

TECHNICAL UNIVERSITY OF CRETE
ELECTRONIC AND COMPUTER ENGINEERING DEPARTMENT
TELECOMMUNICATIONS DIVISION



Cooperative Network Sensor Localization Under Bandwidth Constraints

by

Fasarakis - Hilliard Nikos

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DIPLOMA DEGREE OF

ELECTRONIC AND COMPUTER ENGINEERING

September 2012

THESIS COMMITTEE

Assistant Professor Aggelos Bletsas, *Thesis Supervisor*

Assistant Professor George N. Karystinos

Professor Athanasios P. Liavas

Abstract

The proliferation of low-cost wireless networking options, the advance of sensor network technologies and the inherent importance of location awareness in many scenarios, from item tracking and monitoring to social networks, has generated an immense interest on localization techniques over the last years. This thesis considers the general problem of estimating the coordinates of networked sensors under constraints. Localization is considered "cooperative" in the sense that agents of unknown location cooperate with neighboring agents in order to estimate their position in the network, and "constrained" in the sense that energy, bandwidth or cost limitations may exist.

Specific state-of-the-art localization techniques found in the literature are presented. Their performance is quantified on the basis of offered estimation accuracy and bandwidth requirements, in the presence of Gaussian or non-Gaussian, unimodal ranging error noise. This thesis offers concrete bandwidth-friendly versions of SPAWN, i.e., factor-graph based localization, shown to significantly reduce the total size of exchanged messages, one-to two- orders of magnitude. Furthermore, a computationally efficient flavor is proposed that further reduces the mean squared error (MSE) compared to prior-art. Cramer-Rao bound calculation corroborates the efficiency of the proposed algorithms.

Acknowledgements

.....

Table of Contents

Table of Contents	4
List of Figures	7
List of Tables	11
List of Abbreviations	12
1 Introduction	13
1.1 Wireless Sensor Networks	13
1.1.1 Motivating Application : Under-water Navigation	15
1.2 Formal Problem Statement	16
1.2.1 Measurements: Pair-Wise & Internal	17
1.2.2 Absolute & Relative Positioning	18
1.3 Thesis Contributions	18
1.4 Outline of Thesis	19
2 Ranging Measurements - Models	20
2.1 Received Signal Strength (RSS)	21
2.1.1 Range estimation via RSS	22
2.2 Time of Arrival (TOA)	22
2.2.1 Statistical model for TOA	23
2.2.2 Time-difference of Arrival (TDOA)	24
2.3 Angle of Arrival (AOA)	24
2.3.1 Statistical model for AOA	25
2.4 Connectivity	26
2.5 Chapter 2 Conclusions	27

3	Cooperative Localization Algorithms	28
3.1	Overview of Cooperative Localization Algorithms	29
3.1.1	Centralized Algorithms	29
3.1.2	Distributed Algorithms	30
3.1.3	Centralized / Distributed Comparison	31
3.2	Distributed Weighted Multi-dimensional Scaling (dwMDS)	31
3.2.1	The dwMDS two-stage algorithm	33
3.2.2	Performance of the dwMDS method	38
3.3	Distributed Localization in Random Environments (DLRE)	41
3.3.1	The DILOC algorithm	43
3.3.2	Extension of DILOC to DLRE	45
3.3.3	Performance of the DILOC & DLRE algorithms	48
3.4	GPS-Free Node Localization	52
3.4.1	Two stage positioning algorithm	53
3.4.2	Performance of GPS-Free Node localization	57
3.5	Chapter 3 Conclusions	58
4	The Sum Product Algorithm over Wireless Networks	60
4.1	An introduction to Factor Graphs (FGs)	60
4.1.1	The Sum Product Algorithm (SPA)	61
4.1.2	Factor Graphs and Sequential Estimation	62
4.2	Sum Product Algorithm over Wireless Networks (SPAWN)	63
4.2.1	Modeling Assumptions	63
4.2.2	Creation of Network Factor Graph	65
4.2.3	Prediction and Correction Operations	68
4.3	Performance of SPAWN	70
5	Parameterized Sum Product Algorithm over Wireless Networks (P-SPAWN)	73
5.1	Message representation	74
5.2	Classification of beliefs in a noiseless environment	77
5.3	The P-SPAWN algorithm	80
5.4	Performance of P-SPAWN algorithm	85

Table of Contents

6

6 Conclusion and Future Work	94
Bibliography	95

List of Figures

1.1	The benefit of using cooperative localization: using only distance estimates with respect to the anchors (node 1,2 and 3), agent nodes 4 and 5 are unable to determine their respective positions without ambiguity. By allowing communication between agent nodes 4 and 5 (as depicted by the red arrow), they can cooperate to unambiguously determine their positions . . .	15
3.1	The expected value of the RSS-based estimate of range given that two devices are neighbors and the ideal unbiased performance	36
3.2	Network topology for 2-dimentional localization. Observe that agent are connected only with two anchor and need to cooperate in order to localize.	38
3.3	Mean squared error (MSE) as a function of ranging noise variance σ_r^2 for $2D$ localization with the dwMDS method. Operations are performed with and without assuming prior knowledge regarding the agents locations	39
3.4	Bandwidth (BW) performance as a function of MSE for $2D$ localization with the dwMDS method. Operations are performed with and without assuming prior knowledge regarding the agents locations	39
3.5	Mean squared error (MSE) as a function of ranging noise variance σ_r^2 for $3D$ localization with the dwMDS method. Operations are performed with and without assuming prior knowledge regarding the agents locations	40

3.6	Bandwidth (BW) performance as a function of MSE for 3D localization with the dwMDS method. Operations are performed with and without assuming prior knowledge regarding the agents locations	40
3.7	A three dimensional convex hull inclusion test: Sensor l is depicted by "o" while neighbors are depicted by " ∇ ". (a) $A_k = A_{k \cup \{l\}}$ and (b) $A_k > A_{k \cup \{l\}}$	44
3.8	Toy 2D topology of 49 nodes to be localized through DILOC and DLRE. Note that all agents, depicted by blue "o"s, lie in the convex hull of the anchors, depicted by the red triangles.	48
3.9	Convergence of DILOC estimates to exact sensor locations under a noiseless channel for the topology of figure (3.8). While convergence is guaranteed, it is also slow.	49
3.10	Convergence of DLRE estimates to exact sensor locations under a noisy channel for the topology of figure (3.8). Impressive performance is accompanied by heavy bandwidth requirements expressed as the total amount of real numbers exchanged.	50
3.11	Three-dimensional topology for localization through DLRE. Agent nodes lie in the convex hull formed by the anchors.	51
3.12	Mean square error as a function of the number of iterations for localization through DLRE for the topology of figure (3.11). The conclusions regarding two-dimensional localization can be extended to three-dimensional localization.	51
3.13	Typical movement of two neighbor nodes during a round for relative localization.	53
3.14	Average positioning error as a function of distance and angle measurements for GPS-Free relative node localization.	57
3.15	Percent of nodes unlocalized as a function of distance and angle measurements for GPS-Free relative node localization. Observe that given the right density (collective movement) performance is impressive.	58

4.1	The factor graph of the factorization of factor $p(\mathbf{z}_{\text{rel}}^{(t)} \mathbf{x}^{(t)})$ for the network of example [1.1].	66
4.2	A graphical illustration of a FG corresponding to the expression (4.17) Observe that the message flow, from past to present indicated by the direction of arrows, accounts for temporal constraints.	67
4.3	Network topology for two dimensional localization through SPAWN. 71	
4.4	MSE as a function of the noise variance for SPAWN	71
4.5	Total size of exchanged messages for small and large ranging error noise variance for SPAWN algorithm. Observe the prohibiting order of magnitude.	72
4.6	Cramer-Rao lower bound (CRB) for the topology of figure [4.3]. 72	
5.1	An agent measuring noisy distance from an anchor can be located anywhere on a grommet centered at the anchors location. 74	
5.2	Communication with three anchors leads to a unimodal belief on the $2D$ plane.	75
5.3	The area of received factor graph messages quantifies uncertainty. 76	
5.4	The resulting belief is not modeled adequately neither as a single nor a pair of clusters.	84
5.5	Test topology and node connectivity: three anchors are placed at $[0 \ 0 \ 0]^T, [20 \ 35 \ 0]^T, [50 \ 50 \ 0]^T$ and two agents at $[15 \ 20 \ 20]^T, [45 \ 20 \ 30]^T$, respectively. Notice that each agent can communicate with only two anchors and another agent. For 2D experiments, nodes are assumed on a common plane according to their x and y coordinates.	89
5.6	Mean squared error (MSE) as a function of ranging noise variance σ_r^2 for 2D localization. All algorithms operate without agents location prior knowledge.. . . .	89

-
- 5.7 K-SPAWN and P-SPAWN offer the best tradeoff between MSE and size of exchanged messages (expressed in total real numbers) and reduces total number of real numbers exchanged by two to three orders of magnitude, compared to classic SPAWN. For large ranging error variance, all algorithms achieve similar MSE with vastly different message size requirements. 90
- 5.8 Back-to-back (BTB) schemes can outperform state-of-the-art cooperative localization. 90
- 5.9 BTB-K- and BTB-P-SPAWN require one-of-order of magnitude smaller exchanged messages (in total real numbers) compared to BTB-SPAWN. The latter achieves the smallest MSE. All BTB algorithms achieve similar MSE in relatively large ranging error variance. 91
- 5.10 Approximating the 2D plane with a grid of area δ^2 regions is necessary for numerical calculations. Parameter $\delta = 5$ m offers better MSE results than $\delta = 4$ m, since the latter value, even though smaller, is not the greatest common divisor of all anchor's non-zero coordinates. 91
- 5.11 MSE across all agents as a function of ranging noise variance σ_r^2 and depth/height measurement noise variance $\sigma_z^2 = 1$ for 3D localization. 92
- 5.12 BTB-SPAWN further reduces MSE compared to classic SPAWN, while reducing size of exchanged messages one order of magnitude. K-SPAWN and P-SPAWN can further reduce the required bandwidth by one or two orders of magnitude compared to BTB-SPAWN, at the expense of MSE. 92
- 5.13 The Cramer-Rao bound (CRB) for the 3D long-baseline (LBL) network setup of this work, including ranging noise variance σ_r^2 and depth/height noise variance σ_z^2 . CRB for 2D networks is also depicted as a function of σ_r^2 93

List of Tables

List of Abbreviations

WSN	Wireless Sensor Network
GPS	Global Positioning System
AUV	Autonomous Underwater Vehicle
DVL	Doppler Velocity Logger
INS	Internal Navigation System
BtB	Back to Back
TOA	Time Of Arrival
TDOA	Time Difference Of Arrival
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
AOA	Angle Of Arrival
RF	Radio Frequency
ML	Maximum Likelihood
LOS	Line-of-Sight
DLOS	Direct Line-of-Sight
NLOS	Non Line-of-Sight
UWB	Ultra Wideband
CRB	Cramer-Rao Bound
LBL	Long-Baseline Localization
MDS	Multi-Dimensional Scaling
dwMDS	Distributed Weighted Multi-Dimensional Scaling
SPAWN	Sum Product Algorithm over Wireless Networks
PSPAWN	Parameterized Sum Product Algorithm over Wireless Networks
PF	Particle Filtering
SMACOF	Scaling by MAjorizing a COmplicated Function
CMD	Cayley-Menger Determinant

Chapter 1

Introduction

1.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions and to cooperatively pass their data through the network to a main location. Dramatic advances in technology have made possible the use of large scale wireless sensor networks for a variety of monitoring and control applications. Possible applications of sensor networks are of interest to the most diverse fields. For example, precision agriculture will reduce cost and environmental impact by watering exactly where necessary [8]. Environmental monitoring networks will sense air, water and soil quality and identify the source of pollutants in real time [30]. Animal tracking and monitoring is made possible with dynamic sensor networks, providing insight into their behavior and moves [25]. Even medical research and health-care will greatly benefit from sensor networks: vital sign monitoring and accident recognition are the most natural applications [21]. Countless applications have been enabled by the promise of inexpensive reliable wireless sensor networks, as described in review articles in [2, 14, 35].

Automated network self-configuration is a critical demand for the effective use of today's wireless sensor networks; their scale precludes requiring a human administrator to set up each node in the network. Automatic location estimation of the sensors in these networks is a subset of the self-configuration problem which is of crucial importance. The overwhelming reason is that in order for the data of a sensor to be meaningful, its location must be known. If an application is set up to respond locally to changes in sensor data it must know where those changes are occurring.

The Global Positioning System (GPS), applied in numerous military and

civilian applications has already provided a positioning solution for users world wide. However, the large scale of today's wireless sensor networks renders equipping each sensor with a GPS receiver impractical, energy-wise and cost-wise. Problems such as increased cost, rapid battery depletion and a severe lack of robustness in indoor environments or hash terrain, prohibits extended use of GPS for localization purposes in large wireless sensor networks.

As a result, in many conventional localization systems a limited number of reference nodes, referred to as anchors, with prior knowledge of their location (e.g., equipped with GPS), are deployed. These reference nodes interact with devices of unknown location, hereby referred to as agents, who exploit information such as the coordinates of anchors and the distance measurements to them, in order to infer their own position. The performance of positioning depends heavily on the network connectivity. In the general case, each agent requires at least three anchors to determine it's own position on the two-dimensional plane (Fig. 1.1)). Localization is ensured given communication with a sufficient number of reference nodes, scattered throughout the network.

Performance in both accuracy and coverage can be significantly improved by introducing cooperation between connected agents. Cooperative localization overcomes the reliance on the coverage of anchor nodes (referred to as non-cooperative positioning) by allowing connectivity between pairs of agents within communication range. Fig. 1.1) shows an example where unambiguous localization could not be achieved without the improved coverage cooperation provides. Cooperation and the additional information and coverage it provides is a powerful tool and this work focuses exclusively on cooperative localization techniques.

The focus of this thesis is on the self-configuration problem of estimating a sensor's spatial location using distributed schemes and cooperation between agents under communication constraints. As mentioned, estimating a sensor's physical coordinates is intuitively important: data reported from a sensor needs to be accompanied with an indication of where in space that data was recorded. However, available sensor data is constrained by

the bandwidth and energy limitations imposed by the network. Large quantities of aggregate information exchanged solely for localization purposes in addition to "meaningful" sensed data could deplete batteries and reduce the lifetime of an energy-constrained network, or could be completely infeasible in cases such as underwater systems, due to narrow-band acoustic modems. The focus of this work is not only on localization accuracy, but also on the very important trade-off between localization accuracy and communication bandwidth required for location estimation in motion-less or mobile wireless networks.

1.1.1 Motivating Application : Under-water Navigation

The absence of Global Positioning System signals underwater makes navigation for Autonomous Underwater Vehicles (AUVs) a difficult challenge. Without an external reference in the form of acoustic beacons at known locations, the AUV has to rely on proprioceptive information obtained through

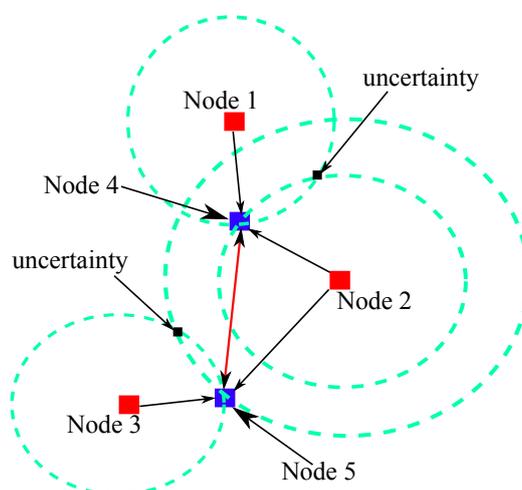


Figure 1.1: **The benefit of using cooperative localization: using only distance estimates with respect to the anchors (node 1,2 and 3), agent nodes 4 and 5 are unable to determine their respective positions without ambiguity. By allowing communication between agent nodes 4 and 5 (as depicted by the red arrow), they can cooperate to unambiguously determine their positions**

a compass, a Doppler Velocity Logger (DVL) or an Internal Navigation System (INS). Independent of the quality of the sensors used, errors in the position estimates introduced based on the aforementioned information grows without a bound [5]. Furthermore, while a position estimate can be obtained through GPS if the AUV surfaces, it is undesirable and even infeasible in some cases (e.g., under ice).

Cooperative localization, as described in the previous section, could be utilized to create a fully mobile network of AUVs that perform acoustic ranging and exchange data with one another to achieve cooperative positioning. The importance of bandwidth conservation is crystal clear in a scenario such as the above. Due to narrow-band state-of-the-art acoustic modems, hard constraints are imposed on the total number of bits that can be exchanged in a limited period of time for localization purposes. Smart cooperative localization techniques with respect to energy and/or bandwidth constraints, as proposed in this thesis, are a key enabling technology, providing such AUVs with the means to achieve accurate position estimations for extended duration missions over large areas without relying on GPS or expensive sophisticated DVL and INS sensors.

1.2 Formal Problem Statement

Before going into detail, it is useful to formally state the problem of cooperative sensor localization. Throughout this thesis a network of $\mathcal{N} + \mathcal{N}_A$ nodes is assumed, with \mathcal{N} *agents* of unknown location and \mathcal{N}_A *anchors* with a priori known coordinates across the entire network. The objective is to estimate the coordinates of the agents, given a priori the known coordinates of the anchors and at least one of a variety of location measurements. In other words, for a two-dimensional localization problem, a total of $2\mathcal{N}$ unknown location parameters must be estimated, $\mathbf{p} = [\mathbf{p}_x, \mathbf{p}_y]$, where

$$\mathbf{p}_x = [x_1, \dots, x_{\mathcal{N}}], \quad \mathbf{p}_y = [y_1, \dots, y_{\mathcal{N}}] \quad (1.1)$$

given the anchors known reference coordinates

$$[x_{N+1}, \dots, x_{N_A}, y_{N+1}, \dots, y_{N_A}]$$

and cooperation between neighboring agents. The set, $H(i)$ stands for the set that includes all neighbors of node i , i.e. all nodes with which node i can exchange information messages or perform ranging measurements. Furthermore, the location of sensor i is also referred to as \mathbf{x}_i , where $\mathbf{x}_i = [x_i, y_i]^T$ for two-dimensional localization. We are interested in both two-dimensional (2D) as well as three-dimensional (3D) localization and \mathcal{D} denotes the corresponding dimensionality of the localization problem.

1.2.1 Measurements: Pair-Wise & Internal

Measurements can be either pair-wise or internal. Pair-wise measurements, $\mathbf{z}_{i,j}$ between communicating sensors i, j , could be any physical reading that indicates distance or relative position, such as time-of-arrival (TOA) requiring synchronized time in one-way ranging measurements, received-signal-strength (RSS), angle-of-arrival (AOA) or even connectivity indicators. Internal measurements such as pedometers, compasses, altitude meters etc. are also considered available. Both type of measurements can contain, directly (e.g. TOA) or indirectly (e.g. RSS), information about the relative position of the sensors in the network.

Pair-wise measurements are allowed only between neighboring nodes. That is, nodes i, j can perform pair-wise measurements if and only if $i \in H(j)$ and $j \in H(i)$. Various measurement methods, such as the ones mentioned above, are presented with greater detail in Chapter 2. It is noteworthy that these pair-wise measurements are adversely affected by the physical environment and due to the fact that the exact environment in which the measurements are performed cannot be known a priori, environmental effects are considered random. Chapter 2 presents in detail each type of pair-wise measurement, with additional information noted as needed.

1.2.2 Absolute & Relative Positioning

While relying on set of anchor nodes with a priori known coordinates is quite common, it is not a necessity. In many applications reference nodes may not exist or may not be of interest and when such is the case, we refer to *relative localization*. A reference point does not exist so an arbitrary coordinate system can be chosen. For example in swarming techniques [1], only distances and angles between nodes are of interest and thus a absolute coordinate system is redundant. In this thesis, relative localization is considered in Section 3.4 providing insight into this alternative localization method.

1.3 Thesis Contributions

The work presented in this thesis examines the tradeoff between localization *accuracy* and communication *bandwidth* required for distributed, cooperative location estimation in wireless networks, i.e. localization with nodes that cooperatively exchange range measurements and exploit network connectivity. Such problems are inspired by practical setups where communication bandwidth is limited, e.g. in underwater systems with narrow-band acoustic modems or in energy-constrained sensor networks where the size of localization messages should be limited. Bandwidth-friendly versions of factor graph based cooperative localization are proposed and compared with wide-band graph based cooperative localization, cooperative particle filtering, as well as distributed multi-dimensional scaling. The trade-off between localization accuracy and size of total messages exchanged is quantified in the presence of both Gaussian and non-Gaussian unimodal ranging error noise. It is shown that the proposed algorithms can significantly reduce the total size of exchanges messages, one-to-two orders of magnitude. Additionally, a computationally-efficient version is shown to further reduce the localization mean square error compared to state-of-the-art. Cramer-Rao bound calculation corroborates the efficiency of the proposed algorithms.

1.4 Outline of Thesis

The goal of this thesis is to quantify the performance of various known localization algorithms and specifically to explore the trade-off between localization accuracy and required bandwidth. First, Chapter 2 introduces various measurement modalities and presents their major sources of error. Chapter 3 reviews a subset of the rich localization literature by introducing and testing various iterative, distributed, cooperative localization algorithms and presents a factor graph based cooperative localization algorithm, the *Sum Product Algorithm over Wireless Networks* (SPAWN). Chapter 4 extends the SPAWN framework by introducing a bandwidth-friendly version, *parameterized SPAWN* (P-SPAWN) which is extensively simulated, evaluated as far as localization accuracy is concerned and shown to significantly reduce required bandwidth without sacrificing localization accuracy compared to the original SPAWN. Furthermore, a back to back scheme (BtB) is proposed and shown to further reduce the localization mean square error, thus out-performing state-of-the-art algorithms. Cramer-Rao bound calculation corroborates the efficiency of the proposed algorithm. Finally, Chapter 5 summarizes the whole thesis, presents the conclusions and gives some suggestions for future work.

Chapter 2

Ranging Measurements & Models

Ranging and angle measurements form the basis of any localization system. Information regarding relative or absolute position can be directly or indirectly extracted by range and/or angle measurements between communicating sensors. Understanding each type of measurement, the major sources of error, as well as the difficulties associated with it is a key to developing reliable localization systems.

The information and models presented in this Chapter have been reported and verified in the literature, to be good approximations for the very complicated behavior of pair-wise measurements in the notoriously unpredictable RF channels. It is extremely important to note that the information presented here is the base of deriving Cramer-Rao bounds on location performance in cooperative localization, lower bounds independent of the particular localization algorithm employed [11, 27]. Part of the assumptions and properties presented here are used in our own derivation of the CRB for 3D Long-Baseline Localization (LBL) systems in Appendix 1.

This Chapter serves the purpose of a general introduction to each type of measurement and briefly presents the key points related to them. Throughout the rest of this thesis it is assumed that range measurements between sensors are obtained either via RSS or TOA or perhaps a combination of the two. Additionally, relative angle measurements are assumed to be obtained exclusively via AOA. Both RSS and TOA can be measured via RF or by acoustic media; both media are subject to multipath and shadow fading phenomena which impair range estimates. In a similar fashion, AOA measurements are impaired by additive noise and multipath.

2.1 Received Signal Strength (RSS)

Received signal strength (RSS) is typically defined as the signal power received by a reference antenna. An indication is provided by the voltage measured at the receiver's received signal strength indicator (RSSI) circuit. When neighboring wireless sensors communicate, RSS of RF signals can be measured relatively inexpensively and in a simple fashion without presenting additional bandwidth or energy requirements. Due to this fact, RSS measurements have become immensely popular in the localization community. Unfortunately RSS measurements are also extremely unpredictable and error-prone.

Range measurements based on RSS degrade with distance. In free space, signal power decays proportional to d^{-2} where d is the distance between transmitter and receiver. In real world channels, objects in the environment have the effect of multiplying the signal energy by attenuation factors. The cumulative effect of many such multiplications, by a central limit argument, results in a log-normal distribution of RSS at the receiving end [17]. If $\mathcal{P}_{i,j}(mW)$, the received power in mW at sensor i transmitted by sensor j , is log-normal, then the received power in dB, $P_{i,j} = 10 \log_{10} \mathcal{P}_{i,j}(mW)$, is Gaussian. Thus $P_{i,j}$ is typically modeled as

$$\begin{aligned} P_{i,j} &\sim \mathcal{N}(\bar{\mathcal{P}}_{i,j}, \sigma_{dB}^2) \\ \bar{\mathcal{P}}_{i,j} &= P_0 - 10n_p \log_{10}(d_{ij}/d_0) \end{aligned} \quad (2.1)$$

where $\bar{\mathcal{P}}_{i,j}$ is the mean power in decibel milliwatts at distance d_{ij} , σ_{dB}^2 is the variance of the shadowing, P_0 is the received power at distance d_0 (typically $d_0 = 1$) calculated from the free space path loss formula and the path loss exponent n_p is determined by the environment [16, 36].

2.1.1 Range estimation via RSS

Using the model for received power presented previously one can derive a maximum likelihood (ML) estimate of the distance between sensors i and j

$$\delta_{i,j}^{ML} = d_0 10^{\frac{(P_0 - P_{i,j})}{10n_p}} \quad (2.2)$$

It is important to note that $\delta_{i,j}^{ML}$ has a log-normal distribution since $\log \delta_{i,j}^{ML}$ is Gaussian and that

$$E \{ \delta_{i,j}^{ML} \} = \mathcal{C} d_{ij} \quad (2.3)$$

where

$$\mathcal{C} = e^{\frac{\gamma}{2}}, \quad \text{where } \gamma = \left(\frac{10n_p}{\sigma_{dB} \log 10} \right)^2 \quad (2.4)$$

i.e., the ML estimator is biased. Therefore, a bias-corrected estimator can be offered by dividing by \mathcal{C} :

$$\delta_{i,j}^{ML} = (d_0/\mathcal{C}) 10^{\frac{(P_0 - P_{i,j})}{10n_p}} \quad (2.5)$$

The most important result of the log-normal model is that RSS-based range estimates (using any of the two estimators presented) have variance proportional to their actual range. This characteristic of RSS-based estimation leads to very high errors at large path lengths, which have limited its application in traditional localization systems. Clearly, RSS is most valuable in high-density sensor networks, where distances between communicating nodes tend to be small.

2.2 Time of Arrival (TOA)

Time-of-Arrival (TOA) is the measured time at which a signal (RF, acoustic, or other) first arrives at a receiver. The measured TOA between transmission at sensor i and reception at sensor j includes the time of transmission plus a propagation-included time delay, $T_{i,j}$, equal to the transmitter-receiver separation distance, d_{ij} , divided by the propagation velocity, u_p . The cornerstone of time-based techniques is the receiver's ability to accurately estimate

the arrival time of the direct line-of-sight (DLOS) signal, a process which is hampered both by additive noise and multipath signals.

The power in the DLOS path is attenuated by any obstacles in between transmitter and receiver and it is quite common for later-arriving non-line-of-sight (NLOS) multipath components to arrive at the receiver with greater of equal power than the DLOS. Later-arriving paths contribute an increasing proportion of the overall power received power, as the distance between two communicating devices increases [23, 36]. Yet, even in the absence of multipath, the accuracy of the arrival time is limited by additive noise, with a lower bound provided in [38] through computation of the Cramer-Rao Bound (CRB) in a multipath-free channel. Finally time delays in the transmitter and receiver hardware and software further add to the measured TOA, introducing additional errors.

2.2.1 Statistical model for TOA

Measurement campaigns in [11, 27] have shown that for short range measurements, measured time delay can be roughly modeled as Gaussian

$$f(T_{i,j} = t|\mathbf{p}) = \mathcal{N}_t\left(\frac{d_{ij}}{u_p} + \mu_T, \sigma_T^2\right) \quad (2.6)$$

where μ_T, σ_T^2 are the mean and variance of the time delay error, \mathbf{p} is defined in (1.1), d_{ij} is the actual range between i and j , u_p is the propagation velocity and the notation $\mathcal{N}_x(\mu, \sigma^2)$ is used to denote the Gaussian distribution with mean μ and variance σ^2 , evaluated in t . Typical values of μ_T and σ_T^2 have been obtained with extensive measurements, as reported in [33] or can be estimated as nuisance parameters [15]. Obviously, an estimator for the time delay can be constructed [10, 41].

It is interesting to note that implementations of Ultra Wideband (UWB) based range measurements using TOA have been reported in [3, 19]. The very high bandwidth of UWB leads to a very high temporal resolution, making it ideal for high precision radiolocation applications. Extensive measurement campaigns using commercial UWB radios in [47] produce a model of the UWB ranging measurements \hat{d} as a function of true distance d in an

environment \mathcal{E} as a Gaussian density described by

$$p(\hat{d}|d, \mathcal{E}) = \mathcal{N}_{\hat{d}}(P_{\mathcal{E},\mu,2(d)}, P_{\mathcal{E},\sigma^2,2(d)}) \quad (2.7)$$

where $P_{\mathcal{E},\mu,2(d)}$, $P_{\mathcal{E},\sigma^2,2(d)}$ are carefully selected quadratic polynomials, depending upon the condition (i.e LOS, NLOS etc.) [47].

2.2.2 Time-difference of Arrival (TDOA)

Time-of-Arrival measurements present a serious drawback: communicating sensors must be synchronized. The state of each sensors clock (its bias compared with absolute time) can also be considered to be an unknown parameter and included in the parameter vector \mathbf{p} . In this case, one-way TOA is measured and input to a localization algorithm which estimates both the sensor coordinates and the biases of each sensors clock [27]. The difference between the arrival times of the same signal at two sensors is called the time-difference of arrival (TDOA). A TDOA measurement does not depend on the clock bias of the transmitting sensor. TDOA methods have been used in source localization for decades for locating asynchronous transmitters, and has application in GPS and cellular localization. Interestingly, under certain weak conditions, it has been shown that TOA with clock bias (treated as an unknown parameter) is equivalent to TDOA [42].

2.3 Angle of Arrival (AOA)

Angle-of-Arrival measurements provide information complementary to the RSS and TOA/TDOA measurements presented in the previous sections. As the name suggests, AOA measurements provide localization information about the direction to neighboring sensors as opposed to distance to neighboring sensors.

Angle-of-Arrival estimation is a conquered problem, with many estimation algorithms and associated properties proposed [32, 43, 45]. The most common method to measure AOA is to use a sensor array combined with

signal processing techniques at the sensor nodes. To be more specific, when such is the case, each sensor node consists of two or more individual sensors (e.g., antennas for RF, microphones for acoustic signal) whose location with respect to the node center are known. The AOA is estimated, similar to time delay estimation, from the differences in the arrival times for a transmitted signal to each of the individual sensors integrated in the receiving node.

Alternatively, it is quite common for two or more directional antennas located on the sensor to be deployed. If the antennas are pointed in different directions such that their main beams overlap, AOA can be estimated using the ratio of their individual RSS values.

If device cost and size is of significant importance, the demand of multiple antenna elements may render the AOA approach impractical. However, in many cases all the required hardware for AOA estimation may be already present in the application. Consider for example [?], where the purpose of the sensor network is to identify and locate acoustic sources. Acoustic sensors are required by the application in order to operate. If such is the case, locating the sensors themselves using acoustics in these applications is a natural extension. Furthermore, despite the cost, localization may not even be achievable without AOA estimation [1], a common case in relative location estimation.

Unfortunately, AOA measurements are impaired by additive noise and multipath phenomena, the very same sources discussed in the TOA section above.

2.3.1 Statistical model for AOA

AOA measurements are typically modeled as Gaussian, with ensemble mean equal to the true angle to the source and standard deviation σ_α . Standard deviation bounds have been reported, both for acoustic based AOA estimation [13] as well as RF AOA estimation (using the RSS ratio method) [4].

As a final note, AOA estimation requires sensors be placed with known orientation (e.g., using a built in compass). If such information is not available, the network localization problem must be extended to consider each

sensor's orientation as an unknown parameter, to be estimated along with position.

For a more detailed discussion on AOA and AOA estimation, the reader is referred to [4, 13, 32, 43, 45] and references therein.

2.4 Connectivity

When simple, inexpensive, backward-compatible location measurements are of interest its quite common for localization research to consider connectivity (also referred to as proximity) measurements [16]. in conjunction with other measurement types.

Two devices can determine whether or not they communicate independently of whether or not accurate RSS measurement circuitry is available at their receivers. In truth, two sensors are considered to be connected if the receiving sensor can successfully demodulate packets transmitted by the other sensor (i.e distance between sensors is not the only decisive factor). If RSS is too low the receiver fails to successfully demodulate packets.

It is apparent that connectivity is a function of RSS, considered random due to the randomness of RSS measurements (see section 2.1). Specifically, if P_{thr} denotes the receiver threshold (dBm) under which packets cannot be demodulated and P_{ij} denotes the received power (dBm) at sensor i transmitted by sensor j , then the connectivity measurement of sensors i and j , $\mathcal{C}_{i,j}$ can be modeled as

$$\mathcal{C}_{i,j} = \begin{cases} 1, & \text{if } P_{ij} \geq P_{thr} \\ 0, & \text{if } P_{ij} < P_{thr} \end{cases} \quad (2.8)$$

i.e connectivity is a binary quantization of Received Signal Strength [16].

Using the above definition of proximity and the model for P_{ij} presented equation (2.1) of section 2.1 it can be shown that the probability mass function (PMF) of $\mathcal{C}_{i,j}$ given the location coordinates $\mathbf{x}_i, \mathbf{x}_j$ of sensors i, j is

$$p(\mathcal{C}_{i,j} = s | \mathbf{x}_i, \mathbf{x}_j) = s + (-1)^s \Phi \left[\sqrt{\gamma} \ln \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{d_s} \right] \quad (2.9)$$

where

$$\gamma = \left(\frac{10n_p}{\sigma_{dB} \log 10} \right)^2 \quad (2.10)$$

where $s \in \{0, 1\}$, and d_s is the range at which the mean received power is P_s , equal to

$$\delta_s = d_0 10^{\frac{(P_0 - P_s)}{10n_p}} \quad (2.11)$$

as shown in section 2.1

Naturally, since connectivity is strongly related to RSS, it suffers from the same sources of error : multipath and channel fading. Furthermore, while randomness of packet errors given the received power level is an additional source of variation in proximity measurements, variation caused by the fading channel is far more severe, allowing packet related errors to be ignored without compromising generality.

2.5 Chapter 2 Conclusions

This chapter presented the basic principles of each type of pair-wise measurements : Received Signal Strength (RSS), Time-of-Arrival (TOA), Angle-of-Arrival (AOA) and connectivity (i.e., proximity) measurements.

The inherent importance of pair-wise measurements become apparent as this thesis progresses. For one, performance bounds are derived using the models presented in the previous section through computation of the Cramer-Rao Bound (CRB). These bounds do not rely on the specific algorithm deployed and thus can be used to quickly judge the precisions possible from various measurement modalities under a variety of scenarios.

In many cases, the available measurement types will determine the specific algorithm used to perform localization. The topics discussed in this chapter are fundamental to the localization algorithms as well as their performance, presented in the following chapter.

Chapter 3

Cooperative Localization Algorithms

Up until this chapter we have limited the discussion on an overview of wireless sensor networks and applications, as well as automatic location estimation and the available ranging measurement types, without considering the means by which accurate position estimation is achieved. Chapter 3 bridges that gap by introducing several cooperative localization algorithms, operating on the available signal metrics presented in chapter 2 to make applications such as the ones presented in chapter 1 possible.

This chapter is organized as follows. Section 3.1 provides a brief overview of the rich literature in sensor localization algorithms. Then section 3.2 presents the distributed weighted multi-dimensional scaling algorithm for node localization in wireless sensor networks.

Section 3.3 shows how the geometry of the topology can be taken advantage of in order to perform localization through an iterative localization algorithm operating on the barycentric coordinates of the sensors. This approach attempts to limit the need for anchor nodes to a minimum.

An introduction to relative location estimation is made in section 3.4. Here, the absence of anchor nodes forces agents to cooperatively localize in a relative coordinate system, using pair-wise distance and angle measurements. Relative location estimation, as presented in this section, opens the door to numerous swarming and routing applications.

Each localization algorithm presented is simulated under the exact same conditions. In section 3.5 conclusions are drawn regarding the limitations, the offered accuracy, as well as the bandwidth requirements of each approach.

3.1 Overview of Cooperative Localization Algorithms

The literature in cooperative sensor localization is rich and ever growing. The proliferation of low-cost wireless networking options, the advance of sensor network technologies and the inherent importance of location awareness in many scenarios (including social applications) has created an immense interest on localization techniques over the last years.

While positioning and navigation have a long history, enabling cooperation requires existing methods to be extended by finding ways to use pair-wise measurements between sensors in communication range. The challenge is to allow sensors of unknown location (agents) which are not in range of any known-location devices (anchors) to be located, and further, to improve the location estimates of all sensors.

The relevant literature is rich and includes centralized schemes, where location-dependent information must be conveyed and processed at a central station and distributed schemes, which rely on location estimation locally (i.e. not at a central server) and involve communication among neighboring nodes, possibly iteratively.

3.1.1 Centralized Algorithms

In centralized localization, the positions of all agents are determined by a central processor. This processor gathers measurements from anchors as well as agents and computes the positions of all the agents. Centralized schemes have demonstrated good estimation performance and the literature involves: a) exploitation of connectivity, proximity and maximum communication range information with geometric techniques (e.g. [9]) or linear programming [20] b) trilateration techniques with (noisy) distance measurements from known-location anchors (as in global positioning system or proprietary systems (e.g. [34],[46]) and c) exploitation of (noisy) distance measurements from anchors as well as unknown-location agents, as in classic multi-dimensional scaling (MDS) [18] or in semi- definite programming (e.g. [6, 7, 29]). Centralized

algorithms are, among others, usually not scalable and thus impractical for large networks

3.1.2 Distributed Algorithms

Research has demonstrated the feasibility of distributed localization algorithms, which are required for scalability and balancing computational costs across large sensor networks. In one approach, the immediate to the anchors neighbors localize first with standard triangulation techniques, then the neighboring nodes to the previous localized nodes perform triangulation and so forth (e.g. [39], where localization error accumulation is addressed with non-linear optimization).

Another approach involves iterative exchange of information among neighboring nodes until certain criteria are met. Examples include work in [31], where distance vectors are propagated (similarly to distance vector routing) and include distance to neighbors as well as hop count, distributed versions of the classic MDS scheme [16, 24, 40], work with barycentric coordinates [26] and finally, work based on factor graphs [47]. Specifically, work in [24] merges local maps in order to find out the global topology; work in [40] estimates (known) location of anchors and iteratively propagates corrections to agents; work in [16] crafts local functions through majorization, which are minimized numerically using gradient descent. Work in [26] assumes all agents in the convex hull formed by the anchors.

Factor graphs provide another distributed iterative method. These methods are particularly promising for sensor network localization - each sensor stores a conditional density of its own coordinates based on its own internal measurements and the conditional density of its neighbors. A good example is the factor graph-based algorithm in [47] which has been experimentally and successfully tested with ultra wide band (UWB) radios. Alternatively particle filtering (PF) methods have each sensor store a set of "particles", i.e., candidate representations of its coordinates, weighted accordingly. [28, 37, 44] These methods have been used to accurately locate and track mobile robots [28] and are a promising open topic for future research.

3.1.3 Centralized / Distributed Comparison

There are three big motivations for developing distributed localization algorithms. First, for some applications, no central processor, or none with enough computational power, is available to handle the calculations. Second, when a large network of sensors must forward all data to a single central processor, there is a communication bottleneck and thus a higher energy drain near the central processor. Third, as mentioned, centralized algorithms usually do not scale well as the network size increases.

Both centralized and distributed algorithms must face the high relative costs of communication. Centralized algorithms require each sensor's measurements to be sent over many hops to a central processor as opposed to distributed where sensors exchange one-hop data (but possibly make multiple iterations). In the general case, when the average number of hops to the central processor exceeds the necessary number of iterations, distributed schemes are likely more energy-efficient.

Some approaches to localization attempt to reduce energy consumption by combining centralized and distributed features. For example, in [24] the sensor network is divided into small clusters and a processor from within each cluster estimates a map of the cluster's sensors in a centralized fashion. Then, cluster processors operate a distributed algorithm to merge and optimize the local estimates.

3.2 Distributed Weighted Multi-dimensional Scaling (dwMDS)

This section introduces a distributed localization algorithm, based on a weighted version of multidimensional scaling (MDS) [18], which naturally incorporates local communication constraints within the sensor network [16]. The key features of dwMDS are:

1. A weighted cost function that allows range measurements to be weighted according to their accuracy; measurements that are believed to be more

accurate are weighted more heavily.

2. An adaptive neighbor selection method that avoids the biasing effects of selecting neighbors based on noisy range measurements (see section 2.1 and subsection 3.2.1).
3. A majorization method which has the property that each iteration is guaranteed to improve the value of an objective cost function to be minimized.

In this approach, sensor position estimation is achieved by minimizing the following global cost function (a.k.a STRESS function [18]):

$$S = 2 \sum_{i=1}^{\mathcal{N}} \sum_{j=i}^{\mathcal{N}+\mathcal{N}_A} w_{ij} (\delta_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 + \sum_{i=1}^{\mathcal{N}} r_i \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2 \quad (3.1)$$

where D -dimensional coordinates $\{\mathbf{x}_i\}$ are found which minimize S given:

- pair-wise measurements δ_{ij} between neighboring sensors,
- an arbitrary weight associated with the predicted accuracy of the measurements,
- imperfect prior information : with accuracy r_i , agent node i is located at coordinates $\bar{\mathbf{x}}_i$, ($1 \leq i \leq \mathcal{N}$),
- and perfect prior information regarding the positions of the anchor nodes j , ($\mathcal{N} < j \leq \mathcal{N}_A$).

In summary, the first \mathcal{N} nodes (agents whose positions are to be estimated) have imperfect prior information while the remaining sensors $\mathcal{N} + 1 \dots \mathcal{N}_A$ have perfect prior coordinate information. If no prior information is available for agent node i , $r_i = 0$, and furthermore if no measurement δ_{ij} is available between sensors i and j , then $w_{ij} = 0$. It is assumed that $w_{ij} \geq 0$, $w_{ii} = 0$ and $w_{ij} = w_{ji}$, i.e., the weights are non-negative and symmetric. After simple manipulations, S can be re-written as follows:

$$S = \sum_{i=1}^{\mathcal{N}} S_i + c \quad (3.2)$$

where c is a constant independent of the nodes location and local cost functions S_i are defined for each agent (i.e., $1 \leq i \leq \mathcal{N}$)

$$\begin{aligned}
 S_i = & 2 \sum_{j=1, j \neq i}^{\mathcal{N}} w_{ij} (\delta_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 + \sum_{j=\mathcal{N}+1}^{\mathcal{N}} 2w_{ij} (\delta_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 \\
 & + r_i \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2
 \end{aligned} \tag{3.3}$$

Motivated by this cost structure, the authors in [16] propose an iterative scheme in which each sensor minimizes its corresponding cost function after collecting distance measurements and position estimates from its neighbors, thus updating its own position estimate (to be shared in subsequent iterations).

3.2.1 The dwMDS two-stage algorithm

Unlike classical MDS, no closed form expression exists for the minimum of the local or global cost function. Instead, in [16], the authors minimize S_i iteratively using quadratic majorizing functions as in SMACOF (Scaling by MAjorizing a COmplicated Function [22]), which generates a sequence of non-increasing STRESS values. A majorizing function $T_i(\mathbf{x}, \mathbf{y})$ of $S_i(\mathbf{x})$ is a function $T_i : \mathbf{R}^{D \times D} \rightarrow \mathbf{R}$ that satisfies the following properties:

1. $S_i(\mathbf{x}) \leq T_i(\mathbf{x}, \mathbf{y}) \quad \forall \mathbf{y}$
2. $S_i(\mathbf{x}) = T_i(\mathbf{x}, \mathbf{x})$

As a result, function $S_i(\mathbf{x})$ can be minimized using the following approach. Starting an an initial condition \mathbf{x}_0 , the majorizing function $T_i(\mathbf{x}, \mathbf{x}_0)$ is minimized as a function of \mathbf{x} . A new majorizing function $T_i(\mathbf{x}, \mathbf{x}_1)$ can then be defined by using the newly found minimum \mathbf{x}_1 . This process can be repeated until convergence (see [22] for details).

By utilizing a simple majorizing function that can be minimized analytically, e.g., a quadratic function, it can be shown [16] that if $\mathbf{X}^{(k)}$ is the matrix whose columns contain the position estimates for all sensors at iteration k ,

an update for the position estimate of node i is given by:

$$\mathbf{x}_i^{(k+1)} = \alpha_i \left(r_i \bar{\mathbf{x}}_i + \mathbf{X}^{(k)} \mathbf{b}_i^{(k)} \right) \quad (3.4)$$

where

$$\alpha_i^{-1} = \sum_{j=1, j \neq i}^{\mathcal{N}} w_{ij} + \sum_{j=\mathcal{N}}^{\mathcal{N}+\mathcal{N}_A} 2w_{ij} + r_i \quad (3.5)$$

and $\mathbf{b}_i^{(k)} = [b_1, \dots, b_{\mathcal{N}+\mathcal{N}_A}]$ is a vector whose entries are given by:

$$\begin{aligned} b_j &= w_{ij} \left[1 - \delta_{ij} / \|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\| \right] & j \leq \mathcal{N}, j \neq i \\ b_i &= \sum_{j=1}^{\mathcal{N}} w_{ij} \delta_{ij} / \|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\| + \sum_{j=\mathcal{N}+1}^{\mathcal{N}+\mathcal{N}_A} 2 w_{ij} \delta_{ij} / \|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\| & (3.6) \\ b_j &= 2w_{ij} \left[1 - \delta_{ij} / \|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\| \right] & j > \mathcal{N} \end{aligned}$$

Note that the weights w_{ij} are non-zero only for nodes j in the neighborhood of node i meaning that the update rule for \mathbf{x}_i will depend only on its neighborhood. The algorithm that implements the aforementioned iterative scheme is presented bellow.

Algorithm 3.2.1: DWMDs($\{\delta_{ij}\}, \{w_{ij}\}, \{r_i\}, \bar{\mathbf{x}}_i, \mathcal{N}_A, \epsilon, \text{initial condition } X^{(0)}$)

Initialize : $S^{(0)}$, compute α_i from equation (3.5), $k \leftarrow 0$

repeat

$k \leftarrow k + 1$

for $i = 1$ **to** \mathcal{N}

 compute $\mathbf{b}_i^{(k-1)}$ from equation (3.6)

$\mathbf{x}_i^{(k)} \leftarrow \alpha_i \left(r_i \bar{\mathbf{x}}_i + \mathbf{X}^{(k-1)} \mathbf{b}_i^{(k-1)} \right)$

 compute $S_i^{(k)}$

$S^{(k)} \leftarrow S^{(k)} - S_i^{(k-1)} + S_i^{(k)}$

 communicate $\mathbf{x}_i^{(k)}$ to neighbors of node i

 communicate $S^{(k)}$ to node $i + 1 \pmod{\mathcal{N}}$

end for

until $S^{(k-1)} - S^{(k)} < \epsilon$

Note the following comments:

1. The choice of the weighting function w_{ij} should reflect the accuracy of the pair-wise measurements between sensors i and j , such that less accurate measurements are down-weighted in the overall cost function. While weights can be tailored to the variance predictions of the underlying noise measurement model, one can adopt a model-independent weight assignment such as

$$w_{i,j} = \begin{cases} \exp \{-\delta_{ij}^2/h_{ij}^2\}, & \text{if } \delta_{ij} \text{ is measured} \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

where $h_{ij} = \max [\{\delta_{i,k}\}_k \cup \{\delta_{k,j}\}_k]$, i.e., a non-zero, symmetric LOESS-based (see [12]) weighting scheme, shown to perform robustly in [16].

2. The values of r_i should quantify the knowledge about the prior information regarding node i 's position. If r_i 's are very high (compared to w_{ij} 's) then the solution to (3.1) will "stretch" range measurements in order to place sensors with prior information at their a-priori coordinates. On the other hand, if r_i 's are very low, then the solution will attempt to preserve range measurements and instead find a global translation and rotation that results in agreement between estimates and prior coordinates. Typical values of r_i should range between 10^{-2} and 10^2 , and as long as r_i 's are within the correct order of magnitude the results will be near-optimal.
3. Regarding initialization, each node requires an initial estimate of its position. In section 3.3 it can be seen that the algorithm is relatively robust with respect to "rough" initial position estimates. Furthermore, while it is assumed in the description of the algorithm that updates are performed in an ordered fashion (i.e., in the order $1, 2, \dots, \mathcal{N}$), many other non-cyclic update rules are possible. One possibility is for clusters of sensors to iterate among themselves until their position estimates stabilize. The estimates can then be transmitted to neighboring clusters, before starting a new iteration step.

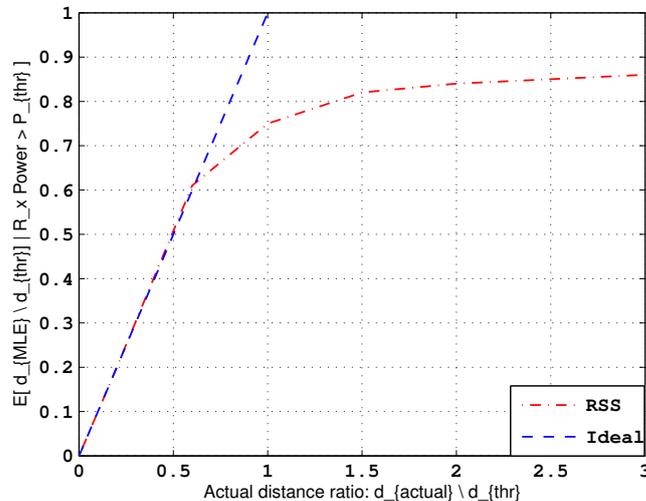


Figure 3.1: The expected value of the RSS-based estimate of range given that two devices are neighbors and the ideal unbiased performance

4. Although the majorization approach used guarantees an non-increasing STRESS function, a major drawback is that, like any gradient search method, it may converge to a local minimum of the cost function.

The process of selecting neighbors based on noisy distance measurements needs to be carefully addressed when the measurement type is RSS-based, due to the fact that the distance estimation provided is in fact biased. The RSS-based biasing effect has been addressed in section 2.1.1. Specifically, the range estimator offered is negatively-biased. Figure 3.1) plots the expected value of the RSS-based estimate of range given that two sensors are connected and the ideal unbiased performance. The channel has $\sigma_{dB}/n = 1.7$ and the $d_R = 1$ (i.e., distances are normalized by d_R which denotes the distance under which sensor are considered neighbors). Ideally, the range estimator should have a mean value equal to the actual range. However, as the range increases, the expected value of δ_{ij} (given that i and j are neighbors) deviates from linear and asymptotically becomes constant. This is a strong negative bias for sensors separated by d_R or greater. Note that the discussion here is also applicable to systems which use noisy TOA-based range estimates for neighbor selection.

To counter the negative biasing effect of selecting neighbors based on noisy distance measurements a two-stage algorithm is proposed, based in the predicted distances between sensors.

Algorithm 3.2.2: SELECTIONALGORITHM($\{\delta_{ij}\}, \{w_{ij}\}, \{r_i\}, \bar{\mathbf{x}}_i, \mathcal{N}_{\mathcal{A}}, \epsilon, X^{(0)}$)

Stage 1: Execute dwMDS with light termination condition ϵ_1

$\{\hat{\mathbf{x}}\} \leftarrow \text{DWMDS}(\{\delta_{ij}\}, \{w_{ij}\}, \{r_i\}, \bar{\mathbf{x}}_i, \mathcal{N}_{\mathcal{A}}, \epsilon_1, X^{(0)})$
 Compute new neighborhood structure w'_{ij} based on interim estimates $\{\hat{\mathbf{x}}\}$

Stage 2: Execute dwMDS using $\{\hat{\mathbf{x}}\}$ as initial condition

$\{\mathbf{x}\} \leftarrow \text{DWMDS}(\{\delta_{ij}\}, \{w'_{ij}\}, \{r_i\}, \bar{\mathbf{x}}_i, \mathcal{N}_{\mathcal{A}}, \epsilon_2, \{\hat{\mathbf{x}}\})$

In the first step the dwMDS algorithm is run with a neighborhood structure based on the available range measurements using (3.7) and a connectivity constraint (two sensors are considered connected if $\delta_{ij} < d_R$ where d_R is a problem specific parameter). An interim estimate $\hat{\mathbf{x}}$ of the sensors location is provided after convergence.

In the second step the (negatively biased) predicted distances resulting from the interim estimate $\hat{\mathbf{x}}$ are used as connectivity constraints to construct a new neighborhood structure. Then using $\hat{\mathbf{x}}$ as an initial condition and the new neighborhood structure, the dwMDS algorithm is re-run, resulting in the final location estimates of the sensors. Note that the predicted distances are used only to select neighbors (i.e., which weights are positive)- the measured ranges δ_{ij} as still used to determine the weight values.

The numerical result in the next section demonstrate that the two stage selection algorithm effectively counters the negative bias phenomenon. It is important to note, that the selection scheme proposed does not imply twice the computation. Since the first step need to provide coarse location information, it does not need to be accurate and can be terminated quickly with a large ϵ . The second step begins with a good (although biased) estimate and will likely require fewer iteration to converge.

3.2.2 Performance of the dwMDS method

The performance of the dwMDS method is demonstrated on the two-dimensional $50 \times 50m$ network of figure 4.3. We consider ranging error measurements given by $\delta_{ij} = d_{ij} + w_{ij}$, where d_{ij} is the true Euclidean distance and w_{ij} is zero mean Gaussian noise of variance σ^2 .

The experiments are run with the following configurations. To achieve the connectivity of 4.3, d_R is set to $35m$, allowing each agent to be connected with only two anchors; cooperation is obviously needed in order to localize the nodes.

Figure 3.3 depicts the mean squared error (MSE) as a function of the ranging error variance σ^2 with and without assuming prior knowledge about the nodes location. Prior knowledge is considered as an a-prior indication of the agents locations. It is evident that, given prior information, the dwMDS shows impressive performance. This is not the case for limited prior knowledge. The dwMDS is thus, sensitive upon initialization.

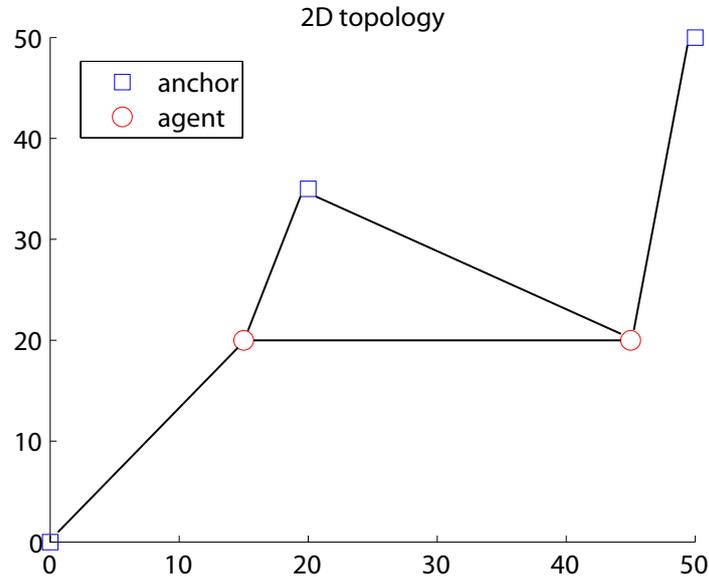


Figure 3.2: Network topology for 2-dimensional localization. Observe that agent are connected only with two anchor and need to cooperate in order to localize.

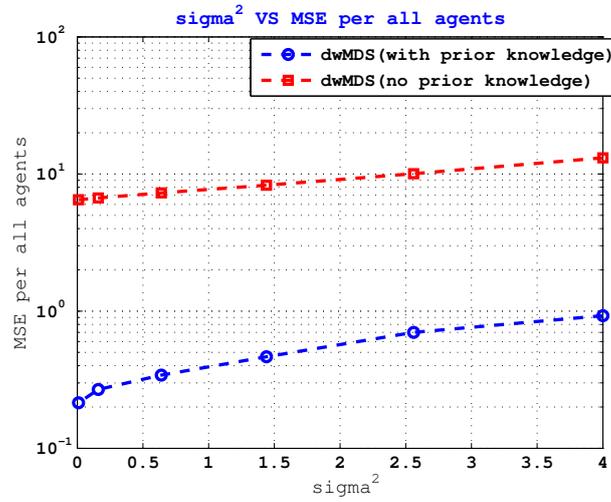


Figure 3.3: Mean squared error (MSE) as a function of ranging noise variance σ_r^2 for 2D localization with the dwMDS method. Operations are performed with and without assuming prior knowledge regarding the agents locations

Bandwidth requirements are presented in figure 3.4 for both cases. The dwMDS scheme has reasonable bandwidth requirements. As the network density, size and dimensionality increases, so do the bandwidth requirements. The same principle applies to the ranging error variance.

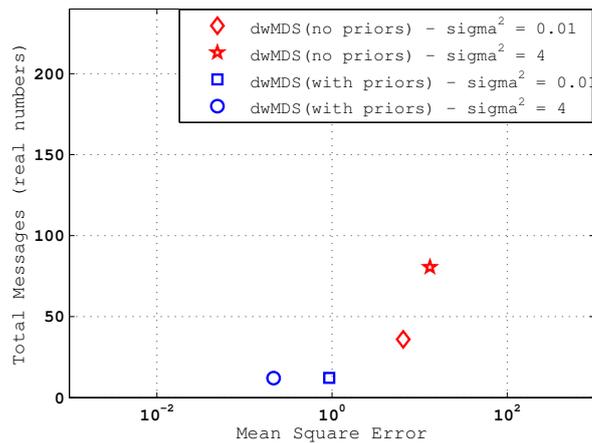


Figure 3.4: Bandwidth (BW) performance as a function of MSE for 2D localization with the dwMDS method. Operations are performed with and without assuming prior knowledge regarding the agents locations

Finally, figures 3.5 and 3.6 depict the same setup for three-dimensional localization. The conclusion drawn in the two-dimensional case can be generalized to higher dimensions.

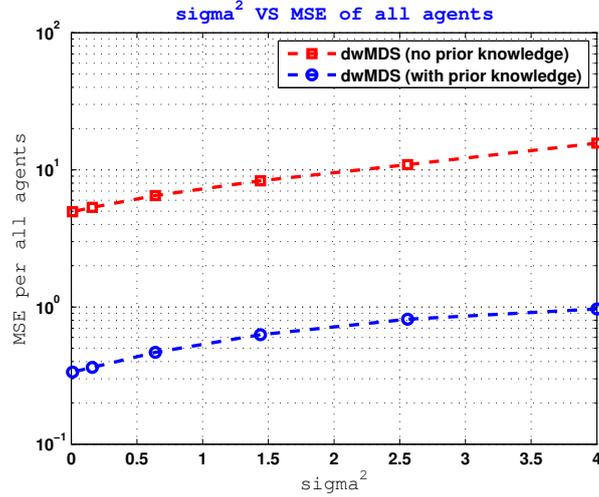


Figure 3.5: Mean squared error (MSE) as a function of ranging noise variance σ_r^2 for 3D localization with the dwMDS method. Operations are performed with and without assuming prior knowledge regarding the agents locations

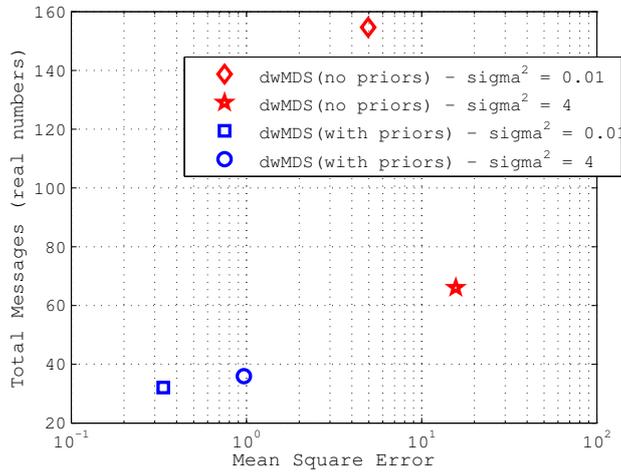


Figure 3.6: Bandwidth (BW) performance as a function of MSE for 3D localization with the dwMDS method. Operations are performed with and without assuming prior knowledge regarding the agents locations

3.3 Distributed Localization in Random Environments (DLRE)

This section introduces *DLRE* [26], a distributed, iterative algorithm to cooperatively locate \mathcal{N} agent in $\mathbf{R}^{\mathcal{N}_A-1}$, with respect to a minimal number of \mathcal{N}_A anchors with a priori known location. The proposed algorithm operates under the strict assumption that all agents lie in the convex hull formed by the anchors and under these conditions is shown to perform robust localization under a variety of random operating conditions, with guaranteed convergence and without requiring more than \mathcal{N}_A anchors.

Before moving on, it is useful to cover the notation used throughout the rest of this section. The agents and the anchors lie in $\mathbf{R}^{\mathcal{N}_A-1}$. Let Θ be the set of nodes in the network decomposed as

$$\Theta = k \cup \Omega \quad (3.8)$$

where k is the set of known location nodes, the anchors, and Ω is the set of agents whose locations are to be estimated. For a set Ψ of nodes, its convex hull (i.e., the smallest convex set containing Ψ) is denoted by $\mathcal{C}(\Psi)$. For example, if Ψ is a set of three non-collinear nodes in a plane, then $\mathcal{C}(\Psi)$ is a triangle. A_Ψ denotes the generalized volume (area in $D = 2$, volume in $D = 3$, and their generalization in higher dimensions) of Ψ . If d_{lk} denotes the true Euclidean distances between two nodes $l, k \in \Theta$ then, the neighborhood of node l in a given radius r_l is:

$$\mathcal{K}(l, r_l) = \{k \in \Theta : d_{lk} < r_l\} \quad (3.9)$$

Finally, due to the iterative nature of the approach presented, the notation $c_l(t)$ is used to represent the estimated location vector of the node l at iteration t , while the true location (to be estimated) of node l is represented by c_l^* .

DLRE is developed under the following assumptions:

- *Convexity*: All agent lie inside the convex hull of the anchors

$$\mathcal{C}(\Omega) \subset \mathcal{C}(k) \quad (3.10)$$

- *Anchor nodes*: The anchors' locations are known a priori, i.e., their state remains constant

$$c_q(t) = c_q^*, \quad q \in k, \quad t \geq 0. \quad (3.11)$$

- *Non-degeneracy*: The anchors do not lie on a hyperplane. Thus

$$A_k \neq 0 \quad (3.12)$$

- *Internode Communication*: There is a communication link between all of the nodes in the set $\{l\} \cup \mathcal{K}(l, r_l)$, $\forall l \in \Omega$.

With the above assumption it follows easily the for every agent $l \in \Omega$, there exists some $r_l > 0$ such that a *triangulation set*, $\Theta_l(r_l)$, satisfying the following conditions:

$$\begin{aligned} \Theta_l(r_l) &\subset \mathcal{K}(l, r_l), \quad l \notin \Theta_l(r_l), \quad l \in \mathcal{C}(\Theta_l(r_l)) \\ |\Theta_l(r_l)| &= \mathcal{N}_{\mathcal{A}} \quad A_{\Theta_l(r_l)} \neq 0 \end{aligned} \quad (3.13)$$

exists, where $|\cdot|$ denotes the cardinality of a set. Finding the triangulation set Θ_l is a crucial step in DLRE, referred to as *triangulation*.

Finally, DLRE is expressed in terms of the barycentric coordinates, α_{lk} , of the node $l \in \Omega$, with respect to the nodes, $k \in \Theta$. The barycentric coordinates are unique and are given by (see [?]] and [?])

$$\alpha_{lk} = \frac{A_{\{l\} \cup \Theta_l \setminus \{k\}}}{A_{\Theta_l}} \quad (3.14)$$

with $A_{\Theta_l} \neq 0$, where " \setminus " denotes the set difference, and $A_{\{l\} \cup \Theta_l \setminus \{k\}}$ is the generalized volume of the set Θ_l with the node l added and the node k removed.

Barycentric coordinates can be computed using only the information available at each node, namely the inter-node distances d_{lk} using the Cayley-Menger determinants. To be specific, for $m + 1$ points in \mathbf{R}^m :

$$A_{\Theta_l}^2 = \frac{1}{s_{m+1}} \begin{vmatrix} 0 & \mathbf{1}_{m+1}^T \\ \mathbf{1}_{m+1} & \mathbf{Y} \end{vmatrix} \quad (3.15)$$

where $\mathbf{Y} = \{d_{l,j}^2\}, l, j \in \Theta_l$ is the matrix of squared distances among the $m + 1$ points in Θ_l and

$$s_m = \frac{2^m (m!)^2}{-1_{m+1}} \quad (3.16)$$

is an integer sequence through which the generalized volume A_{Θ_l} of the convex hull $C(\Theta_l)$ of the $m + 1$ points in \mathbf{R}^m is related to the Cayley-Menger determinant.

3.3.1 The DILOC algorithm

Before addressing realistic scenarios, a DIstributed LOCalization (DILOC) algorithm operating under noise free conditions is presented. The DILOC algorithm is then further extended to account for random link failures and additive channel noise leading to the DLRE algorithm. Both variations include a common *setup (triangulation) phase* as well as an iterative *state updating phase*.

DILOC \ DLRE Setup Phase: In the setup stage, each sensor l attempts to triangulate itself (i.e the triangulation set Θ_l is determined). To this end, each sensor l establishes a communication radius r_l and arbitrarily chooses \mathcal{N}_A nodes within that radius. Node l tests if it lies in the convex hull of these nodes and repeats this process with all collections of \mathcal{N}_A nodes in r_l until Θ_l is determined. If all attempts fail, the communication range is increased and the process is repeated. Due to the fact that node l lies in the convex hull formed by the anchors, success is eventually achieved and node l is triangulated by finding Θ_l with the aforementioned properties.

A straight-forward procedure, aimed to act as a convex hull inclusion test is presented below. Consider a set of \mathcal{N}_A nodes denoted as κ . Clearly, if

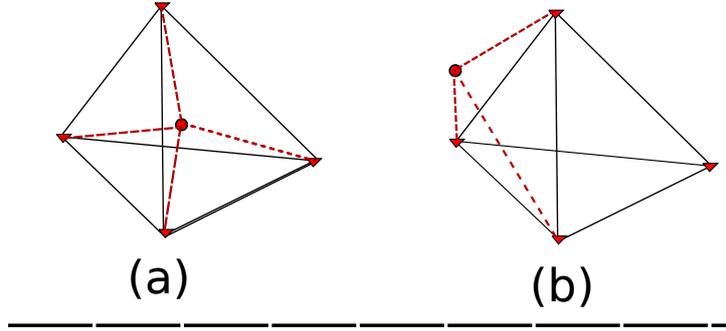


Figure 3.7: **A three dimensional convex hull inclusion test: Sensor l is depicted by "o" while neighbors are depicted by " ∇ ". (a) $A_k = A_{k \cup \{l\}}$ and (b) $A_k > A_{k \cup \{l\}}$**

node l lies in the convex hull $\mathcal{C}(\kappa)$ then

$$\mathcal{C}(\kappa) = \mathcal{C}(\kappa \cup \{l\}) \quad (3.17)$$

i.e. the two convex hulls are one and the same and as a result share the same generalized volume. Thus if A_κ and $A_{\kappa \cup \{l\}}$ denote the generalized volumes of $\mathcal{C}(\kappa)$ and $\mathcal{C}(\kappa \cup \{l\})$ respectively

$$A_\kappa = A_{\kappa \cup \{l\}} = \sum_{k \in \kappa} A_{\kappa \cup \{l\} \setminus \{k\}} \quad (3.18)$$

and a convex hull inclusion test, shown in figure 3.7) can be constructed as

$$A_\kappa = \sum_{k \in \kappa} A_{\kappa \cup \{l\} \setminus \{k\}} \quad (3.19)$$

It is noteworthy that due to the fact that the above inclusion test is based solely on generalized volumes, only the distance information in the Cayley-Menger determinants is required for calculation.

DILOC State Update Phase: Once all agents $l \in \Omega$ have triangulated the DILOC setup phase is complete. At time $t + 1$, each agent l iteratively updates its location estimate, by a convex combination of the states of the

nodes in Θ_l at time t :

$$c_l(t+1) = \begin{cases} c_l(t), & l \in \kappa, \\ \sum_{k \in \Theta_l} \alpha_{lk} c_k(t) & l \in \Omega \end{cases} \quad (3.20)$$

where α_{lk} denote the barycentric coordinates of agent l with respect to node $k \in \Theta_l$, computed via the Cayley-Menger determinant using only local inter-node distances.

It is useful for notation and analysis purposes to write DILOC in matrix form. By indexing the anchors in κ as $1, 2, \dots, \mathcal{N}_A$ and the agents in Ω as $\mathcal{N}_A + 1, \mathcal{N}_A + 2, \dots, \mathcal{N} + \mathcal{N}_A$ the following $(\mathcal{N} + \mathcal{N}_A) \times (\mathcal{N}_A)$ -dimensional coordinate matrix can be defined:

$$\mathbf{C} = [c_1^T, c_2^T, \dots, c_{\mathcal{N}+\mathcal{N}_A}^T]^T \quad (3.21)$$

Using the above notation, DILOC equations can then be written in compact matrix form

$$\mathbf{C}(t+1) = \mathbf{Y}\mathbf{C}(t) \quad (3.22)$$

where the $(\mathcal{N} + \mathcal{N}_A) \times (\mathcal{N} + \mathcal{N}_A)$ matrix \mathbf{Y} can be partitioned as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{I}_{\mathcal{N}_A} & \mathbf{0} \\ \mathbf{B} & \mathbf{P} \end{bmatrix} \quad (3.23)$$

The $(\mathcal{N}) \times (\mathcal{N}_A)$ block $\mathbf{B} = \{b_{lj}\}$ is a zero matrix, except for those entries corresponding to agents $l \in \Omega$ with a direct link to anchors. The $(\mathcal{N}) \times (\mathcal{N})$ matrix $\mathbf{P} = \{p_{lj}\}$ is also sparse; non-zero entries in row l correspond to the sensors in Θ_l . The matrices \mathbf{Y} , \mathbf{B} and \mathbf{P} have important properties that will be used in DILOC's extension to DLRE.

3.3.2 Extension of DILOC to DLRE

While it is apparent that DILOC operates distributively exploiting only local information, its application is limited to unrealistic noise-free scenarios. In practical scenarios, inter-node distances are subject to errors, communication

links between neighboring nodes may fail and, when alive, communication among nodes may be corrupted by noise. This section extends the DILOC algorithm to DLRE, in order to operate in the generic imperfect communication case.

For notational convenience, the update equations of DILOC are first written in terms of the columns $\mathbf{c}^j(t)$, $1 \leq j \leq \mathcal{N}_A - 1$ of the coordinate matrix $\mathbf{C}(t)$ as follows:

$$\mathbf{c}^j(t+1) = \mathbf{Y}\mathbf{c}^j(t), \quad 1 \leq j \leq \mathcal{N}_A - 1 \quad (3.24)$$

Based on the observation that the above update of $\mathbf{c}^j(t+1)$ does not involve the coordinates $\mathbf{c}^j(t)$ at time t , a relaxation parameter α is further introduced.

$$\mathbf{c}^j(t+1) = [(1-\alpha)\mathbf{I} + \alpha\mathbf{Y}]\mathbf{c}^j(t), \quad 1 \leq j \leq \mathcal{N}_A - 1 \quad (3.25)$$

If $\mathbf{c}^j(t)$ is partitioned as

$$\mathbf{c}^j(t) = \begin{bmatrix} \mathbf{u}^j \\ \mathbf{x}^j(t) \end{bmatrix} \quad (3.26)$$

where $\mathbf{u}^j \in \mathbf{R}^{\mathcal{N}_A \times 1}$ corresponds to the j -th coordinates of the anchors (hence the omitted time index) and $\mathbf{x}^j(t) \in \mathbf{R}^{\mathcal{N} \times 1}$ corresponds to the j -th coordinates of the agents at time t , then the state update (performed only on the agents whose positions are to be estimated) can be re-written equivalently as

$$\mathbf{x}^j(t+1) = [(1-\alpha)\mathbf{I} + \alpha\mathbf{P}]\mathbf{x}^j(t) + \alpha\mathbf{B}\mathbf{u}^j \quad (3.27)$$

where the system matrices \mathbf{B} and \mathbf{P} have been defined in the previous section.

In order to account for the partial imperfect information received by an agent at each iteration, the above recursion needs to be modified. In practice, there are several limitations:

- *Randomness in System Matrices:* Due to imperfect inter-node distance measurements, the system matrices \mathbf{B} and \mathbf{P} , needed by each agent in the update procedure, may in fact be random. Thus, since a single

measurement of the inter-node distances in the setup phase may lead to large random errors, it is assumed that a given agent estimates the required distances at each iteration of the algorithm. In other words, at each iteration, the l -th sensor can only obtain estimates of the corresponding rows of the \mathbf{P} and \mathbf{B} system matrices:

$$\begin{aligned}\hat{\mathbf{B}}(t) &= \mathbf{B} + \mathbf{S}_{\mathbf{B}} + \tilde{\mathbf{S}}_{\mathbf{B}} \\ \hat{\mathbf{P}}(t) &= \mathbf{P} + \mathbf{S}_{\mathbf{P}} + \tilde{\mathbf{S}}_{\mathbf{P}}\end{aligned}\quad (3.28)$$

where $\{\tilde{\mathbf{S}}_{\mathbf{B}}(t)\}_{t \geq 0}$, $\{\tilde{\mathbf{S}}_{\mathbf{P}}(t)\}_{t \geq 0}$ are independent sequences of random matrices with

$$\begin{aligned}E[\tilde{\mathbf{S}}_{\mathbf{B}}(t)] &= 0, \forall t, & \sup_{t \geq 0} E[\|\tilde{\mathbf{S}}_{\mathbf{B}}(t)\|^2] &= k_{\mathbf{B}} \geq \infty \\ E[\tilde{\mathbf{S}}_{\mathbf{P}}(t)] &= 0, \forall t, & \sup_{t \geq 0} E[\|\tilde{\mathbf{S}}_{\mathbf{P}}(t)\|^2] &= k_{\mathbf{P}} \geq \infty\end{aligned}\quad (3.29)$$

and $\mathbf{S}_{\mathbf{B}}, \mathbf{S}_{\mathbf{P}}$ are the mean measurement errors. Note that the above does not limit the noise model to additive; it only says that any random object may be written as the sum of the deterministic mean part and the corresponding zero-mean random part.

- *Random Link Failure:* It is assumed that each inter-node communication link may at times fail randomly. If two sensors l and n share a communication link, it is assumed that with probability $1 - q_{ln}$ the link fails. Therefore, each link is associated with a binary random variable $e_{ln}(t)$, such that $e_{ln}(t) = 1$ with corresponding probability q_{ln} indicates an active link, while $e_{ln}(t) = 0$ with probability $1 - q_{ln}$ indicates a link-failure.
- *Additive Channel Noise:* Given that a communication link (l, n) is active, sensor l may receive only a corrupt version of sensor n 's state, c_n^j given by:

$$y_{ln}^j(t) = c_n^j(t) + v_{ln}^j(t) \quad (3.30)$$

This models the communication-impairing channel noise.

Under the random environment model, presented above, the state update

procedure is modified, leading to the Distributed Localization in Random Environment (DLRE) algorithm:

$$\begin{aligned}
 x_l^j(t+1) &= (1 - \alpha(t))x_l^j(t) \\
 &+ \alpha(t) \left[\sum_{n \in \kappa \cap \Theta_l} \frac{e_{ln}(t) \hat{\mathbf{B}}_l n(t)}{q_{ln}} (u_n^j + v_{ln}^j(t)) \right] \\
 &+ \alpha(t) \left[\sum_{n \in \Omega \cap \Theta_l} \frac{e_{ln}(t) \hat{\mathbf{P}}_l n(t)}{q_{ln}} (x_n^j(t) + v_{ln}^j(t)) \right] \\
 l \in \Omega, 1 \leq j \leq \mathcal{N}_{(A)}
 \end{aligned} \tag{3.31}$$

Note that the relaxation parameter is now time varying. This is important for convergence analysis purposes and the reader is referred to [26] for more information.

3.3.3 Performance of the DILOC & DLRE algorithms

Both the DILOC and DLRE algorithm were tested on a two-dimensional topology of 49 nodes depicted in figure 3.8.

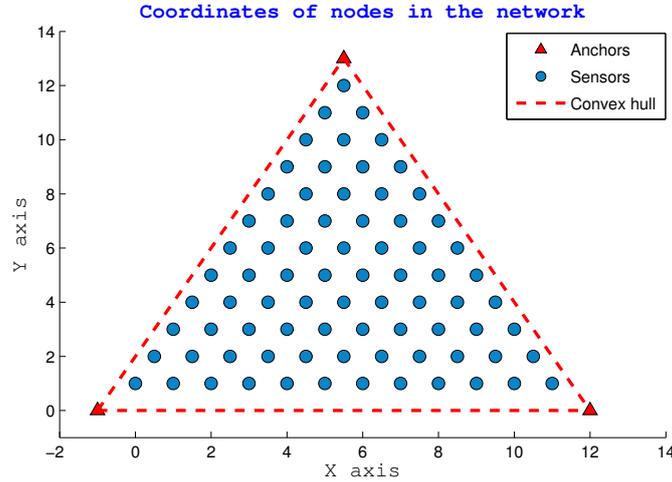


Figure 3.8: Toy 2D topology of 49 nodes to be localized through DILOC and DLRE. Note that all agents, depicted by blue "o"s, lie in the convex hull of the anchors, depicted by the red triangles.

DILOC is executed with no noise in the system. The communication radius is increased until triangulation is achieved and all nodes begin with zero initial conditions. Figure 3.9 shows the estimated coordinates of two arbitrary nodes; this illustrates the geometric convergence of DILOC to the exact sensor location. Observe that, even under a noise free channel, DILOC requires at least 100 iterations for the given setup.

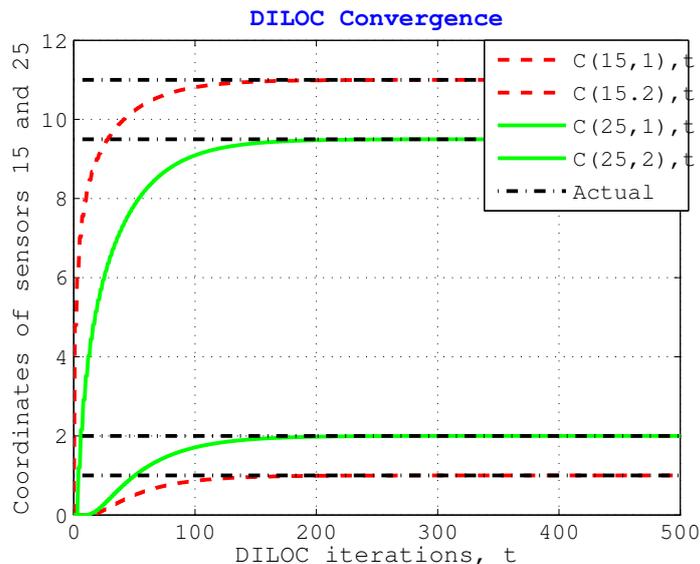


Figure 3.9: Convergence of DILOC estimates to exact sensor locations under a noiseless channel for the topology of figure (3.8). While convergence is guaranteed, it is also slow.

We now consider DLRE, simulated under the following scenario. Communication links are assumed to be active 90% of the time, i.e., $q_{ln} = 0.9 \forall l, n$, subject to $l \in \Omega$, $n \in \Theta_l$. Active links are hampered by additive communication noise that is Gaussian i.i.d with zero mean and unit variance. It is further assumed that the perturbation matrices, $\tilde{\mathbf{S}}_{\mathbf{B}}$ and $\tilde{\mathbf{S}}_{\mathbf{P}}$ are zero-mean Gaussian i.i.d with variance 0.1, since elements of \mathbf{B} and \mathbf{B} lie in the unit interval $[0, 1]$.

Figure 3.10 shows the estimated coordinates of two arbitrary nodes. DLRE's impressive performance is obvious. Even under the realistic noisy scenario, agents estimated location almost converge to the exact locations.

However, note the trade-off between accurate location estimation and bandwidth requirements. DLRE requires approximately 25000 iteration till convergence. Since each iteration requires nodes to exchange positioning information, DLRE requires ample bandwidth in order to localize.

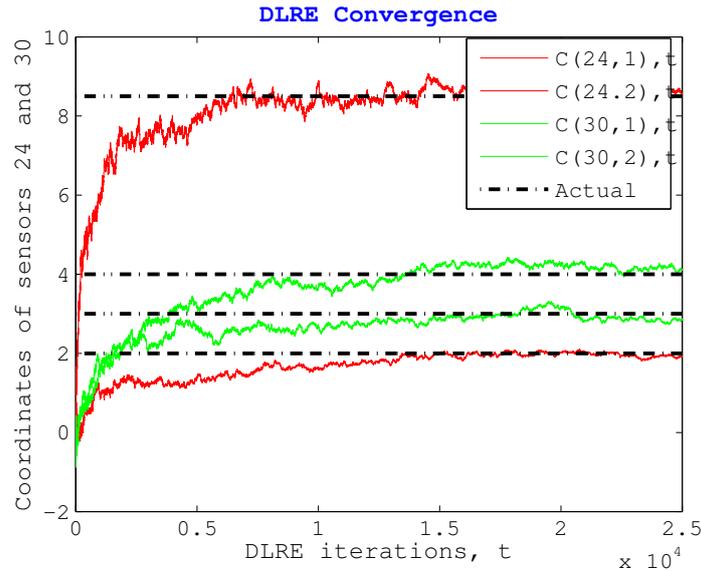


Figure 3.10: Convergence of DLRE estimates to exact sensor locations under a noisy channel for the topology of figure (3.8). Impressive performance is accompanied by heavy bandwidth requirements expressed as the total amount of real numbers exchanged.

The same conclusions can be drawn about the case of three-dimensional localization. Figure 3.11 shows a three-dimensional topology of nodes to be localized. As in the two-dimensional case, agent nodes lie in the convex hull formed by the anchors.

We omit simulating the DILOC algorithm and instead focus on the DLRE algorithm. Under the same setup as in the two-dimensional case, DLRE's performance is characterized by the same trade-off. Figure 3.12 shows a fraction of DLRE's iterations with the corresponding mean square error values. As seen, impressive performance comes with heavy bandwidth requirements.

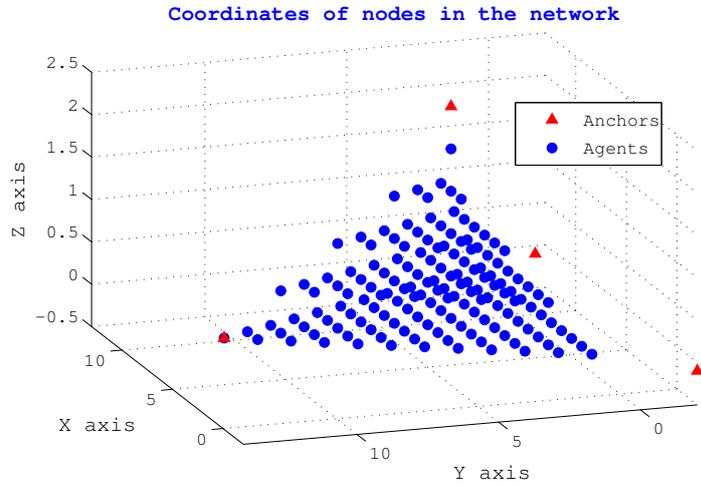


Figure 3.11: Three-dimensional topology for localization through DLRE. Agent nodes lie in the convex hull formed by the anchors.

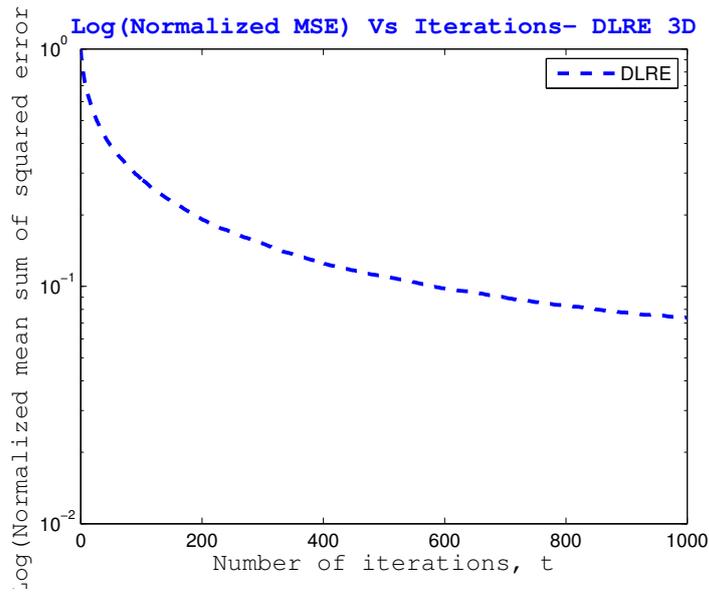


Figure 3.12: Mean square error as a function of the number of iterations for localization through DLRE for the topology of figure (3.11). The conclusions regarding two-dimensional localization can be extended to three-dimensional localization.

3.4 GPS-Free Node Localization

Many wireless sensor networks involve applications in which mobile nodes are required to move in a collaborative manner. Quite often a set of pre-existing anchors with globally known positions may not always be available or their availability may not be desired. Such is the case, in multiple swarming and rooting applications.

This section introduces a localization scheme [1] where each node must be aware of both its position and orientation *relative* to the network. The algorithm presented assumes no anchor nodes, and thus sensors are localized in an arbitrary coordinate system. Furthermore, as numerical results demonstrate, the GPS-Free node localization scheme is robust with respect to errors, is unaffected by the speed of the mobile nodes, and scales well as the network size increases.

The algorithm works under the following assumptions:

- Each node has a compass pointing North (or any other common reference direction)
- Nodes can measure the distance to their neighbors using a well known range measurement method (e.g. Time of Arrival (TOA))
- Motion actuators allow each node to move a specific distance in a specific direction (with respect to North)
- Actuator, compass and distance measurements are subject to errors caused by various real world disturbances
- Other than the above, no additional positioning equipment or infrastructure is required (i.e. anchor nodes are redundant)

Localization is performed in two steps. First, a *Core Localization* algorithm generates an ambiguous position estimate relative to a reference point, for each node in the network. Then, a *Verification* algorithm employs information generated from each node's neighborhood in order to resolve the aforementioned ambiguity. As a result, each node is unambiguously localized in a relative coordinate system.

3.4.1 Two stage positioning algorithm

The *Core Localization* algorithm operates on well defined time-slots, referred to as *rounds*, initiated by nodes whenever they require localization. Each node initiating the core localization process requires at least two neighbors to eventually localize unambiguously. Each round consists essentially of three steps:

(1) It begins with inter-distance measurements between neighbors, (2) it continues with individual, independent node movement, (3) it ends with a mutual exchange between nodes, of distance and angle measurements for that round.

As an example consider figure 3.13 which demonstrates the typical movement of two nodes, n_1 and n_2 , during an arbitrary round. At time t_0 , nodes n_1 and n_2 are located at positions (x_0, y_0) and (x_2, y_2) on a two-dimensional plane respectively, and measure their inter-node distance as d_1 . Between times t_0 and t_1 , each node $\{n_i\}_{i=1}^2$ covers a distance v_i in a direction α_i . At time t_1 both nodes, now at positions (x_1, y_1) and (x_3, y_3) exchange information concerning their movement (v_i, α_i) and measure their new inter-node distance d_2 , signaling the end of the round.

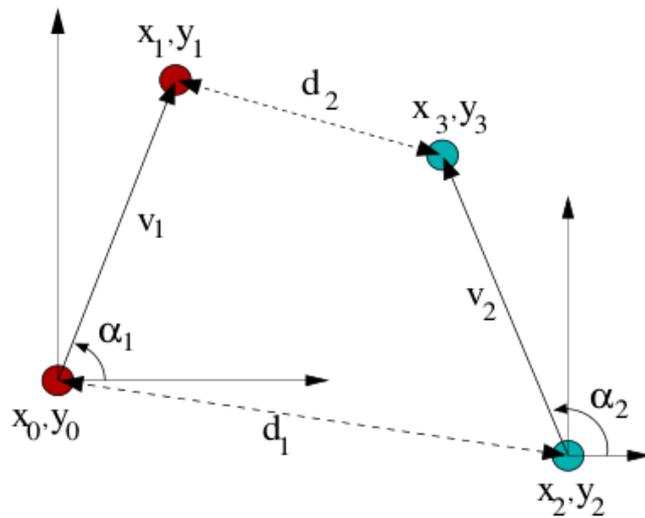


Figure 3.13: Typical movement of two neighbor nodes during a round for relative localization.

To achieve localization, each node selects itself as the origin and localizes its neighbors in its own local coordinate system. Continuing the example, to locate the position of node n_2 in the relative coordinate system of n_1 , the initial position (x_0, y_0) of n_1 is selected as the origin and as a result:

$$\begin{aligned} x_1 &= v_1 \cos \alpha_1, & y_1 &= v_1 \sin \alpha_1, \\ x_3 &= x_2 + v_2 \cos \alpha_2, & y_3 &= y_2 + v_2 \sin \alpha_2, \\ (x_3 - x_1)^2 + (y_3 - y_1)^2 &= d_2^2, & x_2^2 + y_2^2 &= d_1^2. \end{aligned} \quad (3.32)$$

Substituting the first and second pairs of equations in the third pair of equation yields:

$$x_2 A + y_2 B = C \quad (3.33)$$

where:

$$\begin{aligned} A &= v_2 \cos \alpha_2 - v_1 \cos \alpha_1, & B &= v_2 \sin \alpha_2 - v_1 \sin \alpha_1 \\ C &= \frac{1}{2} (d_2^2 - d_1^2 - v_1^2 - v_2^2 + 2v_1 v_2 \cos(\alpha_1 - \alpha_2)) \end{aligned} \quad (3.34)$$

Substituting:

$$x_2 = \frac{C - y_2 B}{A} \quad \text{and} \quad y_2 = \frac{C - x_2 A}{B} \quad (3.35)$$

into $x_2^2 + y_2^2 = d_1^2$, results in:

$$x_2^2 D - 2x_2 E + F = 0, \quad y_2^2 D - 2y_2 G + H = 0, \quad (3.36)$$

where D, E, F, G and H are defined as follows:

$$\begin{aligned} D &= A^2 + B^2, & E &= AC, & F &= C^2 - (d_1 B)^2 \\ G &= BC, & H &= C^2 - (d_1 A)^2 \end{aligned} \quad (3.37)$$

Using equation (3.36), each dimension of n_2 's new location solves independently to:

$$x_2 = \frac{E + \sqrt{E^2 - DF}}{D}, \quad y_2 = \frac{G + \sqrt{G^2 - DH}}{D} \quad (3.38)$$

and, as long as $D \neq 0$, solutions can be paired up using equation (3.33),

resulting in two distinct location estimates. Note that if $D = 0$, localization is impossible, and we have an *exceptional condition*, discussed in greater detail near the end of this section. The core localization algorithm is presented in detail below.

Algorithm 3.4.1: CORELOCALIZATION(n_1, n_2, v_1, α_1)

- (1) : $d_1 \leftarrow \text{inter-distance}(n_1, n_2)$
 - (2) : Move node n_1 by v_1 and α_1
 - (3) : $d_2 \leftarrow \text{inter-distance}(n_1, n_2)$
 - (4) : Retrieve v_2 and α_2 by n_2
 - (5) : Calculate positions of n_2 using (3.33)(3.36)(3.38)
-

After the core localization algorithm terminates for n_1 and n_2 , each node is left with two position estimates for its neighbor. Since only one position is valid (the other is due to symmetry), an additional *verification* step is required, in which both nodes diminish ambiguity by enlisting the aid of a common neighbor, n_3 .

The verification process is straight-forward using a third neighbor. After solving the equations of the core localization algorithm, node n_1 is left with two position estimates $\{n^{1,2}\}_{j=2}^3$ for each of its neighbors n_2 and n_3 . In order to unambiguously localize both neighbors simultaneously, node n_1 retrieves the inter-node distance measurement $d_{2,3}$ from either n_2 or n_3 and selects a pair of positions such that it better matches the received distance measurement.

Algorithm 3.4.2: VERIFICATION(*NeighborListNL*)

```

(1) : for each neighbor pair (m,n) in NL do
(2) :   if m and n are neighbors then
(3) :      $d_{m,n} \leftarrow$  inter-distance(m,n)
(4) :     for each pair  $\{m_i, n_j | i, j = 1, 2\}$  do
(5) :        $D \leftarrow$  Euclidean-distance( $m_i, n_j$ )
(6) :       if  $D - d_{m,n} = \min$  then
(7) :         Mark  $m_i, n_j$  as true positions
(8) :       end if
(9) :     end for
(10) :   end if
(11) : end for

```

Exceptional Conditions: The above two stage algorithm works well for rigid geometries. However, two problematic configurations exist, where the core localization algorithm may be unable to estimate two position estimates per node. These are, the *equal parallel movement* and the *excessive error* configurations.

Equal parallel movement, as the name suggests, occurs when two neighbors move in parallel during the same round. This is equivalent to $A = B = D = 0$. Since both nodes move in parallel with maintaining the same inter-distance, at the end of the round node n_2 lies anywhere on a circular ring centered around n_1 , and vice-versa ; the resulting position estimates are infinite.

The excessive error configuration occurs when highly erroneous d , v and α values create a non-rigid geometry, such that $E^2DF < 0$ or $G^2DH < 0$ leaving the core localization algorithm unable to estimate any positions.

Fortunately, both exceptional configurations may be detected in the core localization algorithm, and nodes can make necessary adjustments (e.g. random changes) to their speed and direction to avoid the same ill-configuration in the next round.

3.4.2 Performance of GPS-Free Node localization

The performance of GPS-Free node localization is tested for a topology of 100 nodes randomly arranged in a $100 \times 100m$ grid during a period of 100 rounds. Nodes move according to a random walk with random speed $[0, 5)$, random angle $[0, 2\pi)$ and a fixed radio range of $6m$. Simulation is performed under the following conditions: to simulate errors, uniform random noise is added to all measurements. For distance measures a percent error relative to the measured value is added, and for angle measures and absolute percent error (percent of 2) is added to the measured value.

Figure 3.14 depicts the average position error as a function of inaccuracies in both distance and angle measurements. The impressive performance of the algorithm is attributed mainly to the fact, that errors introduced in a round don't carry on to the next round, i.e, errors do not accumulate.

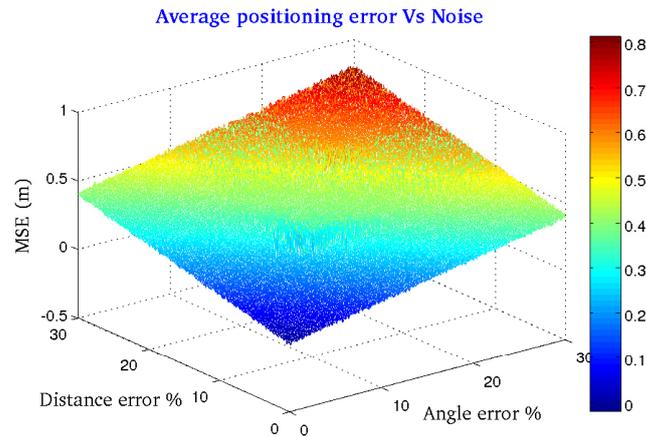


Figure 3.14: Average positioning error as a function of distance and angle measurements for GPS-Free relative node localization.

Figure 3.15 shows the number of nodes unlocalized due to errors in distance and angle measurements. By introducing errors, excessive configurations are more likely to occur. However, even with 30% errors on distance and angle measurements, the portion of unlocalized nodes is below 16%.

It is important to note, that the results presented previously are closely related to the network's dynamic density. For sparse networks, the lack

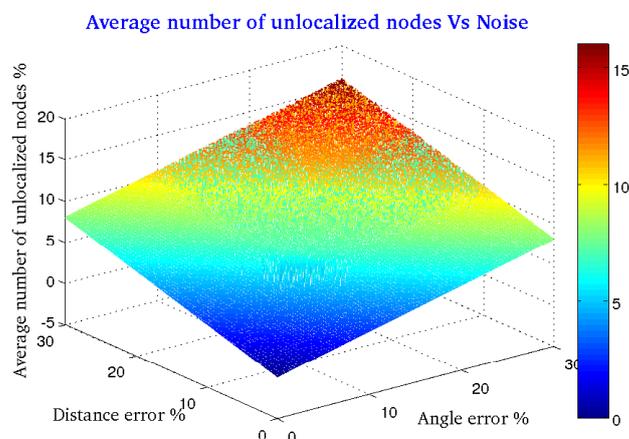


Figure 3.15: **Percent of nodes unlocalized as a function of distance and angle measurements for GPS-Free relative node localization. Observe that given the right density (collective movement) performance is impressive.**

sufficient connectivity brought about by independent node movement hinders the localization process. However, the GPS-Free node localization scheme's performance is excellent in collective movement cases. The interested reader is referred to [1] and references therein for more information on relative localization and collective movement schemes.

3.5 Chapter 3 Conclusions

This chapter presented a subset of the rich localization literature and included a performance evaluation of three distinct localization algorithms. Each algorithm's performance as far as bandwidth-efficiency and offered accuracy is concerned, was quantified under the same noisy scenario.

Distributed weighted Multi-Dimensional Scaling (dwMDS) performs well error-wise without unrealistic bandwidth requirements but may converge to a local minimum and requires some prior information.

The Distributed Localization In Random Environments (DLRE) algorithm offers impressive performance without the need for additional anchor nodes and accounts for various random scenarios including communication link-fails and imperfect node communication. However, while convergence is

guaranteed, it is slow and as a result the DLRE algorithm requires a significant amount of bandwidth for localization purposes.

To serve as an introduction to relative location estimation, the GPS-Free node localization two stage algorithm was tested. The algorithm performs robustly and scales well as the network size increases. Bandwidth requirements are minimal since each node is localized in one step. The mayor drawback of the GPS-Free node localization scheme is its heavy reliance of the network's dynamic density. As a result, its use is limited mainly to collective-movement applications.

The next chapter introduces a factor graph based localization algorithm as well as two variations. The offered algorithms are extensively simulated under identical scenarios and conclusions are drawn.

Chapter 4

The Sum Product Algorithm over Wireless Networks

The Sum Product Algorithm over Wireless Networks [47] is a factor-graph based cooperative location estimation algorithm. The algorithm aims to compute the position estimates of all the nodes consisting the network, based on available information to that time. Available information includes internal and relative, pair-wise measurements (see section 1.2.1) as well as information regarding the network's prior state.

As a first step, the relationship between all variables in the network is determined, then factorized and a factor-graph corresponding to that factorization is constructed. SPAWN then maps that factor graph onto the network topology by associating each node with its local information and develops the message passing scheme over the network factor graph (net-FG). As a result, each node may update its position estimate. based on the rules of the Sum Product Algorithm, implemented atop the network factor graph.

Due to the nature of the approach mentioned above, before going into detail it is useful to introduce the basic concepts of factor graphs and the sum product algorithm.

4.1 An introduction to Factor Graphs (FGs)

A factor graph [47?], is a way to graphically represent a factorization of a function or a distribution. For example, consider a hypothetical function

$$f(x_1, x_2, x_3)$$

and further assume that it can be factorized as follows:

$$f(x_1, x_2, x_3) = f_1(x_1)f_2(x_1, x_2)f_3(x_1, x_2)f_4(x_1, x_3)$$

where $x = \{x_1, x_2, x_3, x_4\}$ are referred to as *variables*, while $f = \{f_1, f_2, f_3, f_4\}$ are referred to as *factors*. The above factorization can be graphically represented by a factor-graph, as seen in figure [?]. It is important to note that factorization followed by a message passing scheme allows difficult functions to be handled.

Observe that the depicted factor-graph is cyclic. Unfortunately, this suggests that messages from the edges of the cycle will iteratively be transmitted within the cycle without an end. However, factorizations are by no means unique: by grouping f_2 and f_3 together, we result in a cycle-free (tree) graph. Due to the lack of cycles, tree graphs are appropriate for use in conjunction with message passing algorithms such as the sum product algorithm, introduced in the following subsection.

4.1.1 The Sum Product Algorithm (SPA)

The sum-product algorithm (or belief propagation) is a popular message passing algorithm operating on factor graphs. The SPA operates by computing messages inside the vertices and sending those messages over the edges. This subsection aims to describe the computational rules of the SPA; readers are referred to [47] and references therein for a more detailed treatment.

Let $\mu_{i \rightarrow j}(x_{ij})$ denote the message sent from node i to node j . Message computation is handled as follows:

- A message from a factor node f_i to a variable node x_i is given by:

$$\mu_{f_i \rightarrow x_i}(x_i) = \int f_i(x_0, x_1, \dots, x_i, \dots, x_n) \prod \mu_{x_k \rightarrow f_i}(x_i) \quad (4.1)$$

- A message from a variable node x_i to a factor node f_i is given by:

$$\mu_{x_i \rightarrow f_i}(x_i) = \prod \mu_{f_k \rightarrow x_i}(x_i) \quad (4.2)$$

- The marginal distribution of node x_i is given by

$$b(x_i) = \mu_{f_i \rightarrow x_i}(x_i) \mu_{x_i \rightarrow f_i}(x_i) \quad (4.3)$$

- The marginal of a cluster of variables is obtained by multiplying the incoming messages with the corresponding factor.

4.1.2 Factor Graphs and Sequential Estimation

Sequential estimation refers to the process of estimating dynamic variables, that is, variables that change over time [47]. A common approach is the estimation of variables at certain times, for example variable $x^{(t)}$ at time t , using a set of independent observation up to and including time t , for example $\mathbf{z}^{(1:t)} = [z^{(1)}, \dots, z^{(n)}]$. By relying on the following assumptions:

$$\begin{aligned} p(x^{(t)} | \mathbf{x}^{(1:(t-1))}) &= p(x^{(t)} | x^{(t-1)}) \\ p(z^{(t)} | \mathbf{x}^{(0:t)}) &= p(z^{(t)} | x^{(t)}) \end{aligned} \quad (4.4)$$

it can be shown that

$$\begin{aligned} p(x^{(t)} | \mathbf{z}^{(1:t)}) &= \int p(x^{(t)}, x^{(t-1)} | \mathbf{z}^{(1:t)}) dx^{(t-1)} \\ &\propto p(z^{(t)} | x^{(t)}, \mathbf{z}^{(1:t-1)}) p(x^{(t)} | \mathbf{z}^{(1:t-1)}) \end{aligned} \quad (4.5)$$

where

$$\begin{aligned} p(x^{(t)} | \mathbf{z}^{(1:t-1)}) &= \int p(x^{(t)}, x^{(t-1)} | \mathbf{z}^{(1:t-1)}) dx^{(t-1)} \\ &= \int p(x^{(t)} | x^{(t-1)}, \mathbf{z}^{(1:t-1)}) p(x^{(t-1)} | \mathbf{z}^{(1:t-1)}) dx^{(t-1)} \\ &= \int p(x^{(t)} | x^{(t-1)}) p(x^{(t-1)} | \mathbf{z}^{(1:t-1)}) dx^{(t-1)}, \end{aligned} \quad (4.6)$$

This allows us to determine the a posteriori distribution $p(x^{(t)} | \mathbf{z}^{(1:t)})$ in a two step process based on the following simple observation. Given all observations before time t , the distribution $p(x^{(t)} | \mathbf{z}^{(1:t-1)})$ can be determined through equation (4.6). This step is referred to as the *prediction operation* for obvious

reasons. Then, through the newly calculated distribution and by taking the new observation $z^{(t)}$ in to consideration, the distribution of interest can be calculated through equation (4.5), in a step referred to as the *correction operation*.

However, what is important is that sequential estimation can also be performed in an automated fashion equivalently by factoring the distribution $p(\mathbf{x}^{(0:T)}|\mathbf{z}^{(1:T)})$, where $\mathbf{x}^{(0:T)}$ and $\mathbf{z}^{(1:T)}$ denote the collection of all variables and observations up to time T , and by applying the sum product algorithm [47]. This important characteristic of sequential estimation will be used in the derivation of the SPAWN algorithm, presented in the following section.

4.2 Sum Product Algorithm over Wireless Networks (SPAWN)

4.2.1 Modeling Assumptions

Inline with the system model and the notation introduced in previous sections, time is considered slotted and the goal is to compute the position estimates $\mathbf{x}^{(0:T)}$ of all the nodes in the network, for all time slots t . Each node i is equipped in order to perform internal measurements at time t , denoted as $z_{i,self}^{(t)}$ and may perform pair-wise measurements with its neighbors, denoted as $z_{i,rel}^{(t)}$.

Work in [47] includes the following modeling assumptions. Let $\mathbf{x}^{(t_1:t_2)}$, $\mathbf{z}_{self}^{(t_1:t_2)}$ and $\mathbf{z}_{rel}^{(t_1:t_2)}$ denote the sequence of positions, available internal and available pair-wise measurements respectively, for all nodes from time t_1 to time t_2 . Then:

- The positions of nodes are a priori independent:

$$p(\mathbf{x}^{(0)}) = \prod_{i=1}^N p(\mathbf{x}_i^{(0)}) \quad (4.7)$$

- The movement of nodes follows a memoryless walk:

$$p(\mathbf{x}^{(0:T)}) = p(\mathbf{x}^{(0)}) \prod_{t=1}^T p(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) \quad (4.8)$$

- Nodes move independently:

$$p(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) = \prod_{i=1}^N p(\mathbf{x}_i^{(t)} | \mathbf{x}_i^{(t-1)}) \quad (4.9)$$

- Given the nodes position, relative measurements are independent of internal measurements:

$$p(\mathbf{z}_{\text{rel}}^{(1:T)} | \mathbf{x}^{(0:T)}, \mathbf{z}_{\text{self}}^{(1:T)}) = p(\mathbf{z}_{\text{rel}}^{(1:T)} | \mathbf{x}^{(0:T)}) \quad (4.10)$$

- Self measurements are mutually independent and depend only on the current and previous node positions:

$$p(\mathbf{z}_{\text{self}}^{(1:T)} | \mathbf{x}^{(0:T)}) = \prod_{t=1}^T p(\mathbf{z}_{\text{self}}^{(t)} | \mathbf{x}^{(t)}, \mathbf{x}^{(t-1)}) \quad (4.11)$$

- Given the nodes position, relative measurements are independent from time slot to time slot :

$$p(\mathbf{z}_{\text{rel}}^{(1:T)} | \mathbf{x}^{(0:T)}) = \prod_{t=1}^T p(\mathbf{z}_{\text{rel}}^{(t)} | \mathbf{x}^{(t)}) \quad (4.12)$$

- The relative measurements at any time slot t are conditionally independent:

$$p(\mathbf{z}_{\text{rel}}^{(t)} | \mathbf{x}^{(t)}) = \prod_{i=1}^N \prod_{j \in \mathcal{S}_{\rightarrow i}^t} p(z_{j \rightarrow i}^{(t)} | \mathbf{x}_j^{(t)}, \mathbf{x}_i^{(t)}) \quad (4.13)$$

Finally, the following assumptions, constituting the local information of any node i in the network at any time t , are made:

- the initial position distribution $p(\mathbf{x}_i^{(0)})$, is known,

- the mobility model at any time t , $p(\mathbf{x}_i^{(t)} | \mathbf{x}_i^{(t-1)})$, is known,
- the likelihood function, $p(\mathbf{z}_{i,self}^{(t)} | \mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})$, is known,
- the likelihood function, $p(\mathbf{z}_{i,rel}^{(t)} | \mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})$, is known,
- both internal and pairwise measurements at time t , $\mathbf{z}_{i,self}^{(t)}, \mathbf{z}_{i,rel}^{(t)}$, are known.

Under the aforementioned assumptions, the SPAWN algorithm is derived in the following sections. For more information regarding the modeling assumptions, the interested reader is referred to [47] and references therein.

4.2.2 Creation of Network Factor Graph

The first step in the derivation of the SPAWN framework is the factorization of the the relationship between all variables in the network, i.e., the posterior distribution $p(\mathbf{x}^{(0:T)} | \mathbf{z}^{(1:T)})$. Under the assumptions presented in the previous section and though application of Bayes rule:

$$\begin{aligned}
 p(\mathbf{x}^{(0:T)} | \mathbf{z}^{(1:T)}) &\propto p(\mathbf{x}^{(0:T)}, \mathbf{z}^{(1:T)}) \\
 &= p(\mathbf{x}^{(0:T)}, \mathbf{z}_{rel}^{(1:T)}, \mathbf{z}_{self}^{(1:T)}) \\
 &= p(\mathbf{z}_{rel}^{(1:T)} | \mathbf{x}^{(0:T)}, \mathbf{z}_{self}^{(1:T)}) p(\mathbf{x}^{(0:T)}, \mathbf{z}_{self}^{(1:T)}) \\
 &= p(\mathbf{z}_{rel}^{(1:T)} | \mathbf{x}^{(0:T)}) p(\mathbf{x}^{(0:T)}, \mathbf{z}_{self}^{(1:T)})
 \end{aligned} \tag{4.14}$$

Due to the assumptions of the memoryless walk, mutual independence of internal measurements and independence of relative measurements in sequential time slots:

$$\begin{aligned}
 p(\mathbf{z}_{rel}^{(1:T)} | \mathbf{x}^{(0:T)}) &p(\mathbf{x}^{(0:T)}, \mathbf{z}_{self}^{(1:T)}) = p(\mathbf{z}_{rel}^{(1:T)} | \mathbf{x}^{(0:T)}) p(\mathbf{z}_{self}^{(1:T)} | \mathbf{x}^{(0:T)}) p(\mathbf{x}^{(0:T)}) \\
 &= p(\mathbf{x}^{(0:T)}) \prod_{t=1}^T p(\mathbf{z}_{rel}^{(t)} | \mathbf{x}^{(t)}) p(\mathbf{z}_{self}^{(t)} | \mathbf{x}^{(t)}, \mathbf{x}^{(t-1)}) \\
 &= \prod_{t=1}^T p(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) p(\mathbf{z}_{rel}^{(t)} | \mathbf{x}^{(t)}) p(\mathbf{z}_{self}^{(t)} | \mathbf{x}^{(t)}, \mathbf{x}^{(t-1)}).
 \end{aligned} \tag{4.15}$$

factor graph (Net-FG). A natural choice is to perform the mapping based on information local to each node. For figure 1.1 this translates to mapping the vertices $\mathbf{h}_i^{(t-1)}(\mathbf{X}_i^{(t-1)}, \mathbf{X}_i^{(t)})$ to node i , as these vertices contain strictly local information. Similarly, in figure 4.1, for every variable $\mathbf{X}_i^{(t)}$, an equality vertex as well as a set of vertices labeled $\phi_{j \rightarrow i}$ exist, with the latter been functions of $z_{j \rightarrow i}^{(t)}$, i.e. information local to node i . Association of node i with all the aforementioned elements as a mapping choice results in a tree-subgraph. This is an extremely important result allows direct application of the SPA.

4.2.3 Prediction and Correction Operations

Before applying the SPA to the newly created local tree-subgraphs, the spatiotemporal constraints of the network need to be taken into consideration, i.e., a message schedule needs to be introduced:

Temporal constraints: To account for temporal constraints messages flow only forward, from past to present. This is indicated by the arrows present in figure ???. This leads to the first part of the SPAWN algorithm, presented below:

Algorithm 1 [47]

- 1: given $p(\mathbf{x}_i^{(0)})$, $\forall i$
- 2: **for** $t = 1$ to T **do**
- 3: \forall nodes $i = 1$ to N **in parallel**
- 4: prediction operation according to SPA rules

$$\begin{aligned} \mu_{h_i^{(t-1)} \rightarrow X_i^{(t)}}(\mathbf{x}_i^{(t)}) \propto & \int \underbrace{p(\mathbf{x}_i^{(t)} | \mathbf{x}_i^{(t-1)}) p(\mathbf{z}_{i,\text{self}}^{(t)} | \mathbf{x}_i^{(t-1)}, \mathbf{x}_i^{(t)})}_{=h_i^{(t-1)}(\mathbf{x}_i^{(t-1)}, \mathbf{x}_i^{(t)})} \\ & \times \mu_{X_i^{(t-1)} \rightarrow h_i^{(t-1)}}(\mathbf{x}_i^{(t-1)}) d\mathbf{x}_i^{(t-1)} \end{aligned}$$

- 5: **end parallel**
 - 6: Correction Operation Algorithm (Algorithm 2)
 - 7: **end loop**
-

Note the presence of both a *prediction* and *correction* operation, indicative of the fact that the location estimation problem, as formulated in this chapter, is indeed a sequential estimation problem. During the correction phase and in compliance with the SPA, node i computes the message $\mu_{h_i^{(t-1)} \rightarrow X_i^{(t)}}(\cdot)$ through message $\mu_{X_i^{(t-1)} \rightarrow h_i^{(t-1)}}(\cdot)$, the mobility model $p(x_i^{(t)} | x_i^{(t-1)})$ and the likelihood function $p(z_{i,self}^{(t)} | x_i^{(t)}, x_i^{(t-1)})$. Note that all information is local. On the other hand, during the correction operation node i relies on the exchange of information with neighboring nodes to compute the message $\mu_{X_i^{(t)} \rightarrow h_i^{(t)}}(\cdot)$ based on all the messages $\mu_{h_k^{(t-1)} \rightarrow X_k^{(t)}}(\cdot) \forall k$ as well as all the metrics $\mathbf{z}_{rel}^{(t)}$.

Spacial constraints: To account for spacial constraints, the message schedule is depicted in figure [?]. *Internode* messages, sent as packets over the wireless link, are denoted by the red arrows. *Intranode* messages, computed internally by each node, are denoted by the blue arrows. According to this schedule, messages flow in one direction over edges, are independent of the recipient node and thus can be broadcast. This leads to the SPAWN correction phase, presented bellow

Algorithm 2 [?]

-
- 1: **nodes** $i = 1$ to N **in parallel**
 - 2: initialize *belief* $b_{X_i^{(t)}}^{(0)}(\cdot) = \mu_{h_i^{(t-1)} \rightarrow X_i^{(t)}}(\cdot)$
 - 3: **end parallel**
 - 4: **for** $l = 1$ to N_{iter} **do**
 - 5: **nodes** $i = 1$ to N **in parallel**
 - 6: broadcast $b_{X_i^{(t)}}^{(l-1)}(\cdot)$
 - 7: receive $b_{X_i^{(t)}}^{(l-1)}(\cdot)$ from neighbors $\in \mathcal{S}_{j \rightarrow i}^{(t)}$
 - 8: convert $b_{X_i^{(t)}}^{(l-1)}(\cdot)$ to a distribution with respect to variable $\mathbf{X}_i^{(t)}$ using SPA rules

$$\mu_{g_{j \rightarrow i} \rightarrow X_i^{(t)}}(\mathbf{x}_i^{(t)}) \propto \int_{\mathbf{x}_j^{(t)}} p(z_{j \rightarrow i}^{(t)} | \mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) b_{X_j^{(t)}}^{(l-1)}(\mathbf{x}_j^{(t)}) d\mathbf{x}_j^{(t)}$$

9: Compute new message using SPA rules

$$b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)}) \propto \mu_{h_i^{(t-1)} \rightarrow X_i^{(t)}}(\mathbf{x}_i^{(t)}) \prod_{j \in \mathcal{S}_{j \rightarrow i}^{(t)}} \mu_{g_{j \rightarrow i} \rightarrow X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})$$

10: **end parallel**

11: **end loop**

12: **nodes** $i = 1$ to N **in parallel**

13: determine outgoing message: $\mu_{X_i^{(t)} \rightarrow h_i^{(t)}}(\cdot) = b_{X_i^{(t)}}^{(N_{iter})}(\cdot)$

14: **end parallel**

where a message broadcast by node i , at iteration l at time t , is denoted by $b_{X_i^{(t)}}^{(l)}(\cdot)$ and referred to as the *belief* of node i .

SPAWN is executed as a combination of both algorithms presented previously. At any given time, each node may obtain and estimate of its location by taking the mean or mode of its local message $\mu_{X_i^{(t)} \rightarrow h_i^{(t)}}$. Note that the algorithm is completely asynchronous; time slots need not be synchronized between different nodes.

4.3 Performance of SPAWN

This section provides and discusses some numerical results. For extensive simulations the reader is referred to Chapter 5. We consider a two-dimensional 50×50 networks of five nodes distributed as seen in figure 4.3. Location performance is evaluated in terms of offered accuracy, through the Mean Square Error (MSE) metric, as well as in terms of bandwidth consumption. As in previous simulations, bandwidth consumption is expressed as the size of the messages (i.e., amount of real numbers) exchanged between all nodes of the network. SPAWN is utilized with a grid resolution of $\delta = