \sum_{k=1}^{\mathbf{K}_{t-1}^m} \frac{\Psi_k d_0^\alpha}{|\mathbf{r}_n - \mathbf{l}_{k,t}^p|^\alpha}$ is the predicted part of the measurement by the continuing target, and $\Upsilon_{s_n,t} = |\mathbf{r}_n - \mathbf{l}_{\text{new},t}|$ is the estimated distance between the new target and each sensor. The symbol Ψ_{new} is the reference power of the new target, and $\mathbf{l}_{k,t}^p$ are the predicted locations of the targets propagating from the previous time step.

- Cancel the power from $\Upsilon_{s_{1:4},t}$ by forming (if α is 2)

$$\left| \frac{\Upsilon_{s_i}}{\Upsilon_{s_j}} \right| \triangleq \gamma_{ij} = \sqrt{\frac{10^{(y_j/10)} - \Gamma_j}{10^{(y_i/10)} - \Gamma_i}}, \quad i = 1 \quad \text{and } j = 2, 3, 4. \quad (8.3)$$

- Apply the LS method to estimate \mathbf{l}_{new} , and then, using \mathbf{l}_{new} , estimate the reference power.

- Compute the weight, $w_t^{M+2+\eta}$.

* $\eta = \eta + 2$

end

- Compute the *weight sum* of each model to select the maximum, then find the MMSE estimate of the state.

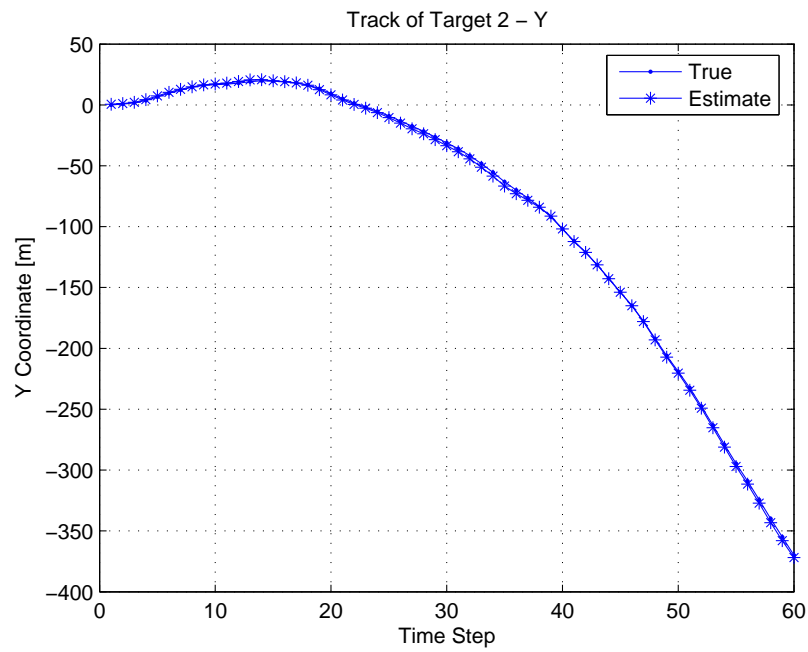


Figure 8.5: Track of target 2 - Y coordinate. The estimate is the mean of 1,000 runs.

Table 8.2: Mean error of the initialized parameters.

	X coordinate	Y coordinate	Reference Power
Target 1	-0.0465	0.2536	-0.0936
Target 2	0.5690	0.5060	0.0327

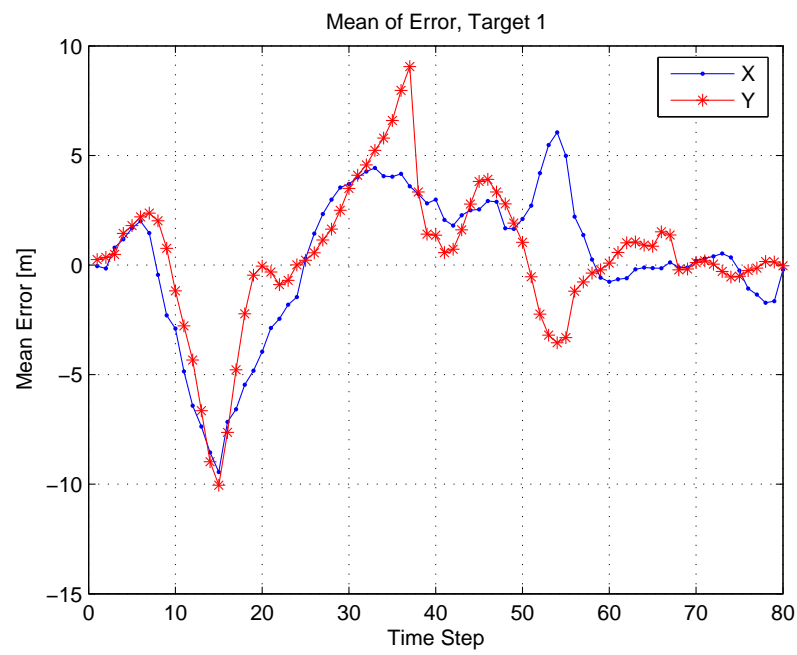


Figure 8.6: Mean of the error for target 1 from 1,000 runs.

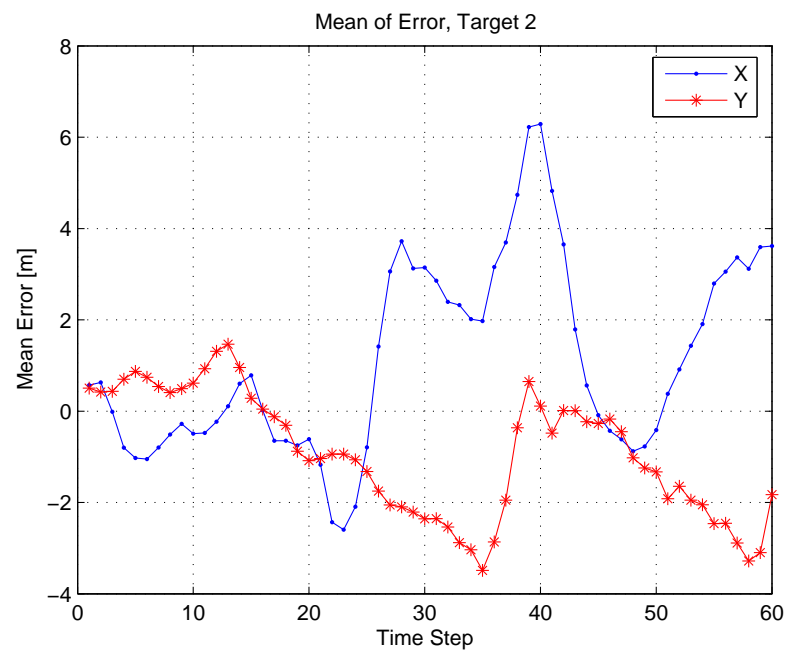


Figure 8.7: Mean of error for target 2 from 1,000 runs.



Figure 8.8: The reference power of target 1. The estimate is the mean of 1,000 runs.

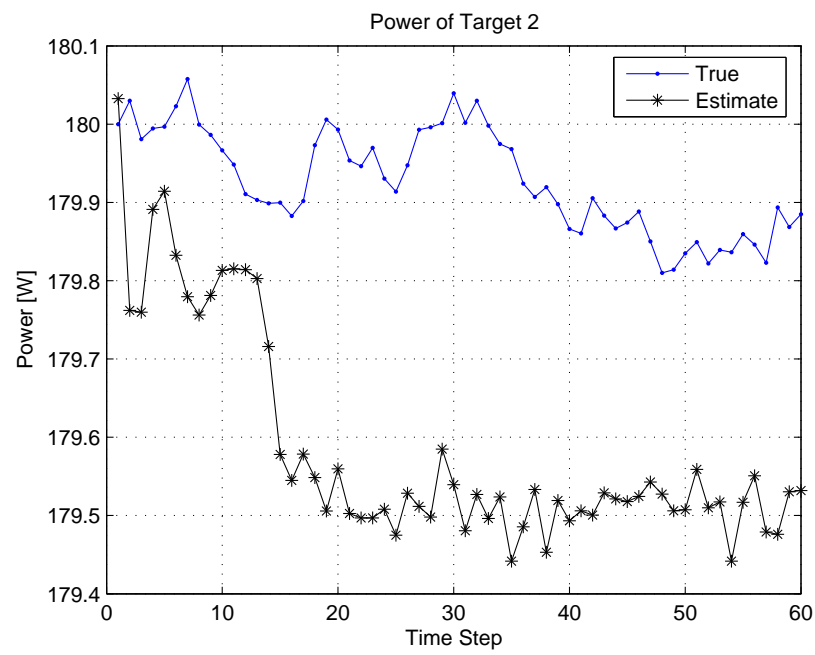


Figure 8.9: The reference power of target 2. The estimate is the mean of 1,000 runs.

Chapter 9

Conclusion

Sensor networks have been attracting increasing research interest, primarily due to the recent advances in micro electronics, array processing, and wireless networking technology. Some of the tasks of these sensor networks include detection, localization, tracking and classification of objects. Sensor networks have many applications in the military and industry.

The received signal strength (RSS) sensors have advantages and disadvantages when compared to sensors that measure TOA, TDOA, and DOA. Generally, RSS measurements represent a superposition of signals whose sources are different and therefore the measurements do not allow one-to-one mapping unless the signal transmitter and receiver are synchronized and able to independently communicate with each other. Thus, the problem of multiple target tracking, especially when the number of targets varies, is quite challenging.

In this dissertation, we proposed a solution to tracking of targets based on RSS measurements. Our solution is valid for cases when at the beginning of the tracking

there is at most one target in the field. The number of targets from then on can vary with time and can increase and/or decrease. The reference powers of the targets are assumed unknown.

In the dissertation we suggest a number of solutions regarding detection, localization, and state estimation of the targets. However, we leave many problems open for further work in the future. One of them is the simultaneous localization/initialization of more than one target. Another problem is the use of compressed RSS measurements (the extreme is binary measurements) for initialization and tracking. It will also be interesting to study the tracking problem when the sensor may provide biased measurements where the biases are also unknown. Tracking of manoeuvring targets in our scenario is also an unresolved problem. Finally, it will be important to test the proposed methods on real data.

Appendix A

The Newton-Raphson method

From (5.1), if we set $\Psi d_0^\alpha = C$, $\alpha = 2$, and $|\mathbf{r}_n - \mathbf{l}|^\alpha = \mathbf{D}_n$, the log-likelihood function becomes

$$\ln p_y(\mathbf{y}; \mathbf{l}) = \ln \left\{ \frac{1}{(2\pi\sigma^2)^{N/2}} \exp \left\{ -\frac{1}{2\sigma^2} \sum \left[y_n - 10 \log_{10} \left(\frac{C}{\mathbf{D}_n} \right) \right]^2 \right\} \right\} \quad (\text{A.1})$$

where $\mathbf{y} = [y_1 \ y_2 \ y_3]^\top$ (these are the three sensor measurements). It can be rewritten as

$$K - \frac{1}{2\sigma^2} \sum \left[y_n - 10 \log_{10} \left(\frac{C}{\mathbf{D}_n} \right) \right]^2 \quad (\text{A.2})$$

where K is a constant which does not depend on the parameter we want to estimate, and we can rewrite it as

$$g(\mathbf{y}) + \frac{1}{\sigma^2} \sum \left\{ y_n 10 \log \left(\frac{C}{\mathbf{D}_n} - \frac{1}{2} \left[10 \log_{10} \left(\frac{C}{\mathbf{D}_n} \right) \right] \right) \right\} \quad (\text{A.3})$$

where $g(\mathbf{y})$ is a function of \mathbf{y} , and the parameter does not depend on it. Therefore,

$$\ln p_y(\mathbf{y}; \mathbf{l}) = g(\mathbf{y}) + \frac{1}{\sigma^2} \sum (A) \quad (\text{A.4})$$

where A is defined as

$$\sum (A) \triangleq \mathbf{d}^\top \mathbf{y} - \frac{1}{2} \mathbf{d}^\top \mathbf{d} \quad (\text{A.5})$$

and

$$\mathbf{d} = \left[10 \log_{10} \left(\frac{C}{D_1} \right) \quad 10 \log_{10} \left(\frac{C}{D_2} \right) \quad 10 \log_{10} \left(\frac{C}{D_3} \right) \right]^\top. \quad (\text{A.6})$$

Therefore,

$$\ln p_y(\mathbf{y}; \mathbf{l}) = g(\mathbf{y}) + \frac{1}{\sigma^2} \sum \left(\mathbf{d}^\top \mathbf{y} - \frac{1}{2} \mathbf{d}^\top \mathbf{d} \right) \quad (\text{A.7})$$

and maximizing the likelihood function is the same as maximizing $\mathbf{d}^\top \mathbf{y} - \frac{1}{2} \mathbf{d}^\top \mathbf{d}$. In order to solve

$$\frac{\partial \ln p(\mathbf{y}; \mathbf{l})}{\partial \mathbf{l}} = 0, \quad \Rightarrow \quad \frac{\partial}{\partial \mathbf{l}} \left(\mathbf{d}^\top \mathbf{y} - \frac{1}{2} \mathbf{d}^\top \mathbf{d} \right) = 0 \quad (\text{A.8})$$

we need to apply an iterative method to find \mathbf{l} that satisfies (A.8) as follows:

$$\mathbf{l}_{k+1} = \mathbf{l}_k - \left[\frac{\partial^2}{\partial \mathbf{l}^2} \left(\mathbf{d}^\top \mathbf{y} - \frac{1}{2} \mathbf{d}^\top \mathbf{d} \right) \right]^{-1} \frac{\partial}{\partial \mathbf{l}} \left(\mathbf{d}^\top \mathbf{y} - \frac{1}{2} \mathbf{d}^\top \mathbf{d} \right) \Big|_{\mathbf{l}=\mathbf{l}_k}. \quad (\text{A.9})$$

Before we proceed, we make the following abbreviations:

$$\frac{1}{|\mathbf{r}_n - \mathbf{l}|^2} = \frac{1}{(r_{nx} - l_x)^2 + (r_{ny} - l_y)^2} = \frac{1}{\mathbf{D}_n} = R_n \quad (\text{A.10})$$

$$X_n = r_{nx} - l_x, \quad Y_n = r_{ny} - l_y \quad (\text{A.11})$$

$$\mathbf{Q} \triangleq \mathbf{d}^\top \mathbf{y} - \frac{1}{2} \mathbf{d}^\top \mathbf{d}. \quad (\text{A.12})$$

Then, $\mathbf{d}^\top \mathbf{y} - \frac{1}{2} \mathbf{d}^\top \mathbf{d}$ can be modified as

$$\mathbf{Q} = \sum y_n 10 \log_{10} \left(\frac{C}{D_n} \right) - \frac{1}{2} \sum \left[10 \log_{10} \left(\frac{C}{D_n} \right) \right]^2 \quad (\text{A.13})$$

$$= \sum y_n 10 \log_{10}(R_n C) - \frac{1}{2} \sum [10 \log_{10}(R_n C)]^2. \quad (\text{A.14})$$

Then, the first and second derivatives of \mathbf{Q} can be computed as

$$\frac{\partial \mathbf{Q}}{\partial l_x} = \sum \left(y_n \frac{\partial R_n}{\partial l_x} \frac{1}{R_n} \right) - \sum \left[10 \log_{10}(R_n C) \frac{\partial R_n}{\partial l_x} \frac{1}{R_n} \right] \quad (\text{A.15})$$

$$= 2 \sum (y_n X_n R_n) - 20 \sum [\log_{10}(R_n C) \cdot X_n R_n]. \quad (\text{A.16})$$

Similarly, we can find the first derivative with respect to the y coordinate,

$$\frac{\partial \mathbf{Q}}{\partial l_y} = 2 \sum (y_n Y_n R_n) - 20 \sum [\log_{10}(R_n C) \cdot Y_n R_n]. \quad (\text{A.17})$$

Next, the second derivative can be computed as

$$\frac{\partial \mathbf{Q}}{\partial l_x^2} = \frac{\partial}{\partial l_x} \left\{ 2 \sum (X_n R_n y_n) - 20 \sum [X_n R_n \cdot \log_{10}(R_n C)] \right\} \quad (\text{A.18})$$

$$= 2 \sum \left[\frac{\partial}{\partial l_x} (X_n R_n y_n) \right] - 20 \sum \left\{ \frac{\partial}{\partial l_x} [X_n R_n \cdot \log_{10}(R_n C)] \right\}. \quad (\text{A.19})$$

The first and second term can be obtained, respectively, as follows:

$$\frac{\partial}{\partial l_x} (X_n R_n y_n) = \frac{\partial X_n}{\partial l_x} (R_n y_n) + X_n \left(y_n \frac{\partial R_n}{\partial l_x} \right) \quad (\text{A.20})$$

$$= -R_n y_n + X_n \cdot 2X_n R_n^2 y_n = R_n y_n (2X_n^2 R_n - 1). \quad (\text{A.21})$$

and the second term is

$$\begin{aligned} \frac{\partial}{\partial l_x} [X_n R_n \cdot \log_{10} (R_n C)] &= \frac{\partial}{\partial l_x} (X_n R_n) \cdot [\log_{10} (R_n C)] + (X_n R_n) \frac{\partial R_n}{\partial l_x} \frac{1}{R_n} \\ &= R_n (2X_n^2 R_n - 1) \cdot [\log_{10} (R_n C)] + 2X_n^2 R_n^2. \end{aligned}$$

Therefore,

$$\frac{\partial Q}{\partial l_x^2} = 2 \sum [R_n y_n (2X_n^2 R_n - 1)] - 20 \sum \{R_n (2X_n^2 R_n - 1) \cdot [\log_{10} (R_n C)] + 2X_n^2 R_n^2\}. \quad (\text{A.22})$$

Similarly,

$$\frac{\partial Q}{\partial l_y^2} = 2 \sum [R_n y_n (2Y_n^2 R_n - 1)] - 20 \sum \{R_n (2Y_n^2 R_n - 1) \cdot [\log_{10} (R_n C)] + 2Y_n^2 R_n^2\} \quad (\text{A.23})$$

where

$$\frac{\partial R_n}{\partial l_x} = \frac{2(r_{nx} - l_x)}{[(r_{nx} - l_x)^2 + (r_{ny} - l_y)^2]^2} = 2X_n R_n^2 \quad (\text{A.24})$$

$$\frac{\partial R_n}{\partial l_y} = \frac{2(r_{ny} - l_y)}{[(r_{nx} - l_x)^2 + (r_{ny} - l_y)^2]^2} = 2Y_n R_n^2. \quad (\text{A.25})$$

We use the initial guess of \mathbf{l}_1 from the solution of the LS method using three measurements (see Section 5.4).

Appendix B

Cramer-Rao Bound (CRB)

We derive the CRB of location of a source target when we use only three measurements from sensors that form a right triangle. The parameter is denoted by $\boldsymbol{\theta} = \mathbf{l} = [x \ y]^\top$. From (5.1), the log-likelihood function is

$$\ln p(\mathbf{y}; \boldsymbol{\theta}) = \ln \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^3 + \left[-\frac{1}{2\sigma^2} \sum_{n=1}^3 (y_n - f_n(\boldsymbol{\theta})) \right] \quad (\text{B.1})$$

where

$$f_n(\boldsymbol{\theta}) = 10 \log_{10} \left[\frac{\Psi}{g_n(\boldsymbol{\theta})} \right] \quad (\text{B.2})$$

and

$$g_n(\boldsymbol{\theta}) = g_n(x, y) = |\mathbf{s}_n - \mathbf{l}|^\alpha = (s_{n_x} - x)^2 + (s_{n_y} - y)^2. \quad (\text{B.3})$$

The derivative with respect to the x coordinate is

$$\frac{\partial \ln p}{\partial x} = \frac{\partial}{\partial x} \left[-\frac{1}{2\sigma^2} \sum_{n=1}^3 (y_n - f_n)^2 \right] = -\frac{1}{2\sigma^2} \sum_{n=1}^3 \underbrace{\left\{ \frac{\partial}{\partial x} [(y_n - f_n)^2] \right\}}_{\mathcal{A}}. \quad (\text{B.4})$$

We have

$$\mathcal{A} = \frac{\partial}{\partial x} [(y_n - f_n)^2] = 2 [y_n - f_n(\boldsymbol{\theta})] \underbrace{\left[-\frac{\partial f_n(x, y)}{\partial x} \right]}_{\mathcal{B}}. \quad (\text{B.5})$$

and

$$-\mathcal{B} = \frac{\partial f_n(x, y)}{\partial x} = \frac{\partial}{\partial x} \left\{ 10 \log_{10} \left[\frac{\Psi}{g_n(x, y)} \right] \right\} \quad (\text{B.6})$$

$$= -10 \frac{\partial}{\partial x} [\log_{10} g_n(x, y)] = \frac{20}{\ln 10} \frac{(s_{n_x} - x)}{g_n}. \quad (\text{B.7})$$

If we plug \mathcal{B} into \mathcal{A} ,

$$\mathcal{A} = 2 [y_n - f_n(\boldsymbol{\theta})] \left[-\frac{20}{\ln 10} \frac{(s_{n_x} - x)}{g_n} \right]. \quad (\text{B.8})$$

Plugging \mathcal{A} into (B.4), we get

$$\frac{\partial \ln p}{\partial x} = \frac{20}{\sigma^2 \ln 10} \sum \left\{ \frac{[y_n - f_n(\boldsymbol{\theta})](s_{n_x} - x)}{g_n(x, y)} \right\}. \quad (\text{B.9})$$

Similarly, we can derive the derivative of the y coordinate

$$\frac{\partial \ln p}{\partial y} = \frac{20}{\sigma^2 \ln 10} \sum \left\{ \frac{[y_n - f_n(\boldsymbol{\theta})](s_{n_y} - y)}{g_n(x, y)} \right\}. \quad (\text{B.10})$$

The second derivative of the x coordinate is

$$\frac{\partial^2 \ln p}{\partial x^2} = \frac{20}{\sigma^2 \ln 10} \sum \frac{\partial}{\partial x} \left[\frac{(y_n - f_n)(s_{n_x} - x)}{g_n(x, y)} \right]. \quad (\text{B.11})$$

If we define

$$P_x(x, y) \triangleq (s_{n_x} - x)(y_n - f_n) \quad \text{and} \quad \mathcal{G} \triangleq \frac{\partial}{\partial x} \left(\frac{P_x}{g_n} \right) = \frac{P_x' g_n - P_x g_n'}{g_n^2} \quad (\text{B.12})$$

where

$$P_x' = -(y_n - f_n) - (s_{n_x} - x)f_n', \quad (\text{B.13})$$

$$f_n' = \frac{20}{\ln 10} \frac{(s_{n_x} - x)}{g_n} \quad \text{from } \mathcal{B}, \quad \text{and} \quad g_n' = -2(s_{n_x} - x). \quad (\text{B.14})$$

$$(\text{B.15})$$

Then,

$$P_x' = -(y_n - f_n) - \frac{20}{\ln 10} \cdot \frac{(s_{n_x} - x)^2}{g_n}. \quad (\text{B.16})$$

Plugging P_x' into \mathcal{G} , we find

$$\mathcal{G} = \frac{2(s_n - x)^2(y_n - f_n) - (y_n - f_n) \cdot g_n - \frac{20}{\ln 10} \cdot (s_{n_x} - x)^2}{g_n^2}. \quad (\text{B.17})$$

Then, (B.11) becomes

$$\frac{\partial^2 \ln p}{\partial x^2} = \frac{20}{\sigma^2 \ln 10} \sum \left[\frac{2(s_n - x)^2(y_n - f_n) - (y_n - f_n) \cdot g_n - \frac{20}{\ln 10} \cdot (s_{n_x} - x)^2}{g_n^2} \right]. \quad (\text{B.18})$$

Similarly, we can derive it with respect to the y coordinate.

$$\frac{\partial^2 \ln p}{\partial y^2} = \frac{20}{\sigma^2 \ln 10} \sum \left[\frac{2(s_n - y)^2(y_n - f_n) - (y_n - f_n) \cdot g_n - \frac{20}{\ln 10} \cdot (s_{n_y} - y)^2}{g_n^2} \right]. \quad (\text{B.19})$$

In finding the elements of the Fisher information matrix, we write

$$\frac{\partial^2 \ln p}{\partial y \partial x} = \frac{\partial}{\partial y} \left\{ \frac{20}{\sigma^2 \ln 10} \sum \left[\frac{(y_n - f_n)(s_{n_x} - x)}{g_n} \right] \right\} \quad (\text{B.20})$$

$$= \frac{20}{\sigma^2 \ln 10} \sum \underbrace{\left[\frac{\partial}{\partial y} \left(\frac{P_x}{g_n} \right) \right]}_{Q_n}. \quad (\text{B.21})$$

$$\frac{\partial Q_n}{\partial y} = \frac{P_x' g_n - P_x g_n'}{g_n^2} = \frac{2(s_{n_x} - x)(s_{n_y} - y)(y_n - f_n)}{g_n^2} - \frac{\frac{20}{\ln 10}(s_{n_x} - x)(s_{n_y} - y)}{g_n^2} \quad (\text{B.22})$$

where $\frac{\partial g_n}{\partial y} = -2(s_{n_y} - y)$, $\frac{\partial f_n}{\partial y} = \frac{20}{\ln 10} \frac{(s_{n_x} - x)(s_{n_y} - y)}{g_n^2}$. Therefore

$$\frac{\partial^2 \ln p}{\partial y \partial x} = \frac{20}{\sigma^2 \ln 10} \sum \left[\frac{2(s_{n_x} - x)(s_{n_y} - y)(y_n - f_n) - \frac{20}{\ln 10}(s_{n_x} - x)(s_{n_y} - y)}{g_n^2} \right]. \quad (\text{B.23})$$

Similarly,

$$\frac{\partial^2 \ln p}{\partial x \partial y} = \frac{20}{\sigma^2 \ln 10} \sum \left[\frac{2(s_{n_x} - x)(s_{n_y} - y)(y_n - f_n) - \frac{20}{\ln 10}(s_{n_x} - x)(s_{n_y} - y)}{g_n^2} \right]. \quad (\text{B.24})$$

The Fisher information matrix is

$$\mathbf{I}(\boldsymbol{\theta}) = \begin{bmatrix} -E \left(\frac{\partial^2 \ln p}{\partial x^2} \right) & -E \left(\frac{\partial^2 \ln p}{\partial x \partial y} \right) \\ -E \left(\frac{\partial^2 \ln p}{\partial y \partial x} \right) & -E \left(\frac{\partial^2 \ln p}{\partial y^2} \right) \end{bmatrix} \quad (\text{B.25})$$

note [expectation of f_n] = y_n , and from (B.18) and (B.23), we have

$$\begin{aligned} E\left(\frac{\partial^2 \ln p}{\partial x \partial y}\right) &= \frac{20}{\sigma^2 \ln 10} \sum \left[\frac{-\frac{20}{\ln 10} \cdot (s_{n_x} - x)(s_{n_y} - y)}{g_n^2} \right] \\ &= -\left(\frac{20}{\sigma \ln 10}\right)^2 \sum \left[\frac{(s_{n_x} - x)(s_{n_y} - y)}{g_n^2} \right]. \end{aligned} \quad (\text{B.26})$$

Similarly,

$$E\left(\frac{\partial^2 \ln p}{\partial y \partial x}\right) = -\left(\frac{20}{\sigma \ln 10}\right)^2 \sum \left[\frac{(s_{n_y} - y)(s_{n_x} - x)}{g_n^2} \right]. \quad (\text{B.27})$$

$$E\left(\frac{\partial^2 \ln p}{\partial x^2}\right) = \frac{20}{\sigma^2 \ln 10} \sum \left[\frac{-\frac{20}{\ln 10} \cdot (s_{n_x} - x)^2}{g_n^2} \right] = -\left(\frac{20}{\sigma \ln 10}\right)^2 \sum \left[\frac{(s_{n_x} - x)^2}{g_n^2} \right]. \quad (\text{B.28})$$

and

$$E\left(\frac{\partial^2 \ln p}{\partial y^2}\right) = \frac{20}{\sigma^2 \ln 10} \sum \left[\frac{-\frac{20}{\ln 10} \cdot (s_{n_y} - y)^2}{g_n^2} \right] = -\left(\frac{20}{\sigma \ln 10}\right)^2 \sum \left[\frac{(s_{n_y} - y)^2}{g_n^2} \right]. \quad (\text{B.29})$$

Therefore,

$$\mathbf{I}(\boldsymbol{\theta}) = \left(\frac{20}{\sigma \ln 10}\right)^2 \cdot \begin{bmatrix} \sum \left[\frac{(s_{n_x} - x)^2}{g_n^2} \right] & \sum \left[\frac{(s_{n_x} - x)(s_{n_y} - y)}{g_n^2} \right] \\ \sum \left[\frac{(s_{n_x} - x)(s_{n_y} - y)}{g_n^2} \right] & \sum \left[\frac{(s_{n_y} - y)^2}{g_n^2} \right] \end{bmatrix}. \quad (\text{B.30})$$

Since

$$\text{var}(\hat{\theta}_i) \geq [\mathbf{I}^{-1}(\boldsymbol{\theta})]_{ii} \quad (\text{B.31})$$

$$\text{var}(\hat{x}) \geq \frac{\left(\frac{\sigma \ln 10}{20}\right)^2 \cdot \sum \left[\frac{(s_{ny}-y)^2}{g_n^2}\right]}{\sum \left[\frac{(s_{nx}-x)^2}{g_n^2}\right] \sum \left[\frac{(s_{ny}-y)^2}{g_n^2}\right] - \left\{ \sum \left[\frac{(s_{nx}-x)(s_{ny}-y)}{g_n^2}\right] \right\}^2}. \quad (\text{B.32})$$

Similarly,

$$\text{var}(\hat{y}) \geq \frac{\left(\frac{\sigma \ln 10}{20}\right)^2 \cdot \sum \left[\frac{(s_{nx}-x)^2}{g_n^2}\right]}{\sum \left[\frac{(s_{nx}-x)^2}{g_n^2}\right] \sum \left[\frac{(s_{ny}-y)^2}{g_n^2}\right] - \left\{ \sum \left[\frac{(s_{nx}-x)(s_{ny}-y)}{g_n^2}\right] \right\}^2}. \quad (\text{B.33})$$

□

Bibliography

- [1] R. Schmidt, “Least squares range difference location,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 32, Jan 1996.
- [2] Y. T. Chan and K. C. Ho, “A simple and efficient estimator for hyperbolic location,” *IEEE Trans. Signal Processing*, vol. 42, Aug 1994.
- [3] O. Jahromi and P. Aarabi, “Time delay estimation and signal reconstruction using multi-rate measurements,” in *2003 int. Conf. Multimedia and Expo*, July 2003, vol. 2, pp. 597–600.
- [4] M.S. Brandstein and H. Silverman, “A robust method for speech signal time-delay estimation in reverberant rooms,” in *Proc. IEEE Conf. Acoustics, Speech and Signal Processing, ICASSP*, May 1996.
- [5] C. H. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-24(4), pp. 320–327, Aug. 1976.
- [6] D. P. O’Leary, “The direction-of-arrival problem: coming at you,” *Computing in Science and Engineering*, vol. 5, pp. 60–63, Nov.-Dec. 2003.

- [7] R. Roy and T. Kailath, *ESPRIT - Estimation of Signal Parameters via Rotational Invariance Techniques*, Springer-Verlag, 1990.
- [8] K.J.R. Liu et al, "URV ESPRIT for tracking time-varying signals," *IEEE Trans. Signal Processing*, vol. 42, pp. 3441–3448, 1994.
- [9] X. Sheng and Y. Hu, "Energy based acoustic source localization," in *The 2nd International Workshop on Information Processing in Sensor Networks*, April 2003, pp. 285–300.
- [10] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Processing, Special Issue on Signal Processing in Networks*, vol. 51, no. 8, pp. 2137–2148, Nov 2003.
- [11] P.J. Harrison and C.F. Stevens, "Bayesian forecasting (with discussion)," *J. the Royal Statistical Society*, vol. 38, pp. 205–247, 1976.
- [12] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/nonGaussian Bayesian state estimation," *IEE Proc-F*, vol. 140, no. 2, pp. 107–113, April 1993.
- [13] E. Punsakaya, C. Andrieu, and A. Doucet, "Particle filtering for optimal detection in fading channels," Tech. Rep. TR.384, CUED(Cambridge University Engineering Department), 2000.
- [14] P. Billingsley, *Probability and Measure*, John Wiley and Sons, Inc., New York, 1979.

- [15] J.E. Handschin and D.Q. Mayne, “Monte Carlo techniques to estimate the conditional expectation in multi-stage nonlinear filtering,” *Int. J. Control*, vol. 9, pp. 547–559, 1969.
- [16] J.E. Handschin, “Monte carlo techniques for prediction and filtering of non-linear stochastic processes,” *Automatica*, vol. 6, pp. 1–23, 1970.
- [17] M. Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Processing*, vol. 50, no. 2, Feb. 2002.
- [18] A. Doucet, S. Godsill, and C. Andrieu, “On Sequential Monte Carlo Sampling Methods for Bayesian Filtering,” *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [19] V. Zaritskii, V. Svetnik, and L. Shimelevich, “Monte Carlo techniques in problems of optimal data processing,” *Automation and Remote*, vol. 12, pp. 95–103, 1975.
- [20] H. Akashi and H. Kumamoto, “Construction of discrete-time nonlinear filter by Monte Carlo methods with variance-reducing techniques,” *Systems and Control*, vol. 19, pp. 211–221, 1975.
- [21] J. Agre and L. Clare, “An integrated architecture for cooperative sensing and networks,” *Computer*, vol. 33, no. 5, May 2000.
- [22] H. Al-Hertani and J. Ilow, “Detection and localization in a wireless network of randomly distributed sensors,” in *IEEE CCECE (Canadian Conference on Electrical and Computer Engineering) 2003*, May 2003, vol. 2, pp. 1239–1242.

- [23] J. Chamberland and V. Veeravalli, “Decentralized detection in sensor networks,” *IEEE Trans. Signal Processing*, vol. 51, pp. 407–416, Feb. 2003.
- [24] J. Liu, X. Koutsoukos, J. Reich, and F. Zhao, “Sensing field : coverage characterization in distributed sensor networks,” in *ICASSP '03*, April 2003, vol. 5, pp. 173–176.
- [25] S. Phoha, N. Jacobson, D. Friedlander, and R. Brooks, “Sensor network based localization and target tracking through hybridization in the operational domains of beamforming and dynamic space time clustering,” in *GLOBECOM '03, IEEE*, Dec. 2003, vol. 5, pp. 2952–2956.
- [26] A. D’Costa and A. Sayeed, “Collaborative signal processing for distributed classification in sensor networks,” in *Proc. IPSN'03*, April 2003, pp. 193–208.
- [27] J.N. Sanders-Reed and et al, “Multi-target tracking in clutter,” in *Proc. SPIE*, April 2002, vol. 4724.
- [28] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” *Automatica*, vol. 8 (2), pp. 451–460, 1975.
- [29] D. Schulz, W. Burgard, D. Fox, and A Cremers, “Tracking multiple moving targets with a mobile robot using particle filters and statistical data association,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2001.
- [30] C. Hue, J-P Le Cadre, and P. Pérez, “Sequential Monte Carlo methods for multiple target tracking and data fusion,” *IEEE Trans. Signal Processing*, vol. 50 (2), pp. 309–325, 2002.

- [31] M. Isard and J. MacCormick, “A Bayesian multiple-blob tracker,” in *IEEE International Conference on Computer Vision, ICCV*, 2001, vol. 2, pp. 34–41.
- [32] I.R. Goodman, R. Mahler, and H. Nguyen, *Mathematics of Data Fusion*, Kluwer Academic, Dordrecht, 1997.
- [33] Ronald P. S. Mahler, ““Statistics 101” for multisensor, multitarget data fusion,” *IEEE A & E Systems Magn*, vol. 19, no. 1, pp. 53–64, January 2004.
- [34] Koen Langendoen and Niels Reijers, “Distributed localization in wireless sensor networks: a quantitative comparison,” *Computer Networks*, vol. 43, pp. 499–518, 2003.
- [35] Anthony J. Weiss, “On the accuracy of a cellular location system based on rss measurements,” *IEEE Transactions on Vehicular Technology*, vol. 52, no. 6, pp. 1508–1518, November 2003.
- [36] J. J. Caffery and G. L. Stuber, “Overview of radiolocation in CDMA cellular systems,” *IEEE Communications Magazine*, vol. 36, no. 4, pp. 38–45, April 1998.
- [37] K. C. Ho and Y. T. Chan, “Solution and performance analysis of geolocation by TDOA,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 4, pp. 1311–1322, October 1993.
- [38] H. Krim and M. Viberg, “Two decades of array signal processing research: The parametric approach,” *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, July 1996.

- [39] M. Sather, J. Snapp, and A. MacDonald, “Mobile-assisted network location system (MNLS) overview,” in *Wireless E911 Phase II Waiver Request Filed by AT & T Wireless Services*. April 2001, FCC docket 94-102.
- [40] Ali H. Sayed, Alireza Tarighat, and Nima Khajehnouri, “Network-based wireless location: challenges faced in developing techniques for accurate wireless location information,” *IEEE Signal Processing Magazine*, vol. 22, pp. 24–40, 2005.
- [41] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, Wiley, New York, 2001.
- [42] Maurizio A. Spirito, “On the accuracy of cellular mobile station location estimation,” *IEEE Transactions on vehicular technology*, vol. 50, no. 3, pp. 674–685, May 2001.
- [43] M. L. Ralston and R. I. Jennrich, “Dud, a derivative free algorithm for non-linear least squares,” *Technometrics*, vol. 20, no. 1, pp. 7–14, Feb. 1978.
- [44] Steven M. Kay, *Fundamentals of statistical signal processing*, vol. 1, Prentice hall signal processing series, 1993, pp. 179.
- [45] P. Deuffhard, *Affine Invariance and Adaptive Algorithms. Springer Series in Computational Mathematics*, vol. 35, Springer, Berlin, 2004.
- [46] F. Gustaffson, F. Gunnarsson, N. Bergman, U. Forssel, J. Jansson, R. Karlsson, and P. J. Nordlund, “Particle filtering for positioning, navigation and tracking,” *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

- [47] P. M. Djurić, M. F. Bugallo, and J. Lim, “Positioning a time-varying number of targets by a wireless sensor,” in *IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2005, pp. 97–100, Puerto Vallarta, México.
- [48] D. Li and Y. H. Hu, “Energy based collaborative source localization using acoustic micor-sensor array,” *J. EUROSIP Applied Signal Process*, vol. 4, pp. 321–337, 2003.
- [49] W. Ng, J. Li, S. Godsill, and J. Vermaak, “A hybrid approach for online joint detection and tracking for multiple targets,” in *Aerospace, 2005 IEEE Conference*, March 2005, pp. 2126–2141.
- [50] J. Lim, “Iterative maximum likelihood locating method based on RSS measurement,” Tech. Rep. 830, CEAS, Stony Brook University, New York, March 2007.
- [51] Arnaud Doucet, Nando de Freitas, and Neil Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.
- [52] Jaechan Lim, “Iterative method initializing two targets simultaneously based on RSS measurement,” Tech. Rep. 831, CEAS, Stony Brook University-SUNY, Stony Brook, New York, March 2007.