ALMA MATER STUDIORUM
UNIVERSITA' DI BOLOGNA
SEDE DI CESENA

Dottorato di Ricerca
in Ingegneria Elettronica, Informatica
e delle Telecomunicazioni

xxii ciclo
ssd: ing-inf/03 - Telecomunicazioni

Cooperative Wireless Networks for Localization

Tesi di:
Ing. Stefano Severi

Relatore:
Chiar.mo Prof. Ing. Marco Chiani

Correlatore:
Chiar.mo Prof. Ing. Davide Dardari

Coordinatore:
Chiar.ma Prof.ssa Ing. Paola Mello

Esame Finale Anno 2010

To Melissa...

Acknowledgements

I would like to acknowledge my Professors Marco Chiani and Davide Dardari for having offered me the study chance and the research environment, for supporting me and for continuously stimulating me with their instructions.

I wish to dedicate a special thank to Professor Giuseppe Abreu, who has hosted me during my 9 months within his research group at the Wireless Communication Centre of the University of Oulu, Finland. He has constantly encouraged, supported and advised me.

I also wish to thank Gianluigi Liva and Giuseppe Destino for their precious suggestions and good works done together.

I can not finally forget all my lab-mates, colleagues and friends at university in Cesena and in Oulu, Andrea Giorgetti, Enrico Paolini, Matteo Mazzotti, Matteo Lucchi, Matteo Fabbri, Michela Varrella, Andrea Mariani, Francesco Guidi, Pedro Nardelli, Carlos Lima, Qiang Xue and Davide Macagnano.

A special thank to my family, Gilberto, Romana and Nadir.

Contents

List of Figures

List of Tables

Acronym

CSDPLA Constrained Semidefinite Programming Localizaton Algorithm

WSN Wireless Sensor Network

MDT Message Delivering Time

CRLB Cramer Rao Lower Bound

FIM Fisher Information Matrix

SIM Simplified Incidence Matrix

IM Incidence Matrix

KVL Kirchhoff's Voltage Law

CSM Cut Set Matrix

SDP Semidefinite Programming

BFS Breadth First Search

CDF Cumulative Density Function

NLOS Non-Line-Of-Sight

LOS Line-Of-Sight

LS Least-Square

DC Distance Contraction

CSDP Constrained SDP

IEEE Institute of Electrical and Electronic Engineers

LAN Local Area Network

WLAN Wireless Local Area Network

MAC Medium Access Control

OS Operative System

i.i.d. independent identically distributed

UDG Unitary Disk Graph

AP Access Point

RSSI Received Signal Strength Indicator

NNSS Nearest Neighbor(s) in Signal Space

NIC Network Interface Card

GPS Global Positioning System

MLA Multihop Localization Algorithm

LP Linear Programming

1

Introduction

Recent progress in microelectronic and wireless communications have enabled the development of low cost, low power, multifunctional sensors, which has allowed the birth of new type of networks named WSNs [2]. Nodes of such networks can be positioned, randomly or following a predetermined scheme, over a given area; can both collect data and, at the same time, act as communication centers.

WSNs can be exploited in an huge variety of applications and scenarios [3, 4]. Some of the application areas are health [5, 6], security [7, 8], logistic and environmental monitoring [9, 10]. Doctors can analyze remotely physiological data about a patient, anti-intrusion system can be set up and even data about potential dangerous natural events, like earthquakes or landslides, can be collected in real time.

At the same time WSNs do not have unlimited energetical and computation power and nodes may fail, i.e. they can go out of order. Other constrains are due to the noisy radio channel used by nodes to communicate and to irregular, sometimes even unknown a-priori, network topologies.

To face these constrains is essential to organize the networks nodes in a cooperative way; traditional algorithms and protocols for ad hoc networks can not be well suited for WSNs [11]. On the contrary, for WSNs, cooperation is necessary to realize more complex functions and to optimize the performance.

This thesis adresses the problem of localization, and analyzes its crucial aspects, within the context of cooperative WSNs. The three main issues discussed in the following are: *network synchronization*, *position estimate* and *tracking*.

1. INTRODUCTION

Time synchronization is a fundamental requirement for every network. Log file accuracy, auditing and monitoring, network fault diagnosis and recovery, file timestamps, directory services, access security and authentication, distributed computing, scheduled operations and real-world time value are some key-areas where time synchronization directly affects network operations. The problem of time synchronization is even more relevant in WSNs, where the absence of a reliable channel, as well as poor node oscillators performance, can introduce serious misalignment between the clocks of the various nodes. In particular for WSNs, in order to guarantee reliable services, it is fundamental to minimize the synchronization error. Sensed data without time information are often useless [1]. Time synchronization is also important for basic mechanisms of the network (e.g., routing, wake-up procedures, Medium Access Control (MAC)) to work efficiently. In this context, a new approach based on the estimation theory is proposed to evaluate the ultimate performance limit in network time synchronization. In particular the lower bound on the variance of the average synchronization error in a fully connected network is derived by taking into account the statistical characterization of the Message Delivering Time (MDT) .

Self-localization capability is a highly desirable characteristicv of WSNs. In environmental monitoring applications such as bush fire surveillance, water quality monitoring and precision agriculture, the measurement data are meaningless without knowing the location from where the data are obtained. Moreover, location estimation may enable a myriad of applications such as inventory management, intrusion detection, road traffic monitoring, health monitoring, reconnaissance and surveillance.

Sensor network localization algorithms estimate the locations of sensors with initially unknown location information by using knowledge of the absolute positions of a few sensors and inter-sensor measurements such as distance and bearing measurements. Sensors with known location information are called anchors and their locations can be obtained by using a Global Positioning System (GPS), or by installing anchors at points with known coordinates. In applications requiring a global coordinate system, these anchors will determine the location of the sensor network in the global coordinate system.

In applications where a local coordinate system suffices (e.g., smart homes), these anchors define the local coordinate system to which all other sensors are referred. Because of constraints on the cost and size of sensors, energy consumption, implementation

environment (e.g., GPS is not accessible in some environments) and the deployment of sensors (e.g., sensor nodes may be randomly scattered in the region), most sensors do not know their locations. These sensors with unknown location information are called target or non-anchor nodes and their coordinates will be estimated by the sensor network localization algorithm [12].

Concerning this issue, i.e. the position estimate problem, two main contributions are given.

The first is a new Semidefinite Programming (SDP) framework to analyze and solve the problem of *flip-ambiguity* that afflicts range-based network localization algorithms with incomplete ranging information. In rigid graph theory the notion of flip-ambiguity refers to the erroneous local geometric realization in some parts of the network due to distance measurement errors [13]. In particular one node position can be *mirrored* respect to the axis formed by its neighbors; this error is a major problem when localizing nodes in WSNs since it can be fastly propagated and consequently it can hugely affect a large part of the network.

The occurrence of flip-ambiguous nodes and errors due to flip ambiguity is studied, then with this information a new SDP formulation of the localization problem is built. Finally a flip-ambiguity-robust network localization algorithm is derived and its performance is studied by Monte-Carlo simulations.

The second contribution in the field of position estimate is about *multihop networks*. A multihop network is a network with a low degree of connectivity, in which couples of given any nodes, in order to communicate, they have to rely on one or more intermediate nodes (*hops*). In this scenario it is impossible to determine the exact distance, required to localize nodes, between the nodes that are aware of their position a-priori (*anchor nodes*) and the others (*target nodes*). This approach is a contrasting alternative to current distributed self-localization algorithms, which are based on the idea of "diffusing" the known location of a few anchors to the entire network via a typically large number of message exchanges amongst neighbors, resulting in high communications costs, low robustness to mobility, and little (location) privacy to end users. It is argued that multihop node-to-anchor distance information is sufficient to allow accurate self-localization in multihop wireless networks. Two new distance-based source localization algorithms, highly robust to distance overestimates, typically present in multihop networks, are presented and studied.

The last point of this thesis discuss a new low-complexity tracking algorithm, inspired by the Fano's sequential decoding algorithm for the position tracking of a user in a WLAN-based indoor localization system. Target tracking is one of the non trivial applications of WSNs which is set up in the areas of field surveillance, habitat monitoring, indoor buildings, and intruder tracking [14]. In this thesis the aim is first to estimate the indoor user's position and then to track his movements inside a building in order to offer location-aware services [15]. The positioning infrastructure consists of a IEEE 802.11 wireless LAN and of a radio map of the indoor environment, which is available to the user. The position estimation is performed comparing the signal strength received by the user's to the signal strength values stored in the radio map. By introducing a tracking algorithm, the accuracy of the system is improved, by reducing the number of position estimates which are less probable or impossible for a given environment topology.

At the end of each chapter the most promising future research directions are discussed.

2

Network Synchronization

In most networks, from the simplest to the most structured and largest, a quite efficient time synchronization among nodes composing the network is a fundamental requirement. Log file accuracy, auditing and monitoring, network fault diagnosis and recovery, file timestamps, directory services, access security and authentication, distributed computing, scheduled operations and real-world time value are some key-areas where time synchronization directly affects network operations.

The problem of time synchronization is even more relevant in wireless networks, where the absence of a reliable channel, as well as poor node oscillators performance, can introduce serious misalignment between the clocks of the various nodes.

In particular for WSNs, in order to guarantee reliable services, is fundamental to minimize the synchronization error. In fact, sensed data without time information are often useless. In additional, time synchronization may help the network work efficiently in terms of MAC, routing as well as wake-up/sleeping strategies [1].

To this purpose a lot of synchronization algorithms are present in the literature tuned on different network architectures. One of the most famous is the network time protocol described in [16], whose purpose is to estimate the clock offset between two different nodes and which is widely used for clock synchronization on Internet. Other approaches use a reference node to synchronize the whole network [17, 18], exploit hierarchical topology [19] or gradient clock synchronization [20].

In general, it is difficult to express the performance of a specific algorithm as well as to compare different algorithms developed for different scenarios. For these reasons,

it is important to understand what are the fundamental limits in time synchronization given a certain network topology and communication model.

The work [21] presents upper and lower bounds for clock synchronization in the case of a fully connected network composed of N nodes. It is proved that, even if the clocks all run at the same rate (non-drifting clocks), an uncertainty (jitter) of $U = D_{max} - D_{min}$ in the Message Delivering Time (MDT) makes it impossible to synchronize the clocks of N nodes any more closely than $U(1 - \frac{1}{N})$, where the D_{max} and D_{min} are respectively, the maximum and the minimum MDT among two nodes. Other bounds are presented in [22], [23] and [24]. However these bounds are based on worst case analysis, i.e., the MDT is considered as a fixed parameter, and do not account for the actual statistical behavior of the uncertainty in MDT.

In the following, performance limits of time synchronization are investigated as an estimation theory problem, where the statistical characterization of the uncertainty is properly taken into account. The lower bound on the variance of the mean synchronization error of the network is derived. It can be used as a benchmark in comparing the performance of different algorithms.

First the case of a 2-nodes network is examined then the analysis is extended to a connected network composed of N nodes. After having introduced the general concept of lower bound on the time synchronization error variance, the theoretical bound for a fully-connected network is finally derived.

2.1 2-Nodes Network

The purpose of any time synchronization scheme is to align the nodes logical clocks, that is, to minimize the clock skew among nodes. This is generally achieved through message exchange between nodes.

The first step in this analysis is considering the simplest network, composed of only 2 nodes, A and B. The local time of nodes A and B are defined t_A and t_B respectively. Furthermore the drift between the two oscillators it is assumed to be negligible during the observation window. Under these conditions the *skew* S is defined as:

$$S = t_A - t_B. \tag{2.1}$$

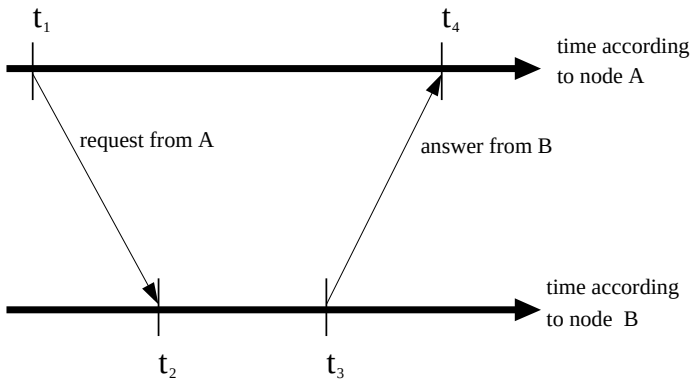


Figure 2.1: Basic time synchronization procedure: two-way message exchange between a pair of nodes [1].

Another useful definition is the MDT $D^{(A,B)}$, i.e., the time that occurs for delivering a message from node A to node B. In general, whatever network type is considered (wired or wireless), the MDT has to be characterized as a random variable to account for the transmission randomness due to several factors such as propagation, Medium Access Control (MAC), Operative System (OS), etc. The variance of $D^{(A,B)}$ is denoted with σ_D^2.

The skew among nodes A and B can be estimated through a classical two-way message exchange (see Fig. 2.1, where A and B exchange their local timestamp [1]). After the message exchange, both nodes can calculate an estimate of the skew S given by

$$\hat{S} = S + \frac{U}{2}, \tag{2.2}$$

where the variable U is the *uncertainty* defined as

$$U \triangleq D^{(A,B)} - D^{(B,A)}. \tag{2.3}$$

In general, several trials are performed because the randomness of MDT, by taking the least value or the average value as estimate. It is furthermore assumed that $D^{(A,B)}$ and $D^{(B,A)}$ are independent identically distributed (i.i.d.) random variables for all pairs of nodes. It follows:

$$\mathbb{E}[U] = 0 \tag{2.4}$$

and

$$\sigma_U^2 = Var(U) = 2\sigma_D^2. \tag{2.5}$$

In order to simplify the notation, the estimation error (*noise*) is defined as

$$X \triangleq \frac{U}{2}, \tag{2.6}$$

and hence

$$\hat{S} = S + X, \tag{2.7}$$

where

$$\mathbb{E}[X] = 0 \tag{2.8}$$

and

$$\sigma_X^2 = \frac{\sigma_U^2}{4} = \frac{\sigma_D^2}{2}. \tag{2.9}$$

2.2 N-Nodes Network

Suppose to have a connected network with N nodes. The network structure can be represented for convenience with an oriented graph $G(\mathbf{N}, \mathbf{A})$, where $\mathbf{N}$ is the set of nodes and $\mathbf{A}$ is the set of arches each one representing the skew between a couple of nodes (see for example Fig. 2.2). The graph is oriented because skews have their own sign; moreover given any closed path inside the network, the sum of all the skews must be zero.

The cardinality of both sets is, respectively

$$|\mathbf{N}| = N \tag{2.10}$$

and

$$|\mathbf{A}| = L. \tag{2.11}$$

The local time of node n is denoted with t_n, with $n = 1, \ldots, N$, and with S_i the skew relative to the ith arch of $G(\mathbf{N}, \mathbf{A})$, with $i = 1, \ldots, L$. Supposing the ith arch involving node n and $n+1$, then the corresponding skew is given by

$$S_i = t_n - t_{n+1}. \tag{2.12}$$

By hypothesis every pair of nodes makes a two way message exchange procedure, thus obtaining the following measurements:

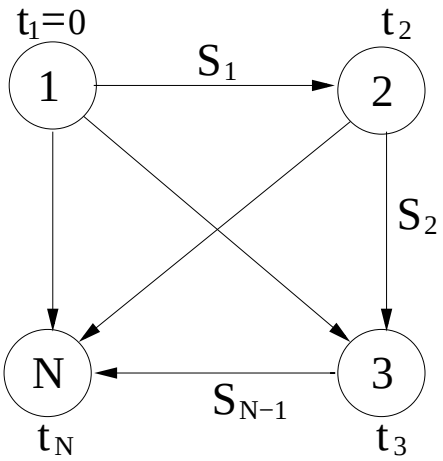


Figure 2.2: N-nodes completely connected network with oriented skews.

$$Y_i = S_i + X_i \qquad i = 1, \ldots, L \tag{2.13}$$

where X_i is the corresponding observation error. The measurements vector[1] is defined as

$$\underline{Y} \triangleq [Y_1, Y_2, \ldots, Y_L]^T, \tag{2.14}$$

and the skews vector

$$\underline{S} \triangleq [S_1, S_2, \ldots, S_L]^T. \tag{2.15}$$

Based on the measurements vector $\underline{Y}$, a time synchronization algorithm attempts to derive the estimation $\hat{S}_j$ of S_j.

The *estimate error* ε_j is defined as the difference between the skew estimate $\hat{S}_j$ and the actual skew S_j, i.e.,

$$\varepsilon_j \triangleq \hat{S}_j - S_j. \tag{2.16}$$

As a consequence, ε_i is in general different from X_i. As it will be clearer later, it is important to note that only the firsts $N-1$ measurements are independent, whereas

[1] operator T denotes the transpose.

the last $L-(N-1)$ measurements are linear combination of the first $N-1$. The average estimate error of the network is

$$\varepsilon \triangleq \frac{1}{L}\sum_{i=1}^{L}\varepsilon_i. \tag{2.17}$$

Now the purpose is to provide a lower bound on the variance of the average estimation error in (2.17).

To this aim we introduce the Incidence Matrix (IM) $\underline{\underline{M}}$ of the graph $G(\mathbf{N},\mathbf{A})$ that was first defined by the physicist Kirchhoff [25]. The IM of a graph has a row for each node and a column for each arch, and each element $M_{v,e} \in \{-1,1,0\}$ [26]. In particular,

- $M_{v,e} = -1$ if arch e is incident upon node v
- $M_{v,e} = 1$ if arch e starts from node v
- 0 in other cases.

Kirchhoff in [25] demonstrated that the rank of the IM is

$$\text{rank}(\underline{\underline{M}}) = N-1. \tag{2.18}$$

In [27] it is shown that it is possibile to choose a reference node in $G(\mathbf{N},\mathbf{A})$ and eliminate the corresponding row in the IM, obtaining a Simplified Incidence Matrix (SIM) $\underline{\underline{Q}}$. Introducing the *time vector* $\underline{T}$,

$$\underline{T} \triangleq [t_2,\ldots,t_N]^T, \tag{2.19}$$

i.e., the vector with all the local times of nodes except for the reference node, the following relationship exists:

$$\underline{\underline{Q}}^T \cdot \underline{T} = \underline{S}. \tag{2.20}$$

Therefore, according to (2.20), the measurements can be represented as follows:

$$\begin{cases} Y_1 = S_1 + X_1 \\ Y_2 = S_2 + X_2 \\ \dots \\ Y_{N-1} = S_{N-1} + X_{N-1} \\ \dots \\ Y_j = (\sum_{i=1}^{N-1} w_{j,i} S_i) + X_j \qquad \text{for} \quad j = N, \dots, L \end{cases} \tag{2.21}$$

where $w_{j,i}$ accounts for the linear dependence of S_j (with $j \geq N$) on $\{S_i\}_{i=1}^{N-1}$. From (2.21) it is evident that it is sufficient, for a complete characterization of the problem, to obtain the estimate

$$\underline{\widehat{S}}_N \triangleq [\widehat{S}_1, \widehat{S}_2, \dots, \widehat{S}_{N-1}] \tag{2.22}$$

of the first $N-1$ skews

$$\underline{S}_N \triangleq [S_1, S_2, \dots, S_{N-1}]. \tag{2.23}$$

The last $L-(N-1)$ estimations can be obtained as linear combination of the firsts $N-1$.

It is shown in [27] that the SIM can always be put in the following canonical form:

$$\underline{\underline{Q}} = [\underline{\underline{I}} | \underline{\underline{W}}], \tag{2.24}$$

where $\underline{\underline{\mathrm{I}}}$ is an $[(N-1)\mathrm{x}(N-1)]$ identity matrix and $\underline{\underline{\mathrm{W}}}$ is a $[(N-1)\mathrm{x}(L\text{-}(N-1))]$ weight matrix.

Thus the $\underline{\underline{Q}}$ is:

$$\underline{\underline{Q}} = \left(\begin{array}{cccc|ccc} 1 & 0 & \dots & 0 & w_{N,1} & \dots & w_{L,1} \\ 0 & 1 & \dots & 0 & w_{N,2} & \dots & w_{L,2} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & w_{N,N-1} & \dots & w_{L,N-1} \end{array} \right). \tag{2.25}$$

The average estimation error (2.17) can therefore be expressed as

$$\varepsilon = \frac{1}{L} \left\{ \sum_{i=1}^{N-1} \varepsilon_i + \sum_{j=N}^{L} \sum_{i=1}^{N-1} w_{ij} \varepsilon_i \right\}. \tag{2.26}$$

2.3 Theoretical Performance Limit on the Time Synchronization Error Variance

A lower bound on the covariance of any estimator is given by the Cramer Rao Lower Bound (CRLB)[1] [28]

$$\mathbb{E}_{\underline{Y}}\left[(\underline{\widehat{S}}_N - \underline{S}_N)^T(\underline{\widehat{S}}_N - \underline{S}_N)\right] \geq \underline{\underline{J}}^{-1}, \tag{2.27}$$

where $\underline{\underline{J}}$ is the Fisher Information Matrix (FIM) which ijth element is given by

$$J_{i,j} = \mathbb{E}_{\underline{Y}}\left[\frac{\delta \ln p_{\underline{Y}}(\underline{y})}{\delta s_i}\frac{\delta \ln p_{\underline{Y}}(\underline{y})}{\delta s_j}\right] \quad \text{for} \quad i,j = 1,\ldots,N-1. \tag{2.28}$$

Note that $\underline{\underline{J}}$ is a $[(N-1)\text{x}(N-1)]$ matrix since it is sufficient to estimate only the first $N-1$ independent skews as previously remarked.

Considering X_i being independent Gaussian random variable, it follows that also the Y_i is Gaussian. Under these hypothesis it is possible to define the correspondent probability density function of $\underline{Y}$:

$$p_{\underline{Y}}(y) = \frac{1}{(\sqrt{2\pi\sigma_X^2})^L}\left(\prod_{i=1}^{N-1} e^{-\frac{(Y_i-S_i)^2}{2\sigma_X^2}} \prod_{i=N}^{L} e^{-\frac{\left(Y_i-\sum_{j=1}^{N-1} w_{i,j}S_j\right)^2}{2\sigma_X^2}}\right). \tag{2.29}$$

The (2.29) can be expressed in the log-scale:

$$\ln p_{\underline{Y}}(y) = k + \frac{1}{2\sigma_X^2}\left[-\sum_{i=1}^{N-1}(Y_i - S_i)^2 - \sum_{i=N}^{L}(Y_i - \sum_{j=1}^{N-1} w_{i,j}S_j)^2\right], \tag{2.30}$$

where the constant k can be neglected for what follows.

By comparing (2.21), (2.25), and (2.30) it is easy to show that the elements of $\underline{\underline{J}}$ are given by the product between the $\underline{\underline{Q}}$ and its transposed $\underline{\underline{Q}}^T$, i.e.,

$$\underline{\underline{J}} = \frac{1}{\sigma_X^2}\underline{\underline{Q}} \cdot \underline{\underline{Q}}^T. \tag{2.31}$$

The resulting FIM is

[1] The notation $\underline{\underline{V}} \geq \underline{\underline{W}}$ means that the matrix $\underline{\underline{V}} - \underline{\underline{W}}$ is positive semi-definite.

$$\underline{\underline{J}} = \frac{1}{\sigma_X^2} \begin{pmatrix} 1 + \sum_{i=N}^{L} w_{i,1} \dots & \sum_{i=N}^{L} w_{i,1} w_{i,N-1} \\ \dots\dots\dots & \\ \sum_{i=N}^{L} w_{i,N-1} w_{i,1} & \dots 1 + \sum_{i=N}^{L} w_{i,N-1} \end{pmatrix}. \tag{2.32}$$

2.4 Theoretical Performance Limit on a Fully Connected Network

In this section the case of a fully connected network with N nodes is examined. Under this further hypothesis, it is possible to give a description of $\underline{\underline{Q}}$.

For a fully connected network, with $L = |A| \triangleq \frac{N(N-1)}{2}$, the sum of the weights of the jth row is

$$\overline{m}_j = \sum_{i=1}^{L} w_{i,j} = 1 + \overline{w}_j = j\,(N-j), \tag{2.33}$$

where

$$\overline{w}_j = \sum_{i=N}^{L} w_{i,j} = j\,(N-j)\; - 1 \tag{2.34}$$

includes only the weights of the right-side of each row.

Using (2.33) and (2.34), for the diagonal elements it is easy to see that

$$J_{jj} = \frac{1}{\sigma_X^2} \left(1 + \sum_{n=N}^{L} w_{n,j} \right) = \frac{1}{\sigma_X^2}\, \overline{m}_j = j(N-j), \tag{2.35}$$

whereas for the off-diagonal elements the expression is

$$J_{ij} = \frac{1}{\sigma_X^2} \left(\sum_{n=N}^{L} w_{n,i} w_{n,j} \right) = j(N-i) \qquad \text{for} \quad i \neq j. \tag{2.36}$$

Finally, the FIM for a completely connected network is

$$\underline{\underline{J}} = \frac{1}{\sigma_X^2} \begin{pmatrix} (N-1) & (N-2) & \dots & 2 & 1 \\ (N-2) & 2(N-2) & \dots & 4 & 2 \\ (N-3) & 2(N-3) & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 3 & 6 & \dots & (N-3)2 & (N-3) \\ 2 & 4 & \dots & (N-2)2 & (N-2) \\ 1 & 2 & \dots & (N-2) & (N-1) \end{pmatrix}. \tag{2.37}$$

which results to be a $[(N-1)\text{ x }(N-1)]$ symmetric matrix.

The reciprocal of the FIM is

$$\underline{\underline{J}}^{-1} = \frac{\sigma_X^2}{N}\begin{pmatrix} 2 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 2 \end{pmatrix}. \tag{2.38}$$

The fact that $\underline{\underline{J}}$ is not completely diagonal means that the skews estimates are not all uncorrelated each other.

Using (2.33) and (2.34), the average estimation error (2.26) for a fully connected network can be further simplified to

$$\varepsilon = \frac{1}{L}\sum_{i=1}^{N-1}(N-i)\,i\,\varepsilon_i. \tag{2.39}$$

The variance of the estimate error is given by

$$\mathrm{Var}(\varepsilon) = \mathbb{E}[\varepsilon^2] = \frac{1}{L^2}\sum_{i=1}^{N-1}(N-i)^2\,i^2\,\mathbb{E}[\varepsilon_i^2] + 2\frac{1}{L}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N-1}(N-i)\,i\,(N-j\,)j\,\mathbb{E}[\varepsilon_i\varepsilon_j]. \tag{2.40}$$

Equation (2.40) can be bounded by the CRLB as follows

$$\mathrm{Var}(\varepsilon) \geq \frac{1}{L^2}\sum_{i=1}^{N-1}(N-i)^2\,i^2\,J_{ii}^{-1} + 2\frac{1}{L}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N-1}(N-i)\,i\,(N-j)\,j\,J_{ij}^{-1}, \tag{2.41}$$

where J_{ij}^{-1} is the ijth element of the $\underline{\underline{J}}^{-1}$. According to (2.38) it is possible to reduce the second term deleting all the null terms of the sum in (2.41). Since

$$J_{ii}^{-1} = \frac{2}{N}\sigma_X^2 \tag{2.42}$$

and

$$J_{ii+1}^{-1} = -\frac{1}{N}\sigma_X^2 \tag{2.43}$$

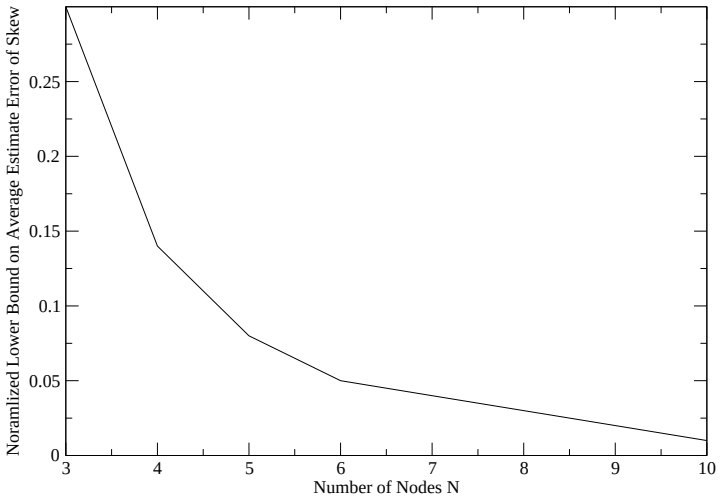


Figure 2.3: Lower bound on the normalized ($\sigma_X^2 = 1$) average estimation error variance as a function of the number of nodes.

the resulting bound is

$$\mathrm{Var}(\varepsilon) \geq \frac{1}{L^2}\frac{2}{N}\sigma_X^2 \left\{ (N-1)^2 + \sum_{i=1}^{N-2} \left[(N-i)^2 i^2 - (N-i)i(N-i-1)(i+1) \right] \right\}.$$

After some algebraical steps (2.44) can be further reduced to

$$\mathrm{Var}(\varepsilon) \geq \sigma_X^2 \frac{4(1+N)}{3(N-1)N^2}, \tag{2.44}$$

which gives the lower bound on the average estimate error of skews in the synchronization process.

For large number of nodes (i.e., $N \gg 1$), (2.44) can be approximated by

$$\mathrm{Var}(\varepsilon) \geq \sigma_X^2 \frac{4}{3N^2}. \tag{2.45}$$

In Fig. 2.3 the lower bound (2.44), normalized to σ_X^2, is plotted as function of the number of nodes. Differently to what predicted by bounds based on worst cases (as in [22, 23, 24]), it can be noted that in a fully connected network the time synchronization accuracy increases with the number of nodes. This is due to the positive effect of the $L-(N-1)$ dependent observations. They introduces a negative covariance ($-\frac{1}{N}\sigma_X^2$)

that reduces the mean synchronization error as N increases. It is important to note that the network synchronization error variance is lower than the error variance (σ_X^2) of the single couple of nodes. This means that maximizing the network connectivity is beneficial for time synchronization.

2.4.1 Future Research Directions

The very next step in this field is the developing of similar limits for different network topologies. In particular straightforward is the extension of the analysis to the star-topology and the tree-topology. The developing of new network synchronization algorithms, with performances close to the CRLB, is the final goal of this work.

3

Position Estimate in Wireless Networks

Localization algorithms for WSN have been widely proposed and studied, expecially in the last years, given the huge variety of localization-aware applications [29]. The goal of position estimate is to determine the physical coordinates of a single or a group of sensor nodes. These coordinates can be global, meaning they are aligned with some externally meaningful system like GPS, or relative, meaning that they are an arbitrary "*rigid transformation*" (rotation, reflection, translation) away from the global coordinate system. *Anchor nodes* (also frequently called *beacon* nodes) are a necessary prerequisite to localize nodes (*target*) within a network in a global coordinate system. Anchor nodes are simply ordinary sensor nodes that know their global coordinates a priori. This knowledge could be set by network manager or acquired through some additional hardware like a GPS receiver. At a minimum, three non-collinear anchor nodes are required to define a global coordinate system in two dimensions. If three dimensional coordinates are required, then at least four non-coplanar beacons must be present [30].

Following the classification introduced by Dardari et al. [1] it is possible to classify the position estimation techniques in the following four categories:

- *Anchor-based*: dome nodes know their location a-priori. In a single-hop or multihop fashion all the others nodes obtain a distance estimate to anchors.

- *Anchor Free*: there are no node with the a-priori knowledge of their position. Only relative/virtual coordinates can be obtained.
- *Range-based*: distance/angle information are obtained through measurements.
- *Range-free*: connectivity is the only information used.

3.1 The Flip-Ambiguity Problem

Flip-ambiguity is one of the most significant potential problems faced by network localization algorithms. This problem arises when the connectivity available in the network is insufficient to ensure that no node to be localized admits multiple location estimates that are equally likely, as per the cost-function that defines the localization algorithm itself.

Consider the example in Fig. 3.1: a typical realization of a WSN suffers, due to the lack of connectivity, a flip-ambiguity. The firsts two target nodes are able to estimate their own position without ambiguity. Target at $(0.5, 0.5)$ is connected to 3 anchor nodes and its position can be estimated with a classic trilateration. The second target at $(0.35, 0.65)$ is connected to 2 anchor nodes and to a target whose position is already determined, so also its location can be estimated. The last target at $(0.8, 0.2)$ is connected to one anchor node and an "already-localized" target node; the connectivity is not sufficient to avoid flip-ambiguity. Consider the cost function typical of any steepest-descend localization algorithm: the problem is now clarified by the existence of two local minima, one in the real position of the target and the second in the mirrored position.

The typical effect of such multiple and equivalent solutions is that optimization methods employed to find the location estimates are unable to distinguish amongst possible different topological configurations that yield equal cost-values, often resulting in catastrophic errors [31].

Due to its nature, there is little hope to find a solution at a reasonable level of complexity to such a problem through the development of new optimization methods. Arguably, this explains the predominant approach found in current related literature which is to identify and eliminate flip-ambiguous nodes from the set of targets to be localized [32, 33].

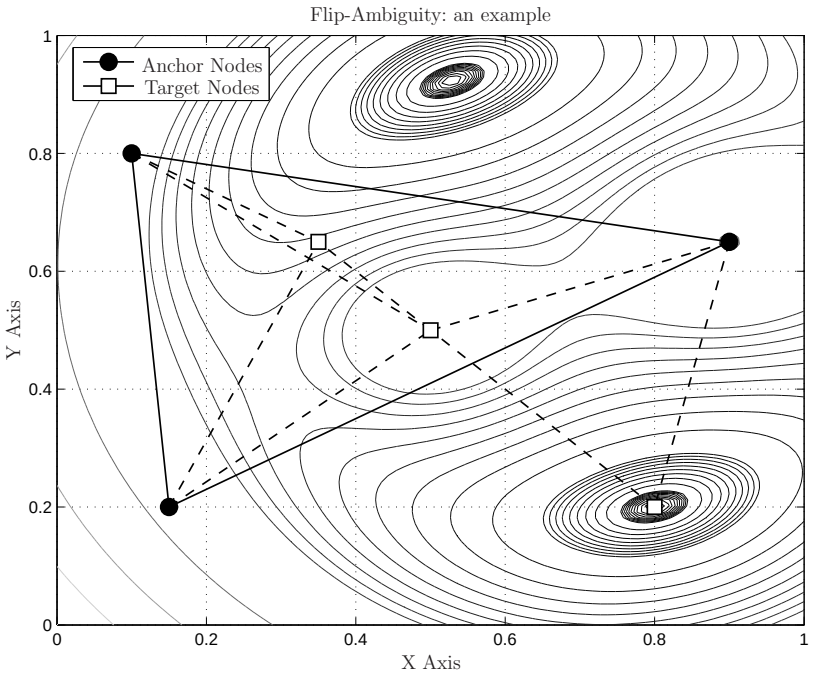


Figure 3.1: A typical occurrence of flip-ambiguity problem within a WSN and the cost function of a classic steepest-descent algorithm.

In other words, the currently common approach amounts to avoid the flip-ambiguity problem and attempt to minimize the location error of the less problematic nodes at the expense of the ones that are harder to localize. While such an approach does improve the performance of localization algorithms – as measured by the errors over the nodes effectively localized – the underlying principle of simply leaving flip-ambiguous nodes without any location estimates may be unacceptable in many applications.

This chapter illustrates a new approach whose aim is to *solve*, rather than avoid, the flip-ambiguity problem by developing a technique that produces location estimates for *all* nodes in the network while minimizing the harmful effect of flip-ambiguity.

To this end, the related analysis starts employing the unique localizability test constructed out of a Semidefinite Programming (SDP) formulation of the network localization problem [34] to study the statistical occurrence of flip-ambiguous nodes. The study reveals that in the majority of situations the number of flip-ambiguous nodes is much smaller than the total number of nodes to be localizable, suggesting that sufficient information remains in the connectivity of non-uniquely localizable networks to

enable flip-ambiguous nodes to be localized as well.

The second step is to study the relationship between unique localizability and constraints that can be imposed over unmeasured distances so as to enable a stable solution to be obtained from the SDP-based localization algorithm. This leads to an impractical (Genie-aided) SDP formulation of the network localization problem which is found indeed to be capable of localizing, with a small average error, all nodes out of network afflicted by flip-ambiguity.

Finally, a concrete solution is obtained by relaxing the unfeasible conditions of the Genie-aided approach and combination with a refining post-processing [35]. An exhaustive Monte Carlo analysis of the performance obtained with the proposed method confirms its ability to resolve flip-ambiguity, at the expense of a negligible increase in localization error.

3.1.1 Localizability Test and Edge-bounding

The localization problem is here studied and discussed for networks lying in the two-dimensional space ($\eta = 2$), which can be studied using the unique-localizability test provided in [34].

In the following, a *network* is understood as a set of N interconnected nodes randomly deployed with uniform distribution inside a square of unitary[1] length ($\ell = 1$).

Nodes whose locations are know exactly and *a priori* will be referred to as *anchors*, while the remaining nodes will be called *targets*.

The total number of nodes N of a network is referred also as its *size*, and all networks studied will contain $n_{\mathrm{A}} \geq 3$ anchor nodes, such that for any realization there are $T = N - n_{\mathrm{A}}$ target nodes.

The scaled Unitary Disk Graph (UDG) model is adopted, so that a pair of target nodes at locations (column vectors) $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, is said to be connected if and only if (iff) their distance

$$d_{ij} \triangleq \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \sqrt{\langle \boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_i - \boldsymbol{x}_j \rangle} \leq R \qquad i, j = 1, \ldots, T, \tag{3.1}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\langle \cdot \rangle$ denotes inner product, and R is a given connectivity range. Likewise, an anchor at the location $\boldsymbol{a}_k$ is connected to a target

[1] This is without loss of generality and amounts to a normalization of all distance quantities.

located at $\boldsymbol{x}_j$ iff $d_{kj} \triangleq \|\boldsymbol{a}_k - \boldsymbol{x}_j\| \leq R$, with $j = 1, \ldots, T$ and $k = 1, \ldots, n_{\mathrm{A}}$. All known distances d_{ij} and d_{kj} are error-free.

It will be assumed that all $d_{ij} \leq R$ and $d_{kj} \leq R$ are known choosing $R = \sqrt{2}$, i.e. R is equal to the diagonal of squared area, of unitary length, of nodes deployment. Since the location of all anchors and their mutual distances are known, anchors are, to all effect considered to be interconnected.

For each network as described above there is an associated graph $G([\mathbf{A}; \mathbf{X}], \mathbf{H})$, where $\mathbf{A} \triangleq [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_{n_{\mathrm{A}}}]$ is the anchor node coordinate matrix carrying the column vectors with the location of all anchors, $\mathbf{X} \triangleq [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T]$ is the target node coordinate matrix carrying the column vectors with the location of all targets, and $\mathbf{H}$ is the IMs defined below.

$$\mathbf{H}_{[N \times \frac{N \cdot (N-1)}{2}]} \triangleq \left[\begin{array}{c|c|c} \mathbf{H}_{\mathrm{A}} & \mathbf{H}_{\mathrm{T}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{E}_{\mathrm{A}} & \mathbf{E}_{\mathrm{T}} \end{array}\right] \tag{3.2}$$

In the above, $\mathbf{H}_{\mathrm{A}}$ and $\mathbf{E}_{\mathrm{T}}$ are the n_{A}-by-$\frac{n_{\mathrm{A}} \cdot (n_{\mathrm{A}}-1)}{2}$ and T-by-$\frac{T \cdot (T-1)}{2}$ Incidence Matrix (IM) of the sub-graphs containing only anchor nodes, and only target nodes, respectively.

In turn, $\mathbf{H}_{\mathrm{T}}$ is a block-diagonal matrix with the structure:

$$\mathbf{H}_{\mathrm{T}} \triangleq \left[\begin{array}{ccc} \boldsymbol{b}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \boldsymbol{b}_{n_{\mathrm{A}}} \end{array}\right], \tag{3.3}$$

where each 1-by-T block $\boldsymbol{b}_k$ is a row vector whose j-th element $\boldsymbol{b}_{kj}$ is 1 iff the k-th anchor is connected to the j-th target.

Finally, the matrix $\mathbf{E}_{\mathrm{A}}$ is formed by appended diagonal matrices such that

$$\mathbf{E}_{\mathrm{A}} \triangleq \left[\mathbf{D}_1 \,\middle|\, \cdots \,\middle|\, \mathbf{D}_{n_{\mathrm{A}}}\right], \tag{3.4}$$

where the j-th element of each T-by-T diagonal matrix $\mathbf{D}_k$ is -1 iff the k-th anchor is connected to the j-th target.

The target-to-target and anchor-to-target squared-norms (or squared-distances) can be concisely written as

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \boldsymbol{e}_{ij}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} \boldsymbol{e}_{ij} = d_{ij}^2, \tag{3.5}$$

$$\|\boldsymbol{a}_k - \boldsymbol{x}_j\|^2 = \left[\boldsymbol{a}_k^{\mathrm{T}} \,\middle|\, \boldsymbol{e}_j^{\mathrm{T}}\right] \left[\begin{array}{c} \mathbf{I}_\eta \\ \hline \mathbf{X}^{\mathrm{T}} \end{array}\right] \left[\mathbf{I}_\eta \,\middle|\, \mathbf{X}\right] \left[\begin{array}{c} \boldsymbol{a}_k \\ \hline \boldsymbol{e}_j \end{array}\right] = d_{kj}^2, \tag{3.6}$$

$$\forall k \leq n_{\mathrm{A}}; i, j \leq T,$$

where $\boldsymbol{e}_{ij}$ is the column vector of $\mathbf{E}_{\mathrm{T}}$ whose i-th and j-th elements are 1 and -1, respectively, while $\boldsymbol{e}_j$ is the column vector of $\mathbf{E}_{\mathrm{A}}$ whose j-th element is -1.

Define the matrix

$$\mathbf{Z} \triangleq \begin{bmatrix} \mathbf{I}_\eta & \mathbf{X} \\ \mathbf{X}^{\mathrm{T}} & \mathbf{Y} \end{bmatrix}. \tag{3.7}$$

Then, the network localization problem can then be stated as follows

$$\begin{aligned} \text{find} \quad & \mathbf{X} \in \mathbb{R}^{\eta \times T}, \mathbf{Y} \in \mathbb{R}^{T \times T} \\ \text{such that} \quad & \boldsymbol{e}_{ij}^{\mathrm{T}} \mathbf{Y} \boldsymbol{e}_{ij} = d_{ij}^2, \\ & \begin{bmatrix} \boldsymbol{a}_k^{\mathrm{T}} \,\vdots\, \boldsymbol{e}_j^{\mathrm{T}} \end{bmatrix} \mathbf{Z} \begin{bmatrix} \boldsymbol{a}_k \\ \hdashline \boldsymbol{e}_j \end{bmatrix} = d_{kj}^2, \\ & \mathbf{Y} = \mathbf{X}^{\mathrm{T}} \mathbf{X}, \\ & \qquad\qquad \forall k \leq n_{\mathrm{A}}; i, j \leq T. \end{aligned} \tag{3.8}$$

The SDP variation of the network localization problem formulated above is obtained [34] by relaxing (see app.A for further details) the equality constraint $\mathbf{Y} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$ into the positive semidefinite condition $\mathbf{Z} \succeq 0$. Under such relaxation and with elementary algebra introduced to eliminate the variables $\mathbf{X}$ and $\mathbf{Y}$ we arrive at

$$\begin{aligned} \text{maximize} \quad & 0 \\ \text{subject to} \quad & \mathbf{Z}_{1:\eta,1:\eta} = \mathbf{I}_\eta \\ & \operatorname{tr}(\mathbf{C}_{\mathrm{T}} \mathbf{Z}) = d_{ij}^2, \\ & \operatorname{tr}(\mathbf{C}_{\mathrm{A}} \mathbf{Z}) = d_{kj}^2, \\ & \mathbf{Z} \succeq 0, \end{aligned} \tag{3.9}$$

where $\mathbf{Z}_{1:\eta,1:\eta}$ is the $\eta \times \eta$ first-minor of $\mathbf{Z}$ and the constants $\mathbf{C}_{\mathrm{T}}$ and $\mathbf{C}_{\mathrm{A}}$ are defined as

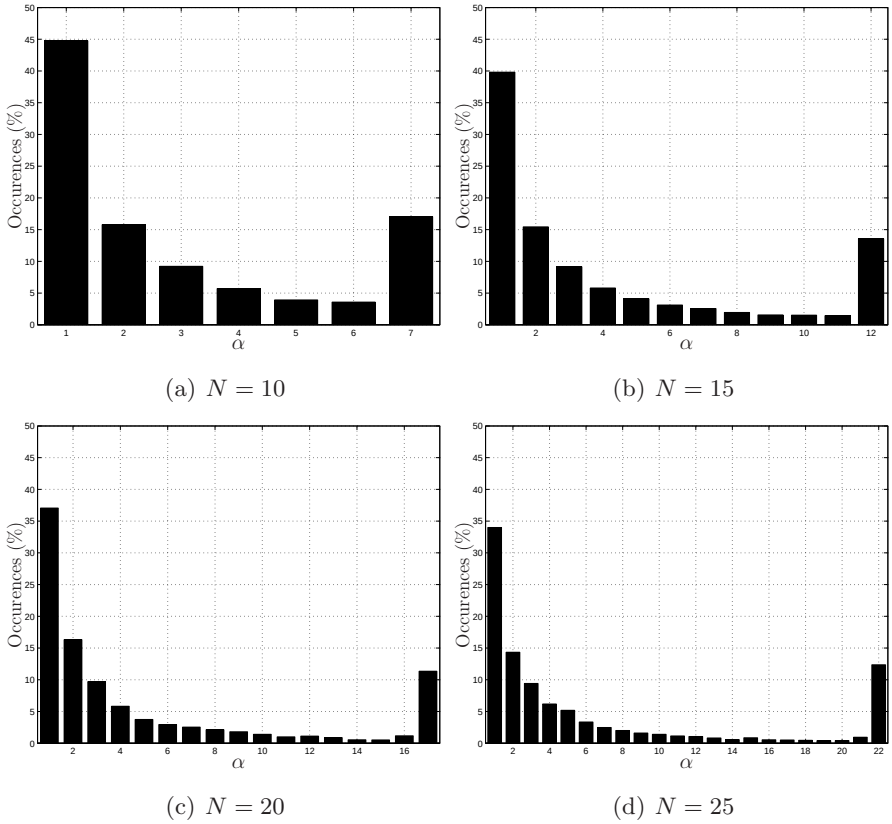
$$\mathbf{C}_{\mathrm{T}} \triangleq \begin{bmatrix} \mathbf{0}_{[\eta \times 1]} \\ \hdashline \boldsymbol{e}_{ij} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{[1 \times \eta]} \,\vdots\, \boldsymbol{e}_{ij}^{\mathrm{T}} \end{bmatrix}, \tag{3.10}$$

$$\mathbf{C}_{\mathrm{A}} \triangleq \begin{bmatrix} \boldsymbol{a}_k \\ \hdashline \boldsymbol{e}_j \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_k^{\mathrm{T}} \,\vdots\, \boldsymbol{e}_j^{\mathrm{T}} \end{bmatrix}. \tag{3.11}$$

The unique localizability theorem by So and Ye [34] can now be invoked.

Theorem 3.1.1 (Unique Network Localizability) *A network represented by the graph* $G([\mathbf{A}; \mathbf{X}], \mathbf{H})$ *is unique localizable iff* $\operatorname{rank}(\mathbf{Z}) = \eta$.

Proof See [34]. ∎



(a) $N = 10$ (b) $N = 15$

(c) $N = 20$ (d) $N = 25$

Figure 3.2: Distribution of the number of flip-ambiguous target nodes α for networks of different sizes $N = \{10, 15, 20, 25\}$.

Corollary 3.1.2 (Identification of Non-localizable Nodes) *All flip-ambiguous nodes of a non-uniquely localizable network* $G([\mathbf{A}; \mathbf{X}], \mathbf{H})$ *can be identified.*

Proof Biswas and Ye [36] demonstrate that the trace of the difference matrix $\mathbf{Y} - \mathbf{X}^{\mathrm{T}}\mathbf{X}$ can be used as a measure of the quality of localization. The smaller the trace, the higher accuracy of the estimation. In particular, the j-th diagonal element of $\mathbf{Y} - \mathbf{X}^{\mathrm{T}}\mathbf{X}$ is proportional to the accuracy of the position estimation of the j-th target node. ∎

Using Corollary 3.1.2 we are able to study the statistics of the number of flip-ambiguous nodes in randomly generated networks of different sizes. Notice that in the context under consideration, since all known distances d_{ij} and d_{kj} are error-free, the only reason for a node not to be localizable is due to flip-ambiguity. If the j-th diagonal element of $\mathbf{Y} - \mathbf{X}^{\mathrm{T}}\mathbf{X}$ is greater than 0, the corresponding node is flip-

ambiguous. To illustrate typical results obtained with a Monte Carlo analysis (10000 network realizations for each network size), histograms of the number of *flip-ambiguous* nodes α for various N are shown in Fig. 3.2. It is important to recall that in all cases considered the number of anchors is the minimum possible, *i.e.*, $n_{\mathrm{A}} = 3$.

As can be seen, the study shows that the number of flip-ambiguous target nodes that make a network non-uniquely localizable is, in the majority of cases, a small number, that is, $\alpha \ll N$. For example, for networks of sizes $N = 10$, it is found that $\alpha = 1$ approximately 45% of the realizations and $\alpha \leq 2$ over 60% of the realizations. Likewise, for $N = 25$, $\alpha \leq 3$ also for approximately 60% of the realizations. In all cases, it is evident that the number of ambiguous nodes is typically only a small fraction of the total number of nodes in the network.

This observation, which albeit empirical is solidly legitimized by the unique localizability test, suggests that the prevailing approach in the literature related to the flip-ambiguity problem in network localization [37], [38] may be too pessimistic. Indeed, current techniques aim to *avoid*, rather than *solve*, flip-ambiguity, by eliminating ambiguous targets from the topology over which localization is attempted. The approach is justified by the well-known fact that flip-ambiguity may lead to catastrophic localization errors, but has the drawback of leaving the remaining nodes without any location estimate, which may be unacceptable in many applications.

The results obtained from Corollary 3.1.2, however, motivated us to investigate a technique to resolve flip-ambiguities, ultimately enabling the localization of all nodes in the network. The mathematical rational behind such a technique, which is latter presented in subsection 3.1.3, is described in the sequel.

3.1.2 Localization with Genie-aided Edge-Bounding

Consider the randomly deployed set of nodes as described in subsection 3.1.1. The median distance between two nodes lying in an ℓ-by-ℓ square was shown in [39] to be

$$\bar{d}_2 = 0.512\ell. \tag{3.12}$$

Applying this result to the case where N nodes lye in the square, and using $\ell = 1$ yields

$$\bar{d}_N = 0.512\sqrt{2/N}. \tag{3.13}$$

Consider the network $G([\mathbf{A};\mathbf{X}],\mathbf{H})$ formed by such nodes under a connectivity range R. In the unitary square, a connectivity range $R_{\text{Max}} = \sqrt{2}$ ensures that the network is fully connected and thus, uniquely localizable. For $R < R_{\text{Max}}$, the unique localizability of $G([\mathbf{A};\mathbf{X}],\mathbf{H})$ clearly depends on the relationship between the connectivity range R and average pairwise distance $\bar{d}_N$, which for $N = 10$ and $N = 25$, for instance, are $\bar{d}_{10} \approx 0.23$ and $\bar{d}_{25} \approx 0.15$, respectively.

Here it is interesting to find a pair of connectivity ranges R_{U} and $R_{\text{L}} = R_{\text{U}} - \Delta R$ such that the network is uniquely localizable under R_{U} and not under R_{L}. In order to find the *edge bounds* R_{U} and R_{L}, consider the decreasing sequence

$$R_{\text{Max}} > R_{\text{Max}} - \Delta R > \cdots > R_{\text{U}} > R_{\text{L}},$$

where $\Delta R \ll R_{\text{Max}} - \bar{d}_N$. For the cases to be studied here ($10 \leq N \leq 25$), $R_{\text{Max}} - \bar{d}_N \gtrsim 1.2$, so that it is enough to set $\Delta R = 0.05$ in order to obtain sufficiently tight bounds.

Let $\mathbf{H}_{\text{U}}$ and $\mathbf{H}_{\text{L}}$ be the incidence matrices corresponding to the uniquely localizable and the non-uniquely localizable networks obtained with R_{U} and R_{L}, respectively. By comparing $\mathbf{H}_{\text{U}}$ against $\mathbf{H}_{\text{L}}$, one is able to identify the set of distances $\tilde{d}_{ij}$ which, if not measured, results in the network not being uniquely localizable. Thus define

$$\tilde{\mathbf{H}} \triangleq \mathbf{H}_{\text{U}} - \mathbf{H}_{\text{L}} = \left[\begin{array}{c|c|c} \mathbf{0} & \tilde{\mathbf{H}}_{\text{T}} & \mathbf{0} \\ \hline \mathbf{0} & \tilde{\mathbf{E}}_{\text{A}} & \tilde{\mathbf{E}}_{\text{T}} \end{array}\right], \tag{3.14}$$

$$\tilde{\mathbf{C}}_{\text{T}} \triangleq \left[\begin{array}{c} \mathbf{0}_{[\eta\times 1]} \\ \hline \tilde{\boldsymbol{e}}_{ij} \end{array}\right] \left[\begin{array}{c|c} \mathbf{0}_{[1\times\eta]} & \tilde{\boldsymbol{e}}_{ij}^{\text{T}} \end{array}\right], \tag{3.15}$$

$$\tilde{\mathbf{C}}_{\text{A}} \triangleq \left[\begin{array}{c} \boldsymbol{a}_k \\ \hline \tilde{\boldsymbol{e}}_j \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{a}_k^{\text{T}} & \tilde{\boldsymbol{e}}_j^{\text{T}} \end{array}\right], \tag{3.16}$$

where $\tilde{\boldsymbol{e}}_{ij}$ is the column vector of $\tilde{\mathbf{E}}_{\text{T}}$ whose i-th and j-th elements are 1 and -1, respectively, while $\tilde{\boldsymbol{e}}_j$ is the column vector of $\tilde{\mathbf{E}}_{\text{A}}$ whose j-th element is -1.

Now the network $G([\mathbf{A};\mathbf{X}],\mathbf{H}_{\text{L}})$, despite failing the unique localizability test of Theorem 3.1.1, can be localized using the SDP formulation of the network localization

given in (3.9), augmented by additional constraints. Specifically,

$$\begin{aligned} \text{maximize} \quad & 0 \\ \text{subject to} \quad & \mathbf{Z}_{1:\eta,1:\eta} = \mathbf{I}_\eta \\ & \mathrm{tr}(\mathbf{C}_\mathrm{T}\mathbf{Z}) = d_{ij}^2, \\ & \mathrm{tr}(\mathbf{C}_\mathrm{A}\mathbf{Z}) = d_{kj}^2, \\ & \cdots\cdots\cdots\cdots\cdots\cdots \\ & R_\mathrm{L}^2 < \mathrm{tr}\Big(\tilde{\mathbf{C}}_\mathrm{T}\mathbf{Z}\Big) \leq R_\mathrm{U}^2, \\ & R_\mathrm{L}^2 < \mathrm{tr}\Big(\tilde{\mathbf{C}}_\mathrm{A}\mathbf{Z}\Big) \leq R_\mathrm{U}^2, \\ & \cdots\cdots\cdots\cdots\cdots\cdots \\ & \mathbf{Z} \succeq 0. \end{aligned} \tag{3.17}$$

The algorithm summarized above is referred to as a flip-ambiguity-robust SDP-based network localization algorithm with edge-bounding via a Genie. This is in allusion to the fact that in practical situations, the bounds R_L and R_U, as well as the incidence matrices $\mathbf{H}_\mathrm{U}$ and $\mathbf{H}_\mathrm{L}$ cannot be known.

In any case, this theoretical exercise offers insight on the potential of resolving flip-ambiguity through the combination of semidefinite programming and edge-bounding, motivating the feasible solution dealt with in the next subsection. To illustrate, the performance of the algorithm summarized by (3.17) are studied in Figs. 3.3 through 3.5. To this end, define the *mean location error* for a network of size N with n_A anchors as

$$\bar{\varepsilon}(N; n_\mathrm{A}) \triangleq \frac{\|\hat{\mathbf{X}} - \mathbf{X}\|}{N - n_\mathrm{A}}, \tag{3.18}$$

where $\hat{\mathbf{X}}$ is the estimate of $\mathbf{X}$.

Next, Fig. 3.3 compares the mean location error as a function of the network size obtained with the unconstrained SDP algorithm of (3.9) against the corresponding error obtained with its flip-ambiguity-robust counterpart. Two curves are shown for each technique, one representing the average situation ($\alpha \leq T$), where any number of target nodes may be subject to flip-ambiguity, and another representing the best possible situation ($\alpha = 1$), where only one target node is not localizable (see subsection 3.1.1).

Notice that the location error generally decreases with the network size, which is explained by the fact that the probability of a large number of flip-ambiguous nodes $\alpha \to T$ decreases with N (see subsection 3.1.1).

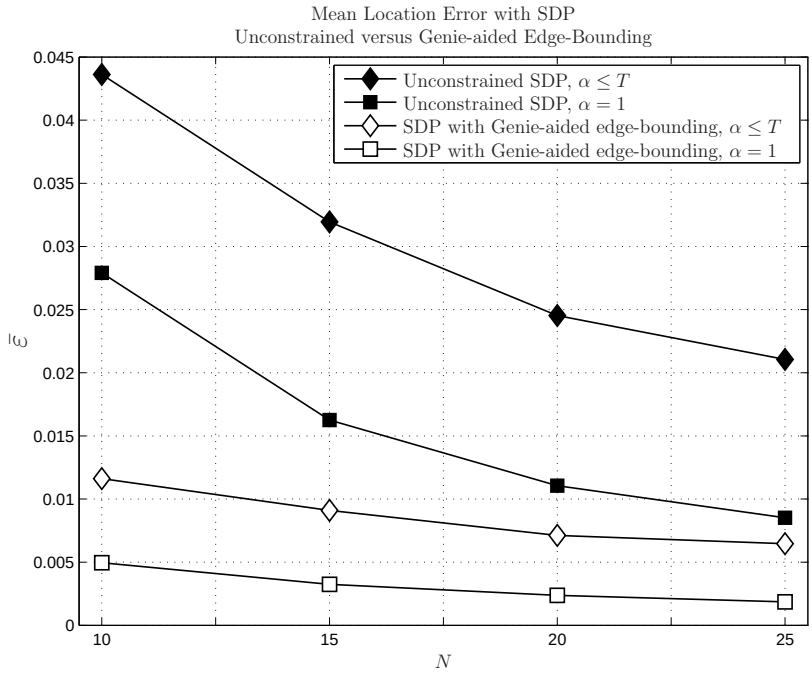


Figure 3.3: Mean location error for different α as a function of network size.

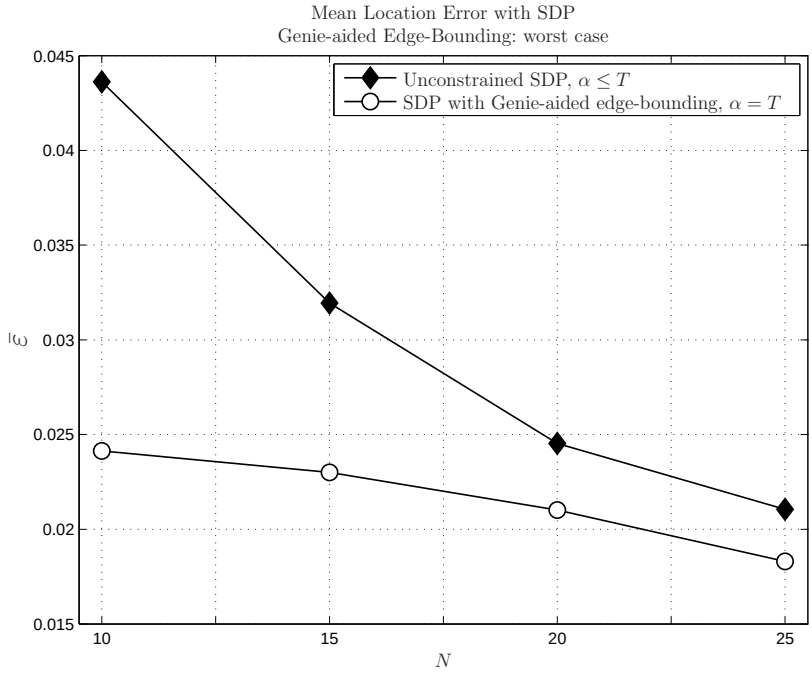


Figure 3.4: Mean location error for unconstrained SDP in the least ambiguous scenario ($\alpha = 1$) versus SDP with Genie-aided edge-bounding in the most ambiguous scenario ($\alpha = T$).

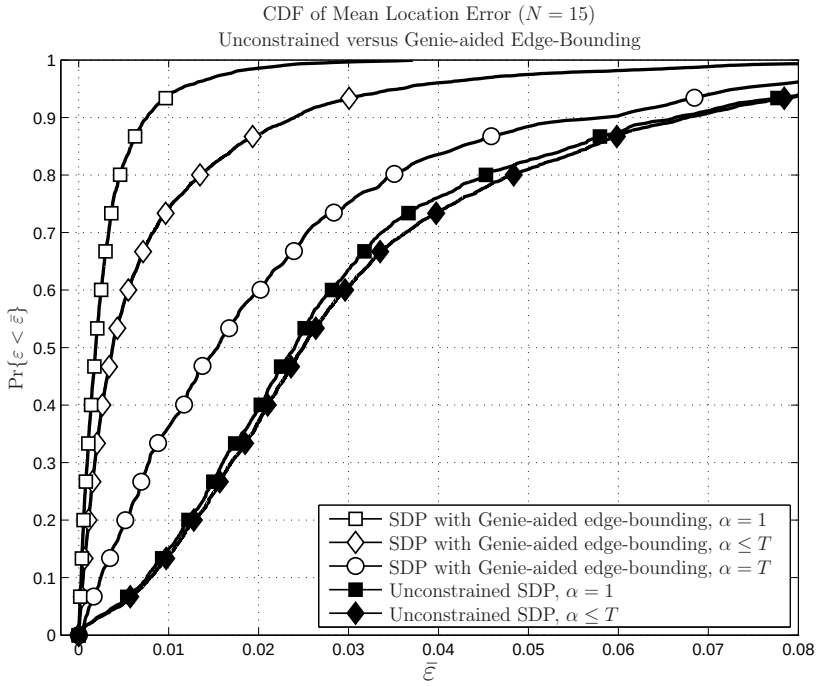


Figure 3.5: Cumulative density function of the mean location error for different number of α.

In any case, the figure clearly demonstrates that the (potential) improvement provided by the edge-bounding technique is very substantial.

This can be better illustrated by comparing the performances of the unconstrained SDP in the average situation ($\alpha \leq T$) against the edge-bounded SDP in the worst cases ($\alpha = T$), which is done in Fig. 3.4.

Finally, in Fig. 3.5, the Cumulative Density Functions (CDFs) of the mean location error $\bar{\varepsilon}$ obtained with the unconstrained and ambiguity-robust solutions, for a fixed network size, are compared. The figure illustrates the fact that in the presence of scenarios with flip-ambiguity, the performance obtained with the edge-bounded SDPs in typical conditions is close to that obtained in the best-possible non-uniquely localizable conditions, that is, when only one target node is flip-ambiguous. In contrast, the unconstrained SDPs performs nearly as bad, regardless regardless of the number of flip-ambiguous target nodes.

As a final remark to this subsection, it is possible to point out that the edge-bounded

SDPs-based flip-ambiguity-robust localization algorithm does *not* turn a non-uniquely localizable network into a uniquely localizable one, since the solution of obtained from (3.17) does not lye generally on the η dimensional space. Instead, what occurs is that a solution is obtained in a space of higher dimension, but whose projection onto the η-dimensional space is less likely to exhibit flipped estimates of node locations. Notice that this is not uncommon in SDP-based approaches, as discussed in detail *e.g.* in [35].

The fact that the solutions of (3.17) do not generally lye in $\mathbb{R}^{\eta\times T}$ is reflected on the non-zero location errors, which are encountered despite the assumptions of perfect distance estimates (for the distances that are measured).

3.1.3 Localization with Shortest-path Edge-bounding

The results in the preceding section, obtained with the SDP network localization augmented by constraints on a few additional unmeasured distances $\tilde{d}_{ij}$ illustrated the remarkable potential of the edge-bounding approach, but are obviously impractical since the information required, specifically $\tilde{d}_{ij}$ and $\mathbf{R}_{\mathrm{U}}$, cannot be obtained in practice. Inspired by those results, it is now easy to design an algorithm that captures the essence of the edge-bounding approach and is implementable.

To this end, it is possible to introduce the additional constraints not only to a few, but to *all* unmeasured distances and replace the constant and unknown tight upper-bound $\mathbf{R}_{\mathrm{U}}$ by the shortest paths between (computed using the Dijkstra's algorithm [40]) any two points whose distance was not measured. This approach, that can be referred to as Constrained SDP (CSDP), leads to

$$\begin{aligned}
\text{maximize} \quad & 0 \\
\text{subject to} \quad & \mathbf{Z}_{1:\eta,1:\eta} = \mathbf{I}_\eta \\
& \mathrm{tr}(\mathbf{C}_{\mathrm{T}}\mathbf{Z}) = d_{ij}^2, \\
& \mathrm{tr}(\mathbf{C}_{\mathrm{A}}\mathbf{Z}) = d_{kj}^2, \\
& \cdots\cdots\cdots\cdots\cdots\cdots \\
& R_{\mathrm{L}}^2 < \mathrm{tr}\Big(\mathbf{C}_{\mathrm{T}}^{\dagger}\mathbf{Z}\Big) \leq d_{ij}^{\dagger 2}, \\
& R_{\mathrm{L}}^2 < \mathrm{tr}\Big(\mathbf{C}_{\mathrm{A}}^{\dagger}\mathbf{Z}\Big) \leq d_{kj}^{\dagger 2}, \\
& \cdots\cdots\cdots\cdots\cdots\cdots \\
& \mathbf{Z} \succeq 0,
\end{aligned} \tag{3.19}$$

where $d_{kj}^{\dagger}$ and $d_{ij}^{\dagger}$ are the shortest-path distances from the k-th anchor to the j-th target node, and between the i-th and the j-th target nodes, respectively.

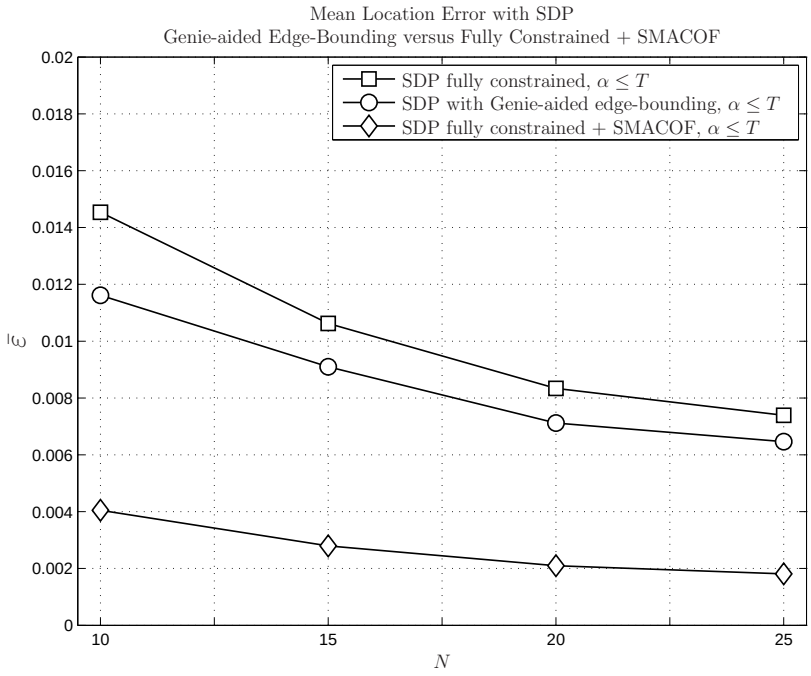


Figure 3.6: Mean location error as a function of network size for SDP with Genie-aided edge-bounding versus Fully constrained SDP and versus Fully constrained SDP+SMACOF.

Let $\mathbf{H}_{\mathrm{F}}$ be the incidence matrix corresponding to the fully connected network. It follows

$$\mathbf{H}^{\dagger} \triangleq \mathbf{H}_{\mathrm{F}} - \mathbf{H}_{\mathrm{L}} = \left[\begin{array}{c|c|c} \mathbf{0} & \mathbf{H}_{\mathrm{T}}^{\dagger} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{E}_{\mathrm{A}}^{\dagger} & \mathbf{E}_{\mathrm{T}}^{\dagger} \end{array}\right], \tag{3.20}$$

$$\mathbf{C}_{\mathrm{T}}^{\dagger} \triangleq \left[\begin{array}{c} \mathbf{0}_{[\eta\times 1]} \\ \hline \boldsymbol{e}_{ij}^{\dagger} \end{array}\right] \left[\begin{array}{c|c} \mathbf{0}_{[1\times\eta]} & \boldsymbol{e}_{ij}^{\dagger\mathrm{T}} \end{array}\right], \tag{3.21}$$

$$\mathbf{C}_{\mathrm{A}}^{\dagger} \triangleq \left[\begin{array}{c} \boldsymbol{a}_k \\ \hline \boldsymbol{e}_{j}^{\dagger} \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{a}_k^{\mathrm{T}} & \boldsymbol{e}_{j}^{\dagger\mathrm{T}} \end{array}\right], \tag{3.22}$$

where $\boldsymbol{e}_{ij}^{\dagger}$ is the column vector of $\mathbf{E}_{\mathrm{T}}^{\dagger}$ whose i-th and j-th elements are 1 and -1, respectively, while $\boldsymbol{e}_{j}^{\dagger}$ is the column vector of $\mathbf{E}_{\mathrm{A}}^{\dagger}$ whose j-th element is -1.

The performance of the algorithm described above can be further improved by feeding the solution obtained from the semidefinite program given in (3.19) into a good steepest-descent algorithm. This relates to the remark made on the last paragraph of the preceding subsection and results from the fact that the semidefine solution does

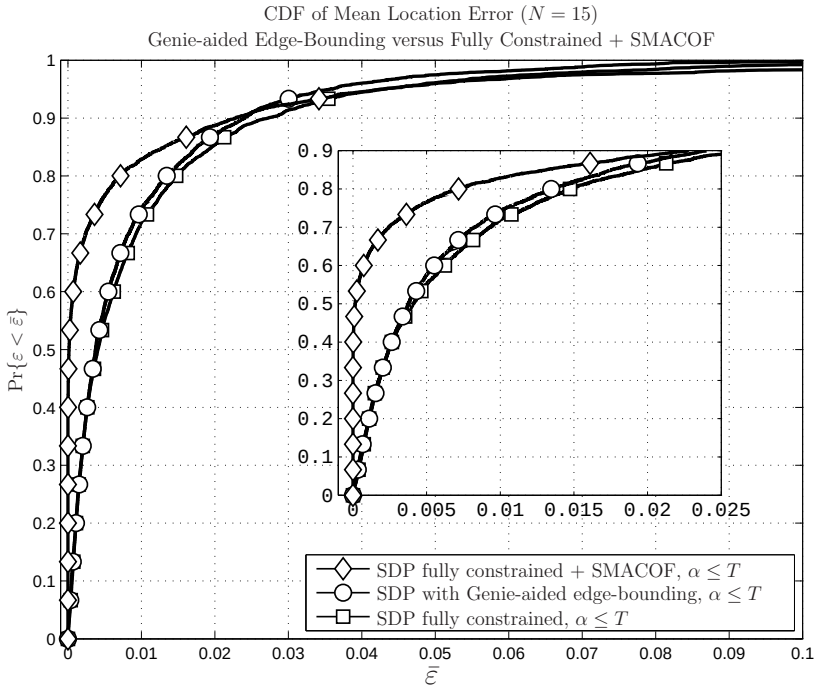


Figure 3.7: Cumulative density function of the mean location error for realizations of SDP and, inside the inner rectangle, a closer-up view of the curves.

not lye in η-dimensional space (see [35] and references thereby). Here SMACOF [41] is selected for such a *refining* algorithm.

Fig. 3.6 compares the mean location error obtained from (3.19), both with and without SMACOF-based refining against that of the algorithm described in section 3.1.2. It is found that the performance of the "raw" CSDP-based solution is slightly worse than that of the algorithm with a Genie. It is remarkable, however, that the performance of the feasible technique is not far off from the unfeasible one. Furthermore, it can the seen that the performance of the feasible solution after refinement via SMACOF is in fact better than that of the algorithm with a Genie and no refinement.

Finally, the CDFs of the mean location errors corresponding to the three schemes compared in Fig. 3.6 are plotted in Fig. 3.7. This final figure reveals that, in percentile terms, the feasible variation of the edge-bouded CSDP-based network localization algorithm with flip-ambiguity robustness is, to all effect, as good as the Genie-aided counterpart, and that the complete solution (including SMACOF refining) is even bet-

ter. Fig. 3.7 also shows that in over 80% of trials, for instance, the proposed technique is capable of maintaining the mean location error at the fraction of 1% of the largest dimension of the network, despite the flip-ambiguity problem that afflicts the networks studied.

3.1.4 Future Research Directions

Results obtained in this chapter, although very important, are empirical. Next step will be a deeper analysis of the flip-ambiguity problem in order to give a mathematical description of this issue. The final aim is to obtain bounds on the performance of the CSDP.

3.2 Multihop Localization Algorithm

In typical multihop networks, it is not realistic to assume that all nodes can estimate their position by their own. On the opposite, the majority of sensors in the network, to perform localization, is forced to cooperate with the small fraction that has an a-priori knowledge of its position.

The nodes have a limited radio range R, and the network is quite far to be fully connected: for this reason only few nodes, between the set of not localized, are linked to one or more that a-priori know their position. On the contrary, the majority of the nodes has none of them between their neighbours. Here a crucial problem is collecting target-to-anchor distances: this imply to define a metrics that can avoid or reduce bias [42] and error propagation.

In typical distributed cooperative algorithms currently considered for self localization in autonomic multihop network, nodes estimate their own location by iteratively collecting information on the location of and distances to neighboring nodes [43, 44, 45]. While such a message-passing concept is solidly based on sequential Bayesian inference, a characteristic problem of those algorithms is that the communication cost or complexity – measured in terms of the number of message exchanges required for convergence [46] – grows geometrically with the number N of nodes in the network. In addition to the resulting low efficiency, when the number n_A of nodes with known location (anchors) is small compared to the network size ($n_A \ll N$), convergence time easily exceeds typical network coherence times causing the localization algorithm to diverge. Last but not least, an inherent disadvantage of such algorithms is that cooperation comes at the cost of non-anonymity, since nodes must disclose their location to neighbors.

The aim of this section is to demonstrate that the aforementioned drawbacks are not a necessity of distributed cooperative localization algorithms in order to achieve good localization performances.

The multihop localization is not a totally new idea. The *Ad hoc Potisioning Algorithm* (APS), proposed by Niculescu and Nath [47, 48, 49], is drawn to be and extension of GPS [50] for indoor environments. It is based on classical multilateration, and the problem that addresses to solve is how to obtain, for every node, the distances to a sufficient number of landmarks to perform a GPS-like trilateration. Several methods of hop to hop distance propagation are studied, like *DV-hop propagation method* and

DV-distance propagation method. In the first case a classic distance vector exchange is performed, so that every node knows, in terms of hops, its distance from all the other nodes. Then, exploiting their a-priori knowledge of reciprocal locations, anchor nodes estimate the average hop-size and the table of distances is updated. Classic multilateration is finally performed with these updated distances. DV-distance propagation method is similar to the previous one, with the only difference that distance between neighbouring nodes is propagated in meters rather than in hops.

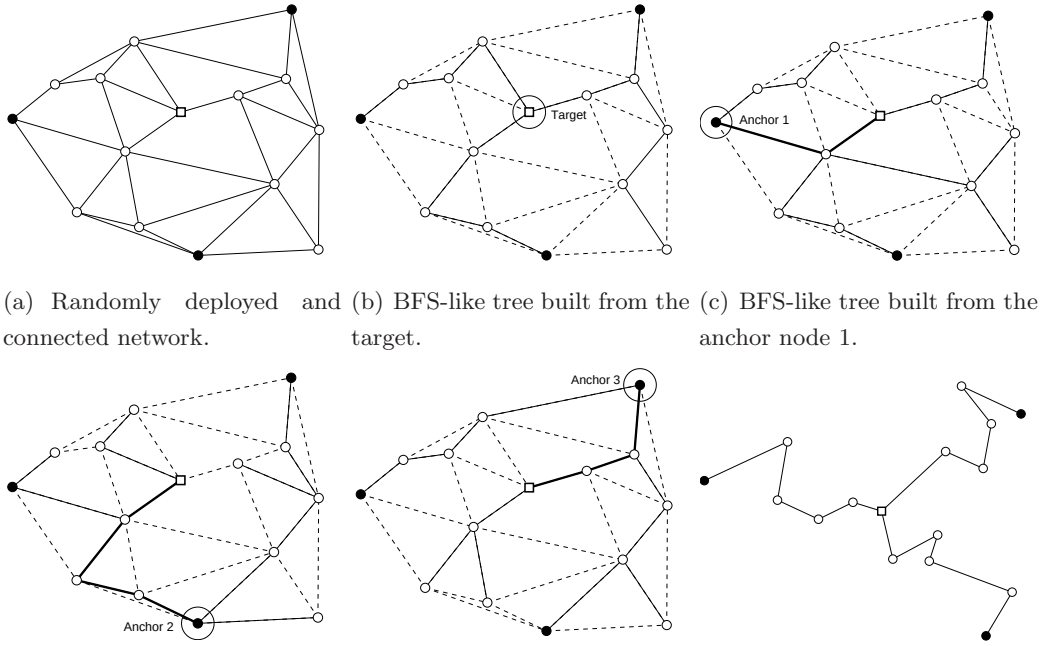
A very similar solution to APS is the *Robust Positioning* algorithm proposed by Savarese [51], but the author proposes a heuristic approach to detect "problematic" network topologies. It is a two-phase algorithm, composed by *Hop-TERRAIN*, based on the same concepts of DV-hop, and an iterative, local, refining phase.

Ad-Hoc Localization System (AHLoS), proposed by Savvides [52, 53], has the same distance propagation method as DV-distance, but localization is performed with Min-Max technique as proposed by Parker [54].

Another Multihop Localization Algorithm (MLA), based on multilateration, is introduced by Will [55]. Here each target computes the distances between itself and all the anchors, and only the shortest paths are stored. Finally 4 anchors are selected and multilateration is performed.

To reduce the overestimate in target-to-anchor distances is the aim also of Wong [45], whose solution is very similar to DV-hop but introducing the concept of *density awareness* to dynamically estimate distances in non-uniformly-distributed networks. A kind of awareness of the network is exploited also by Wei with his *COTA* framework [56], whit the usage of *confidence tags* to improve localization.

Here a self-localization algorithm is described in which any node seeking to find its own location simply collects the multihop distances between itself and a sufficient number of surrounding anchors, which can be performed efficiently, e.g. via distributed tree-based discovery algorithms [46] or geographic routing [57]. While the scheme is distributed (nodes find their own location) and cooperative (neighboring nodes must relay packets towards/from anchors), fundamental differences with respect to current distributed cooperative self-localization algorithms are: a) nodes do not repeatedly update/exchange location information with neighbors (with clear benefits for the security of the network [58]), but rather forward inquires on the location of anchors to the anchors themselves, via multihop routing; b) nodes do need not disclose their location to



(a) Randomly deployed and connected network.

(b) BFS-like tree built from the target.

(c) BFS-like tree built from the anchor node 1.

(d) BFS-like tree built from the anchor node 2.

(e) BFS-like tree built from the anchor node 3.

(f) Network connectivity model used during the simulations.

Figure 3.8: A typical realization of network connectivity (a), a typical BFS-like tree built by the target (b), typical BFS-like trees built by anchor nodes (c,d,e), and a typical multihop network model (f).

neighbors, and instead simply add each estimated hop distance to the routes length; c) the algorithm is not iterative, such that its "convergence" time equals the delay associated with the slowest route to the anchors.

3.2.1 Network Scenarios and Model

A typical example of a *multihop network* is represented in Fig. 3.8(a). Here, the network is understood as a set of N interconnected nodes randomly deployed. Only the location of a small fraction ($n_{\mathrm{A}} \ll N$) of nodes is known exactly *a priori*. These are referred to as *anchors* (marked by black dots) and $\mathbf{A} \triangleq [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_{n_{\mathrm{A}}}]$ is the node coordinate matrix carrying the column vectors with their corresponding locations. The coordinate matrix of the remaining nodes, $\mathbf{X} \triangleq [\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{N-n_{\mathrm{A}}-1}]$ is unknown. The i-th node $\boldsymbol{x}_i$ can however perform ranging measurements to any neighbor $\boldsymbol{x}_j$, obtaining

a noisy estimate $\tilde{d}_{ij}$ (measurement) of their mutual distance d_{ij}, defined as

$$d_{ij} \triangleq \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \sqrt{\langle \boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_i - \boldsymbol{x}_j \rangle}, \tag{3.23}$$

where $\|\cdot\|$ denotes the Euclidean norm and $\langle\cdot\rangle$ denotes inner product.

Without loss of generality it is possible to focus on the localization of the node $\boldsymbol{x}_0$, which will be referred to as the *target* and is depicted by a white square in Fig. 3.8. In the distributed scenario, the target node wants to estimate its own location computing the distances between itself and a sufficiently large subset of anchors ($n_{\mathrm{A}} \geq 3$). Since the target is not directly connected to the anchors, it has to determine those distances through the lengths of the multihop paths to the anchors.

For convenience (and no consequence in the analysis), it will be assumed that each target-to-anchor path has n_{H_i} hops. The estimated multihop distance from a target at $\boldsymbol{x}_0$ to an anchor at $\boldsymbol{a}_i$ is therefore

$$\tilde{p}_{0i} \triangleq \sum_{k=1}^{n_{\mathrm{H}_i}} \tilde{d}_{k-1\,k}, \tag{3.24}$$

where $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_{\mathrm{H}_i-1}})$ are the nodes involved in the path and $\tilde{d}_{n_{\mathrm{H}_i-1}\,n_{\mathrm{H}_i}}$ is the measured distance between anchor $\boldsymbol{a}_i$ and the last node of the path $\boldsymbol{x}_{n_{\mathrm{H}_i}-1}$.

Since the connectivity of a wireless network can not be determined a priori, due to traffic load, packet collisions and similar challenges, it will be assumed that such paths are the result of a distributed version of the well known Breadth First Search (BFS) algorithm [46], taking into account packet losses and collisions, which is referred to as BFS-like algorithm. Consequently, as can be observed from the examples in Figg. 3.8(b)-3.8(e), the branches of the resulting trees are not determined by the shortest paths between the root and the leaves, but rather by less "straight" paths generally growing from the root towards the leaves.

Fig. 3.8(b) illustrates paths obtained in connection to a distributed self-localization scheme, where the target builds its own BFS-like tree in order to find paths to all anchors. In contrast, Figg. 3.8(c)-3.8(e) illustrate the paths obtained in connection with a centralized systemic localization scheme, where anchors (assumed to be connected to a central station) build their own independent BFS-like trees. Obviously in both cases multihop target-to-anchor distances will be estimated under similar circumstances, namely, through irregular routes which result typically in $\tilde{p}_{0i} \gg d_{0i}$.

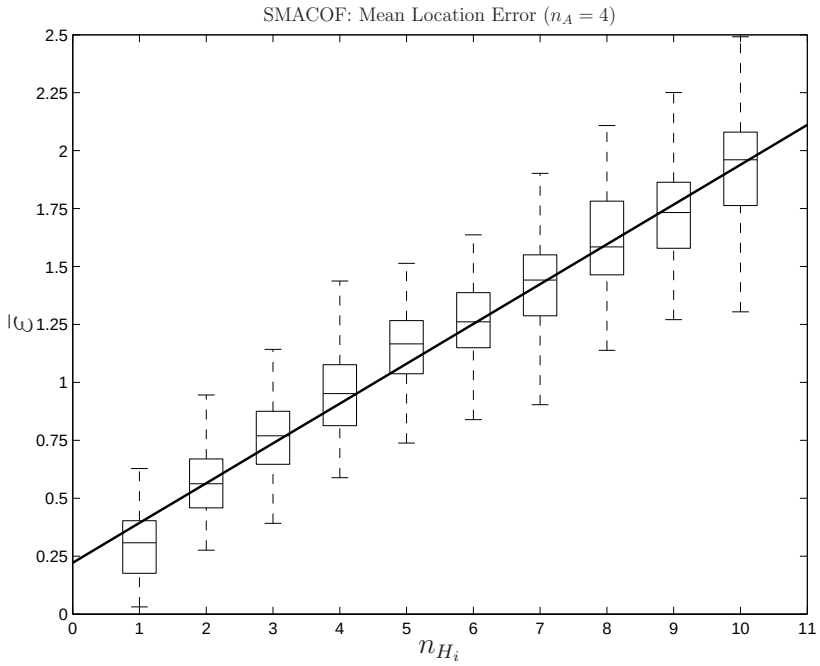


Figure 3.9: Linearized mean location error of SMACOF algorithm for networks with different n_{H_i}. Variance, max and min error are shown for every network size.

With basis on the latter, the performance of localization algorithms will be studied with target-to-anchor distances constructed using a model illustrated in Fig. 3.8(f), in which the target is located at the center of a circle of unitary[1] diameter ($d = 1$) where the anchors lye (randomly placed). The route from the target to the anchor at $\boldsymbol{a}_i$ is built by randomly selecting n_{H_i-1} nodes uniformly distributed inside a cone centered at $\boldsymbol{a}_i$.

3.2.2 A Typical Localization Algorithm in the Multihop Scenario

This subsection it will briefly review a classical localization algorithm, namely the SMACOF technique [41], known for its accuracy and efficiency, and studied its performance under the conditions described in subsection 3.2.1.

This algorithm attempts to find the minimum of a non-convex function by tracking the global minimum of the so-called *majorized convex functions* $\mathcal{T}(\hat{\mathbf{Q}}^{(m)}, \hat{\mathbf{Q}}^{(m-1)})$ which is successively constructed from the original objective and the previous solution

[1] This is without loss of generality and amounts to a normalization of all distance quantities.

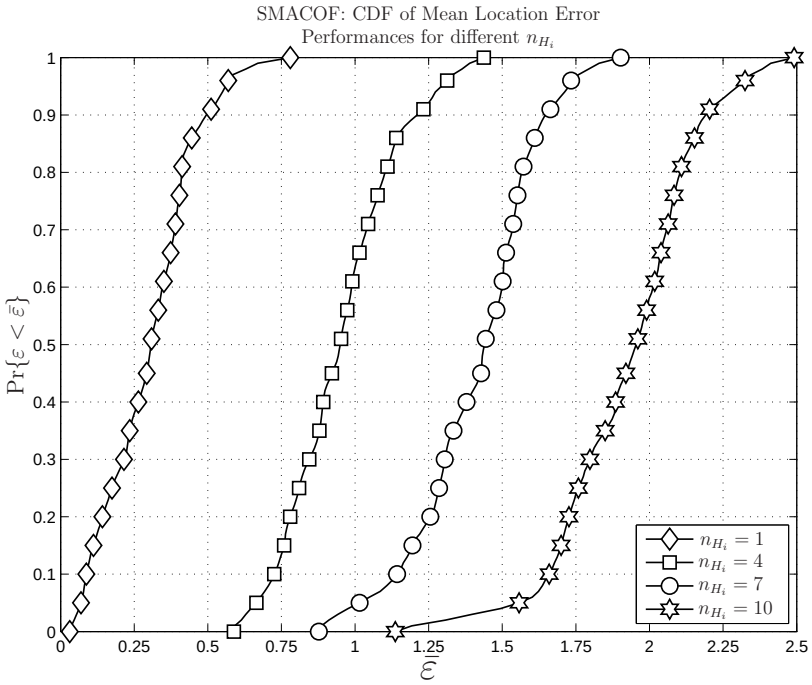


Figure 3.10: Cumulative density functions of the mean location error obtained with SMACOF for network with different number of hops and with $n_{\mathrm{A}} = 4$. From the left to right: $n_{\mathrm{H}_i} = 1, 4, 7, 10$.

obtained in the iterative procedure [59], and where $\hat{\mathbf{Q}}^{(m)}$ is the estimate of $\mathbf{Q} = [\boldsymbol{x}_0; \mathbf{A}]$ after the m-th iteration.

Let

$$p_{ij} = \|\boldsymbol{a}_i - \boldsymbol{a}_j\|, \tag{3.25}$$

and denote all estimated quantities by the corresponding letter topped with a $\hat{}$. Then, it follows:

$$\begin{aligned} \Im(\hat{\mathbf{Q}}^{(m)}, \hat{\mathbf{Q}}^{(m-1)}) \quad &= c + \mathrm{tr}\Big(\hat{\mathbf{Q}}^{(m)\mathrm{T}} \mathbf{U} \hat{\mathbf{Q}}^{(m)}\Big) \\ &\quad -2 \cdot \mathrm{tr}\Big(\hat{\mathbf{Q}}^{(m)\mathrm{T}} \mathbf{B}^{(m)} \mathbf{Q}^{(m-1)}\Big), \end{aligned} \tag{3.26}$$

where c is a constant that can be neglected, the entries of $\mathbf{U}$ are given by

$$u_{ji} = \begin{cases} n_{\mathrm{A}}\,, & i = j, \\ -1\,, & i \neq j, \end{cases} \tag{3.27}$$

while diagonal and above-diagonal elements of the symmetric matrix $\mathbf{B}^{(m)}$ are given by

$$b_{ji}^{(m)} = \begin{cases} \sum_{k=2}^{n_{\mathrm{A}}+1} \frac{\tilde{p}_{0k}}{\hat{p}_{0k}^{(n-1)}}, & i=1, j=1, \\ \sum_{\substack{k=1 \\ i \neq j}}^{n_{\mathrm{A}}+1} \frac{p_{j-1\,k}}{\hat{p}_{j-1\,k}^{(n-1)}}, & i=j, j \neq 1, \\ -\frac{\tilde{p}_{0\,i-1}}{\hat{p}_{0\,i-1}^{(n-1)}}, & i \geq j, j=1, \\ -\frac{p_{j-1\,i-1}}{\hat{p}_{j-1\,i-1}^{(n-1)}}, & i \geq j, j \neq 1. \end{cases} \tag{3.28}$$

At the m-th iteration the global minimum $\hat{\mathbf{Q}}_{\min}^{(m)}$ of the majored function $\mathcal{T}(\hat{\mathbf{Q}}^{(m)}, \hat{\mathbf{Q}}^{(m-1)})$ is computed via the Guttman transform [60],

$$\hat{\mathbf{Q}}_{\min}^{(m)} = \mathbf{U}^{\dagger}\mathbf{B}^{(m)}\hat{\mathbf{Q}}_{\min}^{(m-1)}, \tag{3.29}$$

where † denotes the pseudoinverse.

Fig. 3.9 illustrates the (linearized) mean location error $\bar{\varepsilon}$, its variance, its maximum and its minimum value, for a network with $n_{\mathrm{A}} = 4$, varying the number of hops n_{H_i}. As can be seen, $\bar{\varepsilon}$ is directly proportional to n_{H_i}. To increase the number of hops means also to increase the variance between the real target-to-anchor distances and the multihop paths. As consequence, there is a higher target-to-anchor distances estimate error. For a network characterized by $n_{\mathrm{H}_i} = 2$, $\bar{\varepsilon} \approx 0.5$, i.e., the mean location error is half of the size of the considered scenario.

The linear dependence of the mean location error from the number of hops can be seen also from Fig. 3.10, which illustrates the CDF of $\bar{\varepsilon}$ for networks with $n_{\mathrm{A}} = 4$ but different n_{H_i}. The four CDFs, obtained respectively for $1, 4, 7$ and 10 hops, are almost equispaced on the horizontal axis, meaning that to equal increase of n_{H_i} corresponds equal increase of $\bar{\varepsilon}$.

It follows that a classic localization algorithm like SMACOF is not suitable for scenarios with not-negligible target-to-anchor distances estimate error, which is typical of the multihop scenario. The result is a totally inaccurate estimate of target location.

These considerations motivate the decision to use, in the context of the multihop localization, algorithms more resilient and robust to the target-to-anchor distances estimate error.

3.2.3 Localization in Multihop Networks: Constrained SDP

Following the conclusion of the previous section, here are introduced and described two algorithms able to guarantee accurate localization performances even in presence of a not negligible target-to-anchor distances estimate error.

The first, Constrained SDP (CSDP), is based on the well known theory of the Semidefinite Programming (SDP). The second, Distance Contraction (DC), is developed on the base of the Least-Square (LS) localization algorithm [61].

The target-to-anchor squared-norms (or squared-distances) can be written as

$$\|\boldsymbol{a}_i - \boldsymbol{x}_0\|^2 = \left[\begin{array}{c:c} \boldsymbol{a}_i^{\mathrm{T}} & -1 \end{array}\right] \left[\begin{array}{c} \mathbf{I}_2 \\ \hdashline \boldsymbol{x}_0^{\mathrm{T}} \end{array}\right] \left[\begin{array}{c:c} \mathbf{I}_2 & \boldsymbol{x}_0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{a}_i \\ \hdashline -1 \end{array}\right] = d_{0i}^2, \tag{3.30}$$

where $\mathbf{I}_2$ is a 2-by-2 Identity matrix.

Define the matrix

$$\mathbf{Z} \triangleq \left[\begin{array}{cc} \mathbf{I}_2 & \boldsymbol{x}_0 \\ \boldsymbol{x}_0^{\mathrm{T}} & y \end{array}\right]. \tag{3.31}$$

Then, the multihop localization problem can be stated as follows

$$\begin{array}{rl} \text{find} & \hat{\boldsymbol{x}}_0 \in \mathbb{R}^{2\times 1}, y \in \mathbb{R} \\ \text{such that} & \left[\begin{array}{c:c} \boldsymbol{a}_i^{\mathrm{T}} & -1 \end{array}\right] \mathbf{Z} \left[\begin{array}{c} \boldsymbol{a}_i \\ \hdashline -1 \end{array}\right] = d_{0i}^2, \\ & y = \hat{\boldsymbol{x}}_0^{\mathrm{T}} \hat{\boldsymbol{x}}_0, \end{array} \tag{3.32}$$

The SDP variation of the network localization problem formulated above is obtained [34] by relaxing the equality constraint

$$y = \hat{\boldsymbol{x}}_0^{\mathrm{T}} \hat{\boldsymbol{x}}_0$$

into the positive semidefinite condition

$$\mathbf{Z} \succeq 0.$$

Under such relaxation and with elementary algebra introduced to eliminate the variables $\hat{\boldsymbol{x}}_0$ and y it possible to arrive at

$$\begin{array}{rl} \text{maximize} & 0 \\ \text{subject to} & \mathbf{Z}_{1:2,1:2} = \mathbf{I}_2 \\ & \mathrm{tr}(\mathbf{C}_{\mathrm{A}} \mathbf{Z}) = d_{0i}^2, \\ & \mathbf{Z} \succeq 0, \end{array} \tag{3.33}$$

where $\mathbf{Z}_{1:2,1:2}$ is the 2×2 first-minor of $\mathbf{Z}$ and the constants $\mathbf{C}_\mathrm{A}$ is defined as

$$\mathbf{C}_\mathrm{A} \triangleq \begin{bmatrix} \boldsymbol{a}_i \\ \hdashline -1 \end{bmatrix} \left[\boldsymbol{a}_i^\mathrm{T} \,\vdots\, -1^\mathrm{T} \right]. \tag{3.34}$$

As remarked in the previous section, the knowledge of d_{0i} is not available. It is necessary now to transform the equality of equation (3.33) in a set of two inequalities, using 2 constrains.

As lower bound it can be put simply 0, since it is known that two nodes can not occupy the same physical location, while as upper bound we can use the estimated target-to-anchor distances $\tilde{p}_{0i}$. Finally the problem becomes:

$$\begin{aligned} \text{maximize} \quad & 0 \\ \text{subject to} \quad & \mathbf{Z}_{1:2,1:2} = \mathbf{I}_2 \\ & \cdots\cdots\cdots\cdots\cdots \\ & 0 < \mathrm{tr}(\mathbf{C}_\mathrm{A}\mathbf{Z}) \leq \tilde{p}_{0i}, \\ & \cdots\cdots\cdots\cdots\cdots \\ & \mathbf{Z} \succeq 0. \end{aligned} \tag{3.35}$$

3.2.4 Localization in Multihop Networks: Distance Contraction

The DC algorithm is described in detail in [61], and is designed to minimize the effect of biased measurements in source localization problems under Non-Line-Of-Sight (NLOS) conditions. The inherent properties of the algorithm are, however, suitable to the multihop localization problem considered here.

In particular, the DC algorithm is a reformulation of the distance-based LS localization problem in which distances are contracted (subtracted by an empirically determined sufficiently large) by coefficients structured in relation to the null-space of the Grammanian matrix (or equivalently the relative angular matrix) describing the geometry in the problem (see [61]). The effect of the contraction is that the solution is found within a feasibility region $\mathcal{I}$ where the LS cost function is ensured to be convex. The limitation of the algorithm, however, is that it can only be applied to targets within the convex-hull defined by the anchors.

Here it is utilized a simplified version of the DC algorithm, which can be described as follows. First the feasibility region $\mathcal{I}$ is constructed from the intersection of the following inequalities:

$$(\hat{p}_{0i} - \tilde{p}_{0i}) \leq 0, \ \forall i, \tag{3.36}$$

then it searches, for every anchor at $\boldsymbol{a}_i$, its nearest point such that $\bar{\boldsymbol{a}}_i \in \mathcal{I}$. The contracted distances are computed as follows:

$$\bar{p}_{0i} = \|\bar{\boldsymbol{a}}_i - \hat{\boldsymbol{x}}_0\|. \tag{3.37}$$

Finally the cost function to minimize becomes:

$$f(\mathbf{A}, \hat{\boldsymbol{x}}_0) = \sum_{i=1}^{n_\mathrm{A}} \left(\bar{p}_{0i} - \hat{p}_{0i}\right)^2. \tag{3.38}$$

3.2.5 Performances Comparison and Discussion

The same set of data used to run simulations for SMACOF in subsection 3.2.2, is now used to evaluate the performances of both CSDP and DC algorithms.

Fig. 3.11 illustrates the mean location error $\bar{\varepsilon}$, for networks with $n_\mathrm{A} = 4$ anchors, varying the number of hops n_{H_i}. As it has been remarked before, the error on target-to-anchor distances estimate is proportional to n_{H_i}. It is evident that DC can perform accurate localizations regardless the variation of the hops, i.e., this algorithm is robust to the error that makes SMACOF useless in this scenario. The degradation of the performance proportional to n_{H_i} affects also the CSDP algorithm, even if not like SMACOF. Nevertheless CSDP proves to be efficient especially for a small number of hops ($n_{\mathrm{H}_i} \leq 3$), which is a typical case for wireless multihop networks. Under such conditions it even outperforms DC.

Mean location error of both algorithms for network with $n_{\mathrm{H}_i} = 4$, varying n_A, are presented in Fig. 3.12. It can be noted that the minimum number of anchor nodes ($n_\mathrm{A} = 3$) is sufficient to guarantee accurate performances. There is no substantial gain increasing n_A: from 3 to 8 anchor nodes the mean location error is reduced of 10% for DC and only of 5% for CSDP.

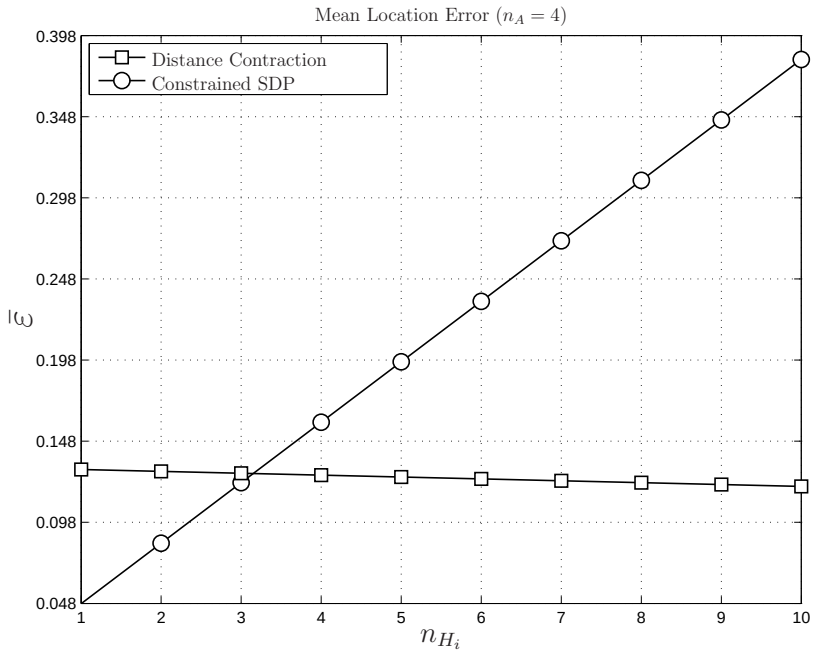


Figure 3.11: Mean location error as function of the number of hops n_{H_i} for networks with $n_{\mathrm{A}} = 4$ anchor nodes.

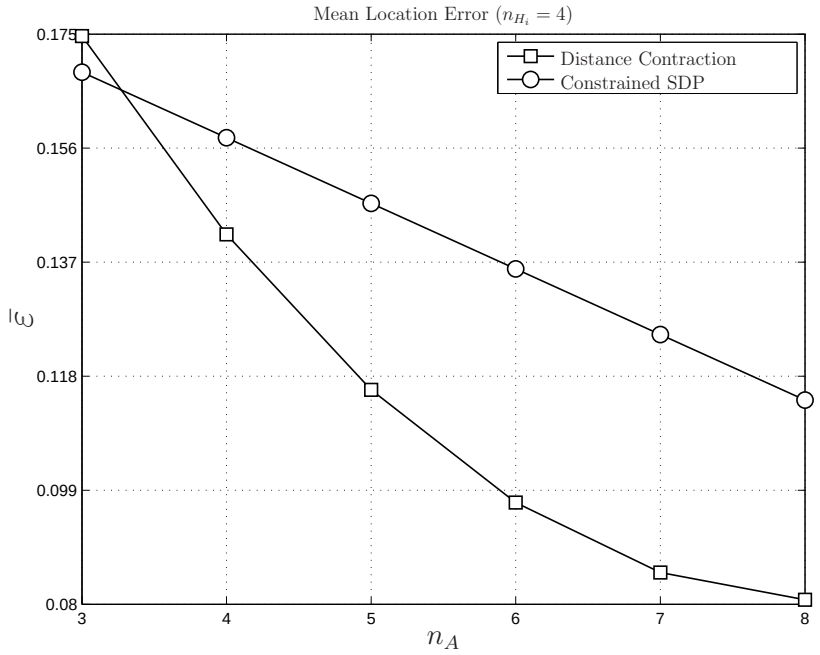


Figure 3.12: Mean location error as function of the number of anchor nodes n_{A} for $n_{\mathrm{H}_i} = 4$ hops networks.

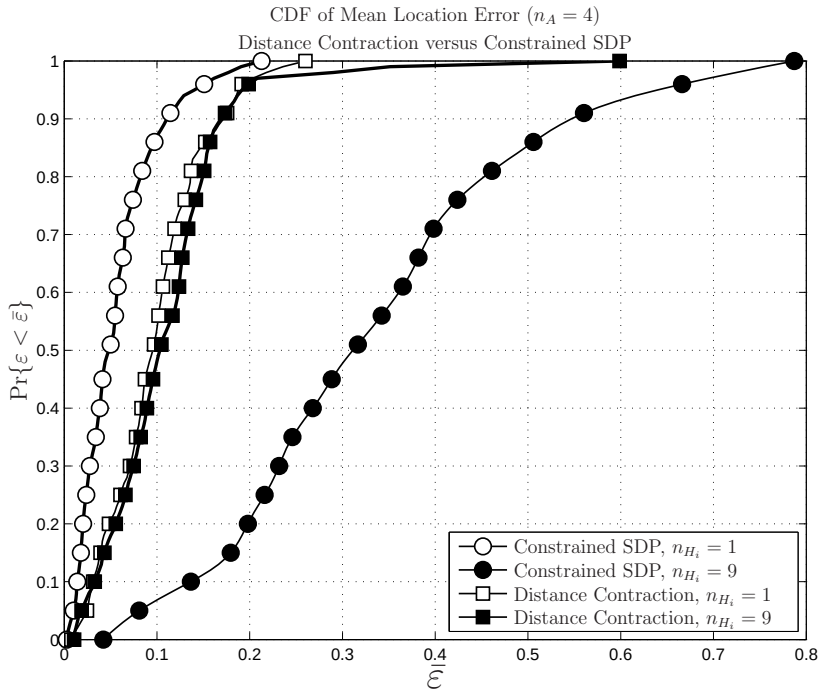


Figure 3.13: Cumulative density function of the mean location error for networks with different number of hops.

Finally Fig. 3.13 illustrates the CDF of the mean location error obtained for a network with $n_{\mathrm{A}} = 4$ and varying the number of hops: $n_{\mathrm{H}_i} = 1$ in the first case (white markers) and $n_{\mathrm{H}_i} = 9$ in the second (black markers). The effect of the number of hops on performances already highlighted in Fig. 3.11, is here even more evident. While for the CSDP the differences between the two CDF curves are substantial, in the case of DC they are almost perfectly matching.

After the simulation session, DC has proved to be a really robust and efficient algorithm, but with two main drawbacks. It can localize only targets inside the convex-hull determined by anchor nodes, and furthermore it can localize only one target per time. In particular the latter drawback can be annoying for the centralized systemic localization scheme (Figg. 3.8(c)-3.8(e)), where the central unit has to run the algorithm as many times as the number of targets to localize.

On the contrary, CSDP can localize more than one target per time and even outside the convex hull. This complementary suggests a possible combined approach for the two algorithms. CSDP can be used to obtain a first location estimate. If the target results to be inside the convex-hull, and if the medium number of hops is greater than 3, then DC can be run to refine the estimate.

3.2.6 Quantitative Comparison

This subsection provides a quantitative comparison of the MLAs, employing the well-known "three-sequences framework" introduced by Langendoen [62] to analyze the behaviour of the various solutions and to select the most general and useful algorithms.

The three-sequences are:

- *Distance Collection*: target-to-anchor measurements are collected;
- *Position Estimation*: a first estimation of target positions is provided;
- *Refining*: localization improvements (optional).

Langendoen analysis shows that between the MLAs present in literature no one is always able to outperform other solutions. Classic MLAs (as those cited in the introductions of this chapter) are instead very ad-hoc solutions, that can provide good performances only under particular scenarios and conditions.

Is therefore interesting use this analysis to understand what are the weak aspects of classic MLAs and, on the contrary, to justify the robustness of new solutions like Constrained Semidefinite Programming Localizaton Algorithm (CSDPLA) or DC.

The key-concept is that minimizing the error in target-to-anchor distances estimation is not the right approach. In general, due to the lack of connectivity it is not possible to have a negligible error: as a consequence the traditional localization algorithms are not well suitable for multihop networks. It is in fact demonstrated by Whitehouse [63, 64], that small variations in the ranging model in multihop networks can lead to large variations in localization error. Since a perfect ranging model has not been proposed for largely irregular topology like the multihop networks, most likely there will not be acceptable localization errors.

3.2.6.1 Phase I: Distance Collection

CSDPLA uses the same distance collection technique than DV-distance algorithm. Recalling the noisy measurement (3.23) of the mutual distance between a pair of nodes at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, follows that the estimated multihop distance from a target at $\boldsymbol{x}_0$ to an anchor at $\boldsymbol{a}_i$ is therefore

$$\tilde{p}_{0i} \triangleq \sum_{k=1}^{n_{\mathrm{H}_i}} \tilde{d}_{k-1\,k}, \tag{3.39}$$

as in (3.24).

It is quite obvious that

$$\tilde{p}_{0i} \geq p_{0i}, \tag{3.40}$$

where p_{0i} is the Line of Sight (LOS) distance between target at $\boldsymbol{x}_0$ to an anchor at $\boldsymbol{a}_i$, and equality is valid only iff all the intermediate nodes lye on the line between them. The estimation error on each path is proportional to the number of hops involved, as demonstrated in [65].

Disequation shown in (3.40) is not anymore valid for DV-hop. Equation (3.24) becomes

$$\tilde{p}_{0i} \triangleq n_{\mathrm{H}_i} \bar{c}_i, \tag{3.41}$$

where $\bar{c}_i$ is estimated average length of a single hop in the path to the i-th anchor.

Defining $\boldsymbol{a}_j$ the location of the farthest anchor node from the one in $\boldsymbol{a}_i$, the single hop length is estimated as follows:

$$\bar{c}_i \triangleq \frac{\|\boldsymbol{a}_i - \boldsymbol{a}_j\|}{n_{\mathrm{H}_i} + n_{\mathrm{H}_j}} = \frac{\sqrt{\langle \boldsymbol{a}_i - \boldsymbol{a}_j, \boldsymbol{a}_i - \boldsymbol{a}_j \rangle}}{n_{\mathrm{H}_i} + n_{\mathrm{H}_j}}. \tag{3.42}$$

This technique can lead, for some irregular topologies, event to underestimated target-to-anchor distances. In this case there is no more proportionality between the number of hops and the relative estimated distance error. This averaging-technique has positive effects for quasi-isotropic networks even with a high number of intermediate hops; on the contrary the performances are affected by highly irregular topologies or by targets lying outside from the convex hull defined by the anchor nodes.

3.2.6.2 Phase II: Position Estimation

In their original formulation, DV-hop and DV-distance use lateration techniques, like GPS [50] to perform Position Estimation. However, Li [66] has shown how, without the knowledge of the statistical signal model, Multidimensional Scaling (MDS) techniques are more efficient than Maximum-Likelihood Estimators (MLEs) like GPS. For this reason, for both DV-hop and Dv-distance algorithms, the SMACOF technique has been employed. As described in section 3.2.2, SMACOF is an MDS algorithm known for its accuracy and efficiency [59].

As previously remarked, main difference between CSDPLA and the classic MLAs is Position Estimation-phase. CSDPLA employs the SDP theory, founding a solution (i.e. target position) in a multidimensional space and projecting it onto the bidimensional space. This is the fundamental reason why CSDPLA is less sensitive to measurement errors coming from the Distance Collection-phase. Mathematical details have been provided in section 3.2.3.

3.2.6.3 Phase III: Refinement

Niculescu and Nath [47] do not employ a Refinement-phase, as on the contrary done by Savarese [51], whose solution is very similar to DV-hop. In this last case, only inter-node distances are considered, and local laterations are performed: all node position estimates are pushed toward local minima. This phase is therefore unnecessary with SMACOF, since nodes position are already estimated within local minima.

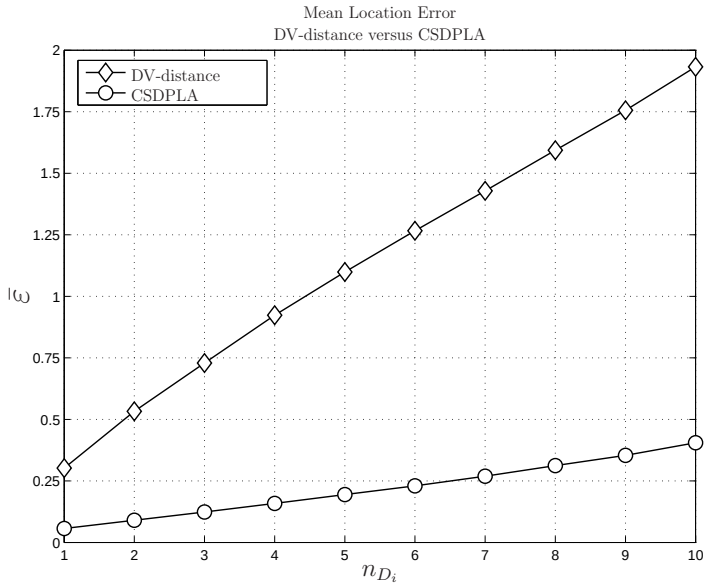


Figure 3.14: Mean location error as function of the number n_{D_i} of intermediate nodes.

On the contrary, for CSDPLA a local refinement, based on solutions like SMACOF, is possible [67]. Since the aim of this subsection is to give a quantitative comparison of MLAs,the Refinement-phase is not performed neither for CSDPLA, clarifying how it outperforms other solutions "as it is".

3.2.6.4 CSDPLA, DV-distance and DV-hop Simulation Comparison

In order to capture the performances under different network topologies, two different simulation scenarios, always according to the model in Fig. 3.8(f) and with a Gaussian Noise on ranging measurements with variance $\sigma^2 = 0.1$, have been implemented.

In the first case, the performances of DV-distance, compared to CSDPLA, have been analyzed. The DV-distance algorithm is very sensitive to the accumulation of error in multihop distances collection. To highlight this drawback, a network where all the anchors lye exactly on a circle, randomly placed, has been chosen, and the target is randomly placed around the circle center. The number of intermediate hops is the main simulation parameter: more the intermediate hops, more the error in target-to-anchor distances estimation.

As can be observed by Fig. 3.14, DV-distance algorithm suffers an heavy depen-

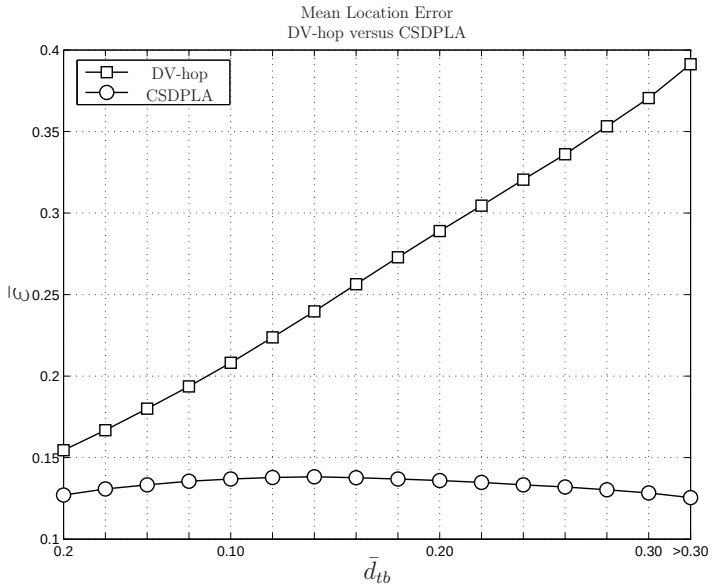


Figure 3.15: Mean location error as function of the distance $\bar{d}_{tb}$ of the target from the barycentre of the network.

dence from the error in target-to-anchor distances estimation. Even with only one intermediate node the mean location error in position estimate is not negligible (more than the 25% of the scenario dimension), and for three hops, i.e. a very common realization for WSNs, the mean location error is 0.75, totally unacceptable. On the contrary, the dependence of CSDPLA from the number of intermediate hops is less relevant, and for $n_{H_i} = 8$ the mean location error is the same than DV-distances for $n_{H_i} = 1$.

DV-hop algorithm performs instead very well for isotropic topologies like those considered in the previous scenario. This solution does not suffer error accumulation in distances collection and the performances are not related to the number of intermediate hops. In order to capture the failures of this algorithm, the anchor positions have been changed; now anchor nodes are randomly placed on three different circles centred on the same point, causing a variation in the isometry of the network. The distance $\bar{d}_{tb}$ of the target, always randomly placed around the circles center, from the barycentre of the network has been computed. This parameter $\bar{d}_{tb}$ can be used to classify the isometry of the network: greater the distance, more non-isometric the topology.

Results in Fig. 3.15 show how DV-hop suffers the dependence from the topology, while the performances of CSDPLA are related only to n_{H_i} and totally insensitive to

$\bar{d}_{tb}$. In the normalized scenario (the mean diameter of the three circles is $d = 1$), if the distance of the target from the barycentre of the network is 0.2 or less, the mean location error of DV-hop is 0.15, almost the same that for CSDPLA. Increasing the value of $\bar{d}_{tb}$, the performances of DV-hop deteriorate rapidly. Even in this scenario, as in the previous, CSDPLA can easily outperforms traditional MLAs.

3.2.7 Future Research Directions

Next activity in the framework of MLAs will be the performance comparison with the traditional message-passing algorithm. It is indeed useful to quantify the advantages of these new MLAs compared to the message-passing family both in terms of reduction of messages exchanged over the network and time-computation. The second action point focused on obtaining a bound on the performance of these new MLA and last, but not least, is the extension of MLA to the anchor-free networks.

4

Tracking

Target tracking is one of the non trivial applications of WSNs which is set up in the areas of field surveillance, habitat monitoring, indoor buildings, and intruder tracking [14].

In these fields the open challenges are multiple and related to energy efficient, computational cost, security aspects and above all reliability of the system. To track a target is even more complex than simply localize it, since the system must have a "memory" to store data, and at every instant the decision about the position must be taken according to the previous. It is clear that this memory introduces a relevant complexity in the system respect to the localization, and this complexity can have very negative effects. With the computation cost also the response-time of the system can grow dramatically, the energy consumption can be unacceptable for WSNs and the hardware requirements to face all these problems can cause increase the cost of a network.

A basic distinction between tracking systems is done: *active* [68], like the one presented in the following, and *passive* systems. In active systems, the target collaborates with the external system to its localization and tracking, in order to get access to some kind of services. On the contrary, in passive system, the targets do not want, or simply can not, cooperate with the network.

4.1 Active Tracking

In the last ten years the necessity of cheap and accurate indoor localization techniques has grown enormously. In many scenarios, museums, big shops, hospitals, hotels or big public buildings, the real time knowledge of the user's location can be helpful to provide a better service.

Indoor positioning systems based on IEEE 802.11 wireless networks gained a large attention in the few past years, due to their remarkable advantages in terms of costs and flexibility with respect to positioning systems based on ad-hoc infrastructures. Various localization techniques are summarized in [69], as well as some of the most interesting application contexts. The simplest and diffused approach with Wireless Local Area Network (WLAN) infrastructure is based on measure of the signal strength to derive user's position [70]. An improvement can be obtained with the *Joint Clustering technique* [71], that uses the clustering of locations to reduce the computational cost of the algorithm. Another technique is the statistical approach through *Bayesian filters* as proposed in [72]. The Bayesian filters offer a powerful mathematical tool for the location problem within a wireless sensor network and could be considered a near optimal solution for indoor positioning problems. However as shown in [73], in order to set-up a positioning system based on a Bayesian filter, it is necessary to determinate both a well-accurate mobility model, and a perception model. Furthermore as will be shown in the following, it is not easy to determine exactly these two models and all the related parameters: a long and time-expensive tuning phase is required and an imperfect choice of the models can significantly affect the precision of the positioning system.

In this section, it is introduced a new tracking algorithm which provides a good trade-off between complexity and performance. The proposed algorithm is inspired by the sequential decoding algorithm proposed in [74] and permits a fine adjustment (*tuning*) of some parameters to face the different user's mobility characteristics.

4.1.1 Positioning Based on Measures of Signal Strength

A widely-diffused approach consists of collecting samples of the power of the signals emitted by each of the N Access Points (APs), in a set of M points of the indoor environment. The samples are then organized in a database with M entries, and each entry is a vector of N Received Signal Strength Indicators (RSSIs). Such a database

(*radio map* of the environment) will constitute the signal space used by the generic user during the positioning phase [70].

Define

$$\mathbf{B}^{(i)} = \left(B_0^{(i)}, B_1^{(i)}, \ldots, B_{N-1}^{(i)}\right)$$

the received signal strength profile for the ith entry. The element $B_j^{(i)}$ is the power (measured in dBm) of the signal due to the jth AP and received at the ith position of the radio map database. The radio map

$$\mathcal{B} = \left\{\mathbf{B}^{(i)}\right\}_0^{M-1}$$

is therefore given by the collection of the received signal strength profiles corresponding to the M points of the indoor environment. Assume the radio map is available at the generic user. Denoting by

$$\mathbf{R}(n) = (R_0(nT_s), R_1(nT_s), \ldots, R_{N-1}(nT_s))$$

the signal strength profile measured by the user at time nT_s (where n is integer and T_s is the sampling period), and location S, the purpose of the algorithm is to produce an estimate of S based on the identification of the database entry $\mathbf{B}^{(i)}$ which best matches the received signal strength profile. The associate position S_i is taken as an estimate of S.

Basic algorithms use the Nearest Neighbor(s) in Signal Space (NNSS) technique to pick the radio entry that best matches the received vector profile [69, 70]. The Euclidean distance between $\mathbf{R}(n)$ and each entry of the radio map is computed; the point corresponding to the entry at minimum Euclidean distance is then chosen as estimated position. Although the algorithm proposed in [70] provides a simple approach to the user's position estimation, the main limits of its first release are due to:

- the necessity of an updated radio map, which requires a large amount of work by the positioning system manager;
- the absence of a user's position tracking mechanism.

While the first issue can be partially faced by the use of proper propagation models, able to predict the received signal strengths for a given position, the second issue requires the definition of a tracking algorithm with a reasonable trade-off between the

computational complexity and the performance improvement. It is in fact quite obvious that the position of a generic user at time nT_s is related to his position at the previous time instant $(n-1)T_s$ (i.e., the user's position is evaluated every T_s seconds or, in an event based approach, at each *event*). Therefore, the knowledge of the user's position at time $(n-1)T_s$ can be exploited to improve the accuracy of the position estimation at time nT_s. Moreover, a tracking algorithm can take advantage from the physical constraints (e.g., the walls) imposed by the real environment to the user's mobility.

4.1.2 Positioning Environment and Infrastructure

The tests have been carried out at the building of the University of Bologna at Cesena, as indicated in Fig. 4.1. There are 23 rooms of different size and a 'U' hallway. It has been mapped 2 or 4 points for every room depending on its size. Other 14 points have been mapped in the hallway, for a total of 74 points, corresponding to 74 radio map entries. This scenario is heterogenous also for the different types of furniture and the size of various walls.

It has been used 4 APs, operating in infrastructure mode and IEEE 802.11b/g-compliant, symmetrically disposed in the hallway: two on the main side, one on the left and the other on the right. The radio entries were collected using a laptop with an integrated Network Interface Card (NIC) 802.11b/g compliant.

The sampling phase, which leads to the construction of the radio map, lasted several days, in order to get accurate measurements. In fact, radio scenarios can be very different also during different hours in the same day: for example a crowd along the hallway or inside the office can significantly modify the received power levels.

For the algorithms test and implementation, the dynamic link library provided in [75] has been modified in order to acquire the RSSIs.

4.1.3 A New Tracking Algorithm

The new tracking algorithm presented in this section is based on a time-varying trellis description of the indoor environment, where each state (*node*) of the trellis corresponds to a physical location (*entry of the radio map*).

The possibility of using event-driven transition instead of time-driven transition has been considered. This solution can reduce remarkably the complexity of the tracking

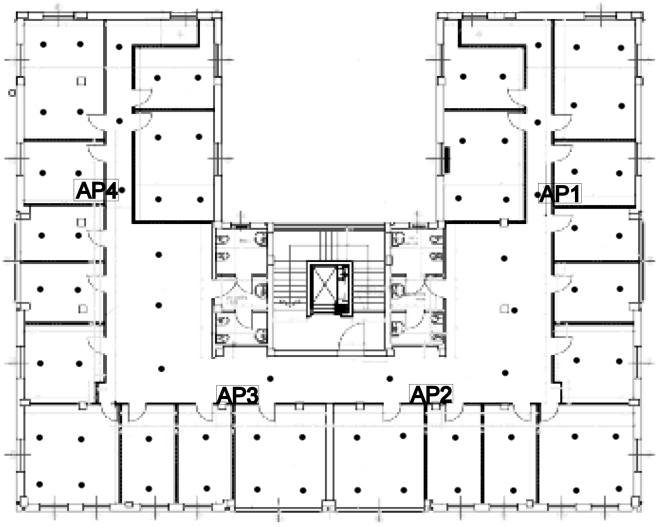


Figure 4.1: Testbed scenario with dots corresponding to entries in radio map and with APs locations.

algorithm: in such a case, the system is allowed to change its internal state only if there is a significant variation in the RSSI.

Define

$$\mathbf{B}^{(i)} = \left(B_0^{(i)}, B_1^{(i)}, \ldots, B_{N-1}^{(i)}\right)$$

being the received signal strength profile for the ith entry and the radio map

$$\mathcal{B} = \left\{\mathbf{B}^{(i)}\right\}_0^{M-1}.$$

The transition between two generic states, denoted by S_i and S_j, is allowed if and only if S_i and S_j correspond to physically-adjacent locations that are not separated by obstacles (e.g., walls). Therefore, define

$$S_i \in \mathcal{N}(S_j),$$

where $\mathcal{N}(S_j)$ is the set of locations reachable from S_j (*neighborhood* of S_j in the following).

Note that the complexity of the tracking algorithm would inevitably deal with the trellis dimensions (e.g., number of states, number of transitions to be considered, etc.) that is generally so large to prevent practical implementations. The main idea which

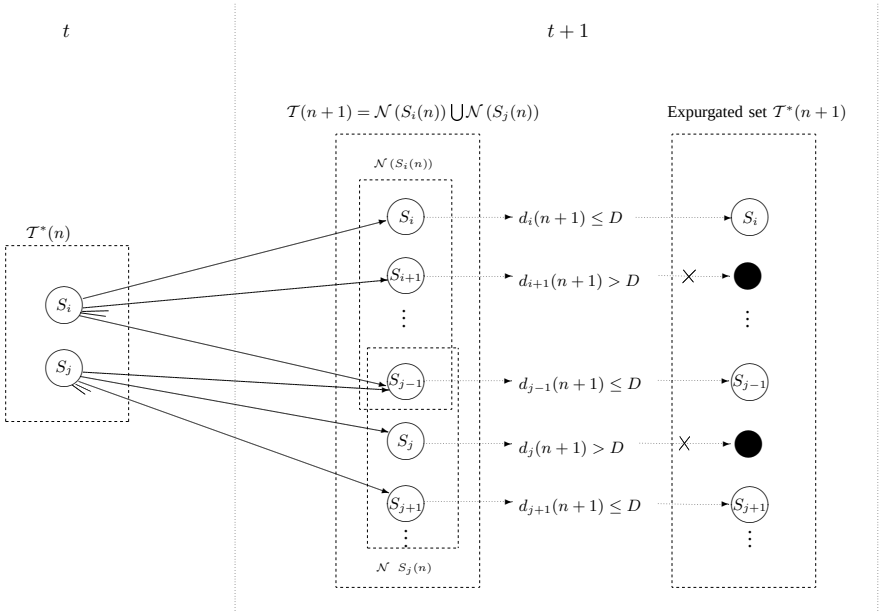


Figure 4.2: Trellis growth and expurgation.

resides in the proposed algorithm consist of a trellis-reduction (expurgation) procedure which dynamically eliminates the less probable states at a given time instant. Therefore, the trellis-reduction results in a low-complexity tracking algorithm that could be easily implemented. Define now $\mathcal{T}(n)$ the set of states present in the $n-$th section of the trellis, prior to the trellis expurgation. The trellis-reduction procedure will produce the $L(n)$ most probable states from $\mathcal{T}(n)$. The expurgated set of states (referred as *active states*) is denoted by $\mathcal{T}^*(n)$. Obviously, $\mathcal{T}^*(n) \subseteq \mathcal{T}(n)$ and $|\mathcal{T}^*(n)| = L(n)$.

4.1.3.1 Initialization

The initialization proceeds as follows. $\mathcal{T}(0)$ consists of the set of M possible states, corresponding to the M entries of the radio map. It is assumed that there is no a-priori knowledge on the initial user's location. Once the RSSIs vector $\mathbf{R}(0)$ is received, the Euclidean distance $d_j(0)$ between $\mathbf{R}(0)$ and the $j-$th entry of the radio map is computed as

$$d_j(0) = \| \mathbf{R}(0) - \mathbf{B}^{(j)} \|, \tag{4.1}$$

for $j = 0 \dots M-1$. The expurgated set $\mathcal{T}^*(0)$ is then produced, selecting the $L(0)$ entries at the lower distances. The choice of the initial number of active states, $L(0)$,

is a design parameter. Moreover denote by $\mu_j(n)$ the accumulated metric for the jth state at the time nT_s. At the initialization we set $\mu_j(0) = 0\ \forall j$.

4.1.3.2 Trellis Construction

The tracking algorithm proceeds with a recursive trellis growth. Assume $\mathcal{T}^*(n)$ to be the set of active states at time nT_s. The set of states (prior to the expurgation) at $(n+1)T_s$ is obtained as

$$\mathcal{T}(n+1) = \bigcup_i \mathcal{N}\left(S_i(n)\right), \tag{4.2}$$

where the union is over i such that $S_i(n) \in \mathcal{T}^*(n)$. Hence, $\mathcal{T}(n+1)$ is given by the union of all the neighbors of the active states at time nT_s. The edges linking the nodes in $\mathcal{T}^*(n)$ and the nodes in $\mathcal{T}(n+1)$ are drawn according to the topology of the indoor environment, i.e., the states $S_i(n) \in \mathcal{T}^*(n)$ and $S_j(n+1) \in \mathcal{T}(n+1)$ are connected if and only if $S_j(n+1) \in \mathcal{N}\left(S_i(n)\right)$.

4.1.3.3 Trellis Expurgation

Consider the case in Fig. 4.2. Once $\mathcal{T}(n+1)$ has been determined, the Euclidean distance $d_j(n+1)$ is computed between RSSIs vector $\mathbf{R}(n+1)$ and the radio map entry $\mathbf{B}^{(j)}$, for each active state $S_j(n+1) \in \mathcal{T}(n+1)$, as

$$d_j(n+1) = \| \mathbf{R}(n+1) - \mathbf{B}^{(j)} \| . \tag{4.3}$$

Define now the *Euclidean distance threshold* D. The threshold D represents the maximum Euclidean distance between the received RSSIs vector and the radio map entry (related to the actual position) which is considered "likely". The expurgation of $\mathcal{T}(n+1)$ proceeds therefore by eliminating the states whose Euclidean distance exceeds the threshold D. Thus,

$$\mathcal{T}^*(n+1) = \left\{ S_j(n+1) \subseteq \mathcal{T}(n+1) : d_j(n+1) \leq D \right\}.$$

4.1.3.4 Accumulate Metric Computation and Paths Storage

Consider a generic active state at time $(n+1)T_s$, $S_i(n+1)$. The incoming edges in $S_i(n+1)$ come clearly from any $S_j(n)$ such that $S_j(n) \in \mathcal{T}^*(n) \cap \mathcal{N}\left(S_i(n+1)\right)$.

The accumulated metric $\mu_i(n+1)$ of the active state $S_i(n+1)$ is computed as

$$\mu_i(n+1) = \min\left\{\mu_j(n) + w_{ji} d_i(n+1)\right\}, \tag{4.4}$$

$\forall j$ such that $S_j(n) \in \mathcal{T}^*(n) \cap \mathcal{N}(S_i(n+1))$, and where w_{ji} represents the transition weight between the states S_j and S_i. The term $w_{ji}d_i(n+1)$ is the *transition metric* between the states $S_j(n)$ and $S_i(n+1)$. The transition weights w_{ji} must be selected to take the user's mobility characteristics into account. Consider the example of a user with low mobility. In that case, w_{ii} (i.e., the weight associated to the permanence of the user in the ith location) should be much lower than any other transition weight. Together with the accumulate metric computation, the algorithm stores the survivor paths for each active state in the current time instant. It is now introduced the operator $\mathcal{P}^{(k)}(S_i(n))$ which returns, for a generic active state $S_i(n)$, the kth precursor on the survivor path which ends in $S_i(n)$. Therefore, $\mathcal{P}^{(k)}(S_i(n))$ will be the active state at the time instant $(n-k)$ belonging to the survivor path terminated by $S_i(n)$.

4.1.3.5 Position Estimation - The Proposed Algorithm

The user's location estimate at time nT_s is evaluated as follows. Assume $S_i(n)$ is the state belonging to $\mathcal{T}^*(n)$ with the lowest accumulated metric, $\mu_i(n)$. The decision about the user's position is delayed of m time intervals and corresponds to

$$\hat{S}(n) = \mathcal{P}^{(m)}(S_i(n)). \tag{4.5}$$

The choice of the parameter m must be adjusted taking into account the typical user's speed, the maximum latency that can be tolerated for a real-time service and the position estimation accuracy. On one hand, large values of m would in fact introduce large delays and may lead to obsolete position estimates for users characterized by a sufficiently high speed. On the other hand, a large value of m is an appealing solution for the algorithm performance improvement, since it permits to perform a decision, about the position of the user at time $(n-m)T_s$, taking into account the RSSIs samples on both the intervals $[0, (n-m)T_s]$ and $[(n-m+1)T_s, nT_s]$.

An improvement of the previously-presented algorithm can be obtained by a modified accumulated metric computation. In particular, through several tests it has been observed the presence of error propagation phenomena for several position estimations,

after a moderate/long fading event. In order to limit the effects due to fading correlation, our algorithm computes the accumulated metrics taking into account only the last h transition metrics of the associated paths.

A further improvement is introduced as follows. Consider the case where at the $(n-m)$th section of the trellis the two active states S_x and S_i are present. Their neighborhoods do not coincide, i.e., some states belonging to $\mathcal{N}(S_x)$ are not reachable from S_i, and viceversa. Assume $S_j(n)$ is the state with the lowest accumulated metric at time nT_s, and that $\mathcal{P}^{(m)}(S_j(n)) = S_x(n-m)$. Therefore, the estimated position at time nT_s will be $\hat{S}(n) = S_x(n-m)$. Assume now that the state with the lowest minimum distance at the time $(n+1)T_s$ is S_l and that $\mathcal{P}^{(m)}(S_l(n+1)) = S_i(n-m+1)$. Obviously, the estimated position at time $(n+1)T_s$ will be $\hat{S}(n+1) = S_i(n-m+1)$. Since $S_i \notin \mathcal{N}(S_x)$, the user transition $S_x(n-m) \rightarrow S_i(n-m+1)$ is inconsistent, i.e., it deals with locations that are not physically adjacent. To limit the effect of inconsistent hops, a further refinement to the proposed algorithm has been introduced. Assume $S_i(n)$ is the state with minimum accumulated metric in $\mathcal{T}^*(n)$ and consider its qth precursor $\mathcal{P}^{(q)}(S_i(n))$. The algorithm imposes that $\mathcal{P}^{(q)}(S_i(n))$ belongs to each survivor path at the nth section of the trellis. To obtain this results, each state $S_j(n)$ belonging to $\mathcal{T}^*(n)$ for which

$$\mathcal{P}^{(q)}(S_j(n)) \neq \mathcal{P}^{(q)}(S_i(n)),$$

with $m \leq q \leq h$ is subject to a further expurgation from $\mathcal{T}^*(n)$. The parameter q has therefore the role of setting a common precursor for the survivor paths in the last h time instances.

4.1.4 Algorithm Test

An ad-hoc software has been developed in order to compare the proposed algorithm both to the RADAR [70] and to the Bayesian filter described in [73]. The test has been organized as follows:

- During the "off-line phase" the radio map (consisting of 74 entries) has been constructed. For each point 4000 samples at the frequency of 10 Hz have been collected.

This phase is remarkably time-expensive, also because environmental changes and temporary changes could make the radio map obsolete or not accurate.

Note that the method of [76] has been used to mitigate the power attenuation caused by the presence of the user's body between NIC and AP, which could affect the measured values of the signal strength (especially if the body obstructs the LOS path connecting the laptop NIC and an AP). Therefore, during the off-line phase, for each point the laptop has been turned in the four cardinal directions every 100 seconds.

This "off-line phase" was necessary also for the Bayesian filter because, notwithstanding it does not use the radio map, the collected measures are very useful to calibrate the parameters inside both mobility model and perception model.

- Some routes has been defined in the indoor environment used for the tests. A set of measured signal strengths has been collected on the above-mentioned routes, walking at a speed ranging from 1.25 m/s to 1.78 m/s, which captures approximately 70% of people's constant speed walking [77]. The samples have been collected with a 10Hz rate, and have been stored into a file jointly with the real position of the user.

- Finally the Bayesian filter, RADAR algorithm and the new tracking algorithm have been implemented assuming as input the above-mentioned sample files. This procedure permits to compare the behavior of the three algorithms in the same conditions (i.e., same radio map and same on-line samples). The software produces as output the decision for both the solutions with the frequency of 1 Hz, i.e., every 10 samples.

4.1.5 Performance Evaluation through On-Field Measurements

The performance evaluation for the presented tracking algorithm has been performed taking into account two different scenarios. The first case (*dynamic scenario*, in the following) was referred to users characterized by high mobility, while a second case considered (almost) static users (*static scenario*, in the following). It must be now remarked that, for the purpose of this section, the attention has been focused on the

Table 4.1: Cumulative distribution function of the error (in meters) for Bayesian filter, RADAR and the new tracking algorithm based on-field dynamic measurements.

Error distance (meters)	CDF Bayesian filters	CDF RADAR	CDF tracking algorithm
0.0	25.1%	15.4%	22.9%
1.8	33.0%	21.4%	35.1%
2.6	38.0%	38.1%	56.0%
3.5	55.9%	56.3%	66.0%
4.4	63.5%	59.0%	69.6%
7.3	90.7%	94.9%	100.0%
21.0	100.0%	100.0%	100.0%

first case, where the tracking algorithm was expected to provide the largest performance gain respect to other positioning systems.

After some preliminary tests the tracking parameters have been determined, in order to reach a reasonably good trade-off between complexity and performance . The number of the initial states of the trellis, $L(0)$, is set to 10, and the threshold D to 50. The memory length h results to provide good performance when it is set to 5, while the delay (m) and the point of convergence (q) are set to 2 and 3 respectively. Finally is defined the cost of each type of transition. A unitary cost $(w_{ii} = 1)$ is associated to the permanence inside an office, as well as a transition between two states in the hallway. All the other costs are associated to $w_{ij} = 1.1$ for every $i \neq j$

The Cumulative Density Function (CDF) for the error distance (in meters) of Bayesian filter, RADAR and the new tracking algorithm for the on-field dynamic measurements, as shown in Tab. 4.1, reveals that the new tracking algorithm produces a better tracking of the user movement than both the RADAR and the Bayesian filter approaches. The error is within 2.6 meters in the 56% of locations for the new tracking algorithm, against only about 38% for both RADAR and Bayesian filter. The maximum error distance for our algorithm is 7.3 meters against 21 meters for both RADAR and Bayesian filter. Note that, although the Bayesian filter approach could be improved by a more accurate tuning of the models and parameters, these results confirm the expected performance improvement of our proposed algorithm in the dynamic scenario.

Table 4.2: Cumulative distribution function of the error (in meters) for Bayesian filter, RADAR and the new tracking algorithm based on-field static measurements.

Error distance (meters)	CDF Bayesian filter	CDF RADAR	CDF tracking algorithm
0.0	24.3%	21.9%	29.0%
1.8	29.0%	28.3%	34.6%
2.6	30.9%	51.1%	59.0%
3.5	36.7%	78.7%	80.0%
4.4	43.0%	81.5%	81.8%
8.7	94.0%	99.2%	100.0%
19.7	99.0%	100.0%	100.0%
35.4	100.0%	100.0%	100.0%

The second test provides a performance estimation for the static case, where the advantages introduced by the tracking algorithm are expected to be less significant. In Tab. 4.2, the CDF of the error distance (in meters) for Bayesian filter, the new tracking algorithm and the RADAR algorithm is reported, assuming static users. Although the parameters for the new tracking algorithm have been optimized for the dynamic case, it performs still slightly better than both RADAR and Bayesian filter. The error is within 2.6 meters in the 59% of locations for the new algorithm against 51.1% for RADAR and 30.9% for Bayesian filter. The maximum error distance for our algorithm is 8.7 meters to be compared with 19.7 m for RADAR and 35.4 m for the Bayesian filter.

Conclusions

In this thesis cooperative WSNs for localization have been analyzed. The localization problem has been studied and divided in three main issues: *synchronization*, *position estimation* and *tracking*, in order to have a deeper knowledge of how cooperation among nodes composing the network can help in improving the performances.

The first result has been the proposal of a new approach to derive the ultimate performance limit in time synchronization in WSNs based on the estimation theory [78]. The case of a connected network with a Gaussian MDT jitter has been investigated, obtaining a lower bound on the variance of the average synchronization error for a fully connected network. This theoretical limit can be used as a benchmark in comparing the performance of different time synchronizations algorithms. In particular, it has been shown that a better performance than single two-way message exchange is obtained as the number of nodes increases. This means that maximizing the network connectivity is beneficial for time synchronization.

Very interesting are also the results in the field of position estimate. The employing the unique localizability test recently proposed in [34] to study the problem of flip-ambiguity in network localization and motivate a novel formulation of the SDP localization algorithm that is robust to *flip-ambiguity* [67].

This marks a departure from the dominant approach to handle the problem found in current literature, which is to avoid, rather than resolve, flip-ambiguities. Although the work is preliminary, its results are inspiring and insightful, as it demonstrates the feasibility of concrete flip-ambiguous solutions with excellent performance.

One topic for future work is the extension of the result through the application of the strong localizability test so that better capture the effect of small perturbations on the node locations and their mutual distances. One further step under consideration is the formulation of a similar approach based on a simpler steepest-descent optimization

method, for example by translating the edge-bounding technique onto an equivalent weighing strategy [79].

The second main source of errors in position estimate, beyond the flipping problem, is the lack of connectivity between anchor and target nodes. To analyze and mitigate this problem, in this thesis two localization algorithms to obtain accurate performances in multihop networks only operating with the multihop target-to-anchor distance estimates have been proposed [65, 80]. The result permits an huge reduction of the complexity with respect to the classic algorithms, in which every node needs to collect location and distance estimates from all its neighbors. This is possible because the new approach is robust and resilient to the target-to-anchor distances estimate error, that is proportional to the number of hops, instead of performing message-passing as done by other schemes.

The gain in the algorithmic complexity reduction is a reduction of the network traffic, with evident benefits regarding the energy consumption and packet collisions reduction.

Another advantage of this new approach to multihop localization is the improvement of the network security. In fact it is sufficient that only anchors send their location all over the network whereas targets can compute their position estimate without sharing it with their neighbors.

As last issue, a tracking algorithm has been proposed, implemented and tested [81].

This new low-complexity solution is for the position tracking of users in a WLAN-based indoor positioning system. The performance of this new algorithm has been compared with the well-established RADAR algorithm [70] and Bayesian Filters [73]. The on-field results of these tests proved that our tracking algorithm has the best performance and it can be therefore used with good trade off between accuracy and complexity for indoor localization.

Appendix A

Basic Recall of Semidefinite Programming

The network localization problem stated in (3.8), is a non-convex optimization problem, difficult to solve. Semidefinite Programming (SDP) technique can be a powerful tool to solve, after some algebraic manipulations, problems as the one considered.

The basic idea behind SDP technique, an extension of Linear Programming (LP), is to convert non-convex quadratic distance constraints, as in (3.5) and (3.6), into linear constraints by introducing a relaxation to remove the quadratic term in the formulation. Non-negativity constraints on vector variables are replaced by positive semidefiniteness constraints on matrix variables.

A.0.6 Linear Programming

It is now useful to introduce the standard formulation of LP, a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints.

$$\begin{array}{rl} \text{maximize} & \operatorname{tr}\left(\mathbf{C}^T\mathbf{X}\right) \\ \text{subject to} & \operatorname{tr}\left(\mathbf{A}_i^T\mathbf{X}\right) = \mathbf{b}, \end{array} \tag{A.0.1}$$

where $\mathbf{X}$ represents the n-by-n matrix (to be determined), $\mathbf{b}$ is a vectors of (known) coefficients and $\mathbf{C}$ and $\mathbf{A}_i$'s are (known) matrices of coefficients. The expression to be *maximized* or *minimized* is called the objective function ($\operatorname{tr}\left(\mathbf{C}^T\mathbf{X}\right)$ in this case). The equations $\operatorname{tr}\left(\mathbf{A}_i^T\mathbf{X}\right) = \mathbf{b}$ are the constraints which specify a *convex set* over which the objective function is to be optimized.

A.0.7 Semidefinite Technique

SDP is a subfield of convex optimization concerned with the optimization of a linear objective function over the intersection of the cone of positive semidefinite matrices with an affine space. Semidefinite relaxation of (A.0.1) is easily obtaining imposing

$$\mathbf{X} \succeq 0, \tag{A.0.2}$$

i.e. imposing that matrix $\mathbf{X}$ is positive semidefinite. Now the problem, in its standard form, becomes:

$$\begin{array}{rl} \text{maximize} & \operatorname{tr}\big(\mathbf{C}^T\mathbf{X}\big) \\ \text{subject to} & \operatorname{tr}\big(\mathbf{A}_i^T\mathbf{X}\big) = \mathbf{b}, \\ & \mathbf{X} \succeq 0. \end{array} \tag{A.0.3}$$

All linear programs can be expressed as SDPs.

In particular, the network localization problem stated in (3.8) does not require to maximize a particular cost function but, instead, it is only a "*feasibility problem*". It is sufficient only to satisfy all the constraints. For this reason term $\operatorname{tr}\big(\mathbf{C}^T\mathbf{X}\big)$ is not more important and it can be replace by 0.

Finally, the feasibility semidefinite problem becomes:

$$\begin{array}{rl} \text{maximize} & 0 \\ \text{subject to} & \operatorname{tr}\big(\mathbf{A}_i^T\mathbf{X}\big) = \mathbf{b}, \\ & \mathbf{X} \succeq 0. \end{array} \tag{A.0.4}$$

Bibliography

[1] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, in *Wireless Sensor and Actuator Networks: enabling technologies, information processing and protocol design*. Elsevier, 2008. vii, 2, 5, 7, 17

[2] M. Lucchi, "Cooperative communication and distributed detection in wireless sensor networks," Ph.D. dissertation, II Faculty of Engineering - Alma Mater Studiorum University of Bologna, 2006-2007. 1

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002. 1

[4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008. 1

[5] T. Gao, D. Greenspan, M. Welsh, R. R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *in In Proceedings of the 3rd International Conference on Information Communication Technologies in Health (ICICTH05*, 2005. 1

[6] K. Lorincz, D. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: challenges and opportunities," *IEEE Pervasive Computing for First Response (Special issue)*, October-December 2004. 1

[7] G. Simon, M. Marti, kos Ldeczi, G. Balogh, B. Kusy, A. Ndas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *Second International Conference on Embedded Networked Sensor system (Sensys)*. Baltimore, MD: ACM Press, 2004, pp. 1–12. 1

[8] J. Yick, B. Mukherjee, and D. Ghosal, "Analysis of a prediction-based mobility adaptive tracking algorithm," in *Second International Conference in Broadband Networks (Broadnets)*, Boston, 2005. 1

[9] M. Castillo-Effen, D. H. Quintela, R. Jordan, W. Westhoff, and W. Moreno, "Wireless sensor networks for flash-flood alerting," in *5th IEEE International Caracas Conference on Devices, Circuits and Systems*, Caracas, November 2004, pp. 142–146. 1

[10] G. Werner-Allen, M. R. K. Lorincz, O. Marcillo, J. Jhonson, J. Lees, and M. Walsh, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, March-Aprile 2006. 1

[11] C. E. Perkins, *Ad Hoc Networks*. Reading, MA: Addison-Wesley, 2000. 1

[12] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless Sensor Network Localization Techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007. 3

[13] A. A. Kannan, B. Fidan, and G. Mao, "Derivation of flip ambiguity probabilities to facilitate robust sensor network localization," in *Proceedings of the 2009 IEEE conference on Wireless Communications and Networking Conference (WCNC'09:)*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 2027–2032. 3

[14] S. Bhatti and J. Xu, "Survey of target tracking protocols using wireless sensor network," *Wireless and Mobile Communications, International Conference on*, vol. 0, pp. 110–115, 2009. 4, 51

[15] E. Kaasinen, "User needs for location-aware mobile services," *Personal Ubiquitous Comput.*, vol. 7, no. 1, pp. 70–79, 2003. 4

[16] D. L. Mills, "Internet time synchronization: The network time protocol," in *Zhonghua Yang and T. Anthony Marsland (Eds.), Global States and Time in Distributed Systems, IEEE Computer Society Press*, 1994. 5

[17] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks," in *Parallel and Distributed Processing Symposium., Proceedings 15th International*, 2001, pp. 1965–970. 5

[18] J. Elson and K. Romer, "Wireless sensor networks: a new regime for time synchronization," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 149–154, 2003. 5

[19] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing synch protocol for sensor networks," in *ACM SenSys*, Los Angeles, CA, Nov. 2003. 5

[20] R. Fan and N. Lynch, "Gradient clock synchronization," in *PODC04, Proceedings of the fifth annual ACM symposium on Principles of distributed computing*, St. Johns, Newfoundland, Canada, Jul. 2004. 5

[21] J. Lundelius-Welch and N. A. Lynch, "An upper and lower bound for clock synchronization." *Information and Control*, vol. 62, pp. 190–204, 1984. 6

[22] S. Biaz and J. L. Welch, "Closed form bounds for clock synchronization under simple uncertainty assumptions," *Information Processing Letters*, vol. 80, no. 3, pp. 151–157, 2001. 6, 15

[23] R. Fan and N. Lynch, "Gradient clock synchronization," in *PODC04, Proceedings of the fifth annual ACM symposium on Principles of distributed computing*, St. Johns, Newfoundland, Canada, Jul. 2004. 6, 15

[24] T. K. Srikanth and S. Toueg, "Optimal clock synchronization," *J. ACM*, vol. 34, no. 3, pp. 626–645, 1987. 6, 15

[25] G. Kirchhoff, "Uber die auflsung der gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer strme gefhrt wird," *Ann. Phys. Chem*, no. 72, pp. 497–508, 1847. 10

[26] S. Skiena, *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Addison-Wesley, Ed., 1990. 10

[27] L. Chua, C. Desoer, and E. Kuh., *Linear and Nonlinear Circuits.,*. McGraw-Hill., 1987. 10, 11

[28] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, 1st ed. New York, NY 10158-0012: John Wiley & Sons, Inc., 1968. 12

[29] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer*, vol. 34, pp. 57–66, 2001. 17

[30] J. Bachrach and C. Taylor, *Handbook of Sensor Networks: Algorithms and Architectures*. Wiley, 2005, ch. 1 Localization in Sensor Networks. 17

[31] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. J. Teller, "Anchor-Free Distributed Localization in Sensor Networks." ACM, 2003, pp. 340–341. 18

[32] N.-H. Z. Leung and K.-C. Toh, "An SDP-based divide-and-conquer Algorithm for large Scale noisy Anchor-free Graph realization," *SIAM Journal on Scientific Computing*, vol. 31, pp. 4351–4372, 2009. 18

[33] A. Kannan, B. Fidan, G. Mao, and B. Anderson, "Analysis of Flip Ambiguities in Distributed Network Localization," in *Information, Decision and Control, 2007. IDC '07*, Feb. 2007, pp. 193–198. 18

[34] A. M. So and Y. Ye, "Theory of Semidefinite Programming for Sensor Network Localization," in *Proc. IEEE Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Vancouver, British Columbia Canada, January 2005, pp. 405–414. 19, 20, 22, 40, 63

[35] P. Biswas, T.-C. Liang, K.-C. Toh, and T.-C. Wang, "Semidefinite Programming Based Algorithms for Sensor Network Localization with noisy Distance Measurements," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 2, pp. 188–220, May 2006. 20, 29, 31

[36] P. Biswas and Y. Ye, "A Distributed Method for Solving Semidefinite Programs Arising from Ad Hoc Wireless Sensor Network Localization," Tech. Rep., 2003. 23

[37] A. Kannan, B. Fidan, and G. Mao, "Robust Distributed Sensor Network Localization based on Analysis of Flip Ambiguities," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM 2008)*, December 2008, pp. 1–6. 24

[38] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," in *IEEE Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys04)*, 2004, pp. 50–61. 24

[39] L. E. Miller, "Distribution of Link Distances in a Wireless Network," *Journal of Research of the National Institute of Standards and Technology*, vol. 106, pp. 401–412, 2001. 24

[40] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959. 29

[41] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, 2nd ed. Chapman & Hall/CRC, 2000. 31, 37

[42] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win, "Ranging with Ultra-wide Bandwidth Signals in Multipath Environments," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 404 –426, feb. 2009. 33

[43] F. Gustafsson and F. Gunnarsson, "Mobile Positioning using Wireless Networks: Possibilities and Fundamental Limitations based on Available Wireless Network Measurements," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 41–53, July 2005. 33

[44] U. Ferner, H. Wymeersch, and M. Win, "Cooperative anchor-less localization for Large Dynamic Networks," in *Ultra-Wideband, 2008. ICUWB 2008. IEEE International Conference on*, vol. 2, Sept. 2008, pp. 181–185. 33

[45] S. Y. Wong, J. G. Lim, S. Rao, and W. Seah, "Multihop Localization with Density and Path Length Awareness in Non-Uniform Wireless Sensor Networks," vol. 4, May-1 June 2005, pp. 2551–2555 Vol. 4. 33, 34

[46] B. Awerbuch and R. Gallager, "A New Distributed Algorithm to find Breadth First Search Trees," *Information Theory, IEEE Transactions on*, vol. 33, no. 3, pp. 315–322, May 1987. 33, 34, 36

[47] D. Niculescu and B. Nath, "Dv based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003. 33, 47

[48] ——, "Ad hoc positioning system (aps) using aoa," vol. 3, march-3 april 2003, pp. 1734 – 1743 vol.3. 33

[49] ——, "Ad hoc positioning system (aps)," vol. 5, 2001, pp. 2926 –2931 vol.5. 33

[50] B. Parkinson and J. Spilker, *Global Positioning System: Theroy and Application*. American Istitute of Aeronautics and Astronautics, 1996. 33, 47

[51] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," in *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2002, pp. 317–327. 34, 47

[52] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2001, pp. 166–179. 34

[53] A. Savvides, H. Park, and M. B. Srivastava, "The Bits and Flops of the n-Hop Multilateration Primitive for Node Localization Problems," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 112–121. 34

[54] R. Parker and S. Valaee, "A Robust Min-Max Localization," sept. 2006, pp. 1000 –1005. 34

[55] N. D. Heiko Will and J. Schiller, "Distance-Based Distributed Multihop Localization in Mobile Wireless Sensor Networks," in *8th GI/ITG KuVS Fachgesprch Drahtlose Sensornetze (FGSN'09)*, 2009. 34

[56] Z. Y. Yawen Wei and Y. Guan, "COTA: A Robust Multi-hop Localization Scheme in Wireless Sensor Networks," in *DCOSS*, 2006, pp. 338–355. 34

[57] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient Geographic Routing in Multihop Wireless Networks," in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2005, pp. 230–241. 34

[58] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, 2004. 34

[59] J. de Leeuw and P. Mair, "Multidimensional Scaling Using Majorization: SMACOF in R," 2008. [Online]. Available: http://repositories.cdlib.org/uclastat/papers/2008010903 38, 47

[60] L. Guttman, "A general nonmetric technique for finding the smallest coordinate space for a configuration of points," *Psychometrika*, vol. 33, no. 4, pp. 469–506, 1968. 39

[61] G. Destino and G. Abreu, "Reformulating the Least-Square Source Localization Problem with Contracted Distances," *Asilomar Conference on Signals, Systems, and Computers*, 2009. 40, 41

[62] K. Langendoen and N. Reijers, "Distributed Localization in Wireless Sensor Networks: a Quantitative Comparison," *Comput. Netw.*, vol. 43, no. 4, pp. 499–518, 2003. 45

[63] K. Whitehouse and D. Culler, "A Robustness Analysis of Multi-Hop Ranging-based Localization Approximations," in *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*. New York, NY, USA: ACM, 2006, pp. 317–325. 46

[64] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, "The Effects of Ranging Noise on Multi-hop Localization: An Empirical Study," in *Fourth International Conference on Information Processing in Sensor Networks (IPSN 2005)*, April 2005, pp. 73–80. 46

[65] S. Severi, G. Abreu, G. Destino, and D. Dardari, "Efficient and Accurate Localization in Multihop Networks," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, November 2009. 46, 64

[66] X. Li, "Collaborative Localization With Received-Signal Strength in Wireless Sensor Networks," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 6, pp. 3807 –3817, nov. 2007. 47

[67] S. Severi, G. Abreu, G. Destino, and D. Dardari, "Understanding and Solving Flip-Ambiguity in Network Localization via Semidefinite Programming," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM 2009)*, December 2009. 48, 63

[68] S. Yi, J. Yi, and H. Cha, "Tracking multiple mobile objects using ieee 802.15.4-based ultrasonic sensor devices," in *Proceedings of the 2007 ACM symposium on Applied computing (SAC '07)*. New York, NY, USA: ACM, 2007, pp. 1638–1639. 51

[69] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Computer*, vol. 8, no. 34, pp. 57–66, August 2001. 52, 53

[70] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," *IEEE Infocom 2000, Tel Aviv, Israel*, March 2000. 52, 53, 59, 64

[71] M. A. Youssef, A. Agrawala, and A. U. Shankar, "WLAN Location Determination via Clustering and Probability Distributions," *IEEE PerCom 2003, Fort Worth, Texas*, pp. 143–152, March 2003. 52

[72] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for local estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24–23, 2003. 52

[73] T. Pavani, G. Costa, M. Mazzotti, A. Conti, and D. Dardari, "Experimental results in indoor localization techniques through wireless sensor network," in *Proc. IEEE Vehicular Tech. Conf. (VTC 2006-Spring)*, vol. 2, Melbourne, Australia, May 2006, pp. 663–667. 52, 59, 64

[74] R. M. Fano, "A Heuristic Discussion of Probabilistic Decoding," *IEEE Transactions on Information Theory*, vol. 9, pp. 64–74, April 1963. 52

[75] WRAPI 2.0. [Online]. Available: http://sysnet.ucsd.edu/pawn/wrapi/ 54

[76] R. Valenzuela, O. Landron, and D. Jacobs, "Estimating Local Mean Signal Strength of Indoor Multipath Propagation," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 1, 2 1997. 60

[77] R. L. Knoblauch, M. T. Pietrucha, and M. Nitzburg, "Field Studies of Pedestrian Walking Speed and Start-Up Time," *Transportation Research Record*, no. 1538, 1996. 60

[78] S. Severi and D. Davide, "Performance Limits of Time Synchronization in Wireless Sensor Networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, May 2008, pp. 2124 –2128. 63

[79] G. Destino, G. Abreu, "Weighing Strategy for Network Localization under Scarce Ranging Information," *IEEE Transactions on Wireless Communications*, 2009. 64

[80] S. Severi, G. Abreu, and D. Dardari, "A Quantitative Comparison on Multihop Algorithms," in *Proc. Workshop on Positioning, Navigation and Communication (WPNC'10)*, March 2010. 64

[81] S. Severi, G. Liva, M. Chiani, and D. Davide, "A New Low-Complexity User Tracking Algorithm for WLAN-Based Positioning Systems," in *Mobile and Wireless Communications Summit, 2007. 16th IST*, July 2007, pp. 1 –5. 64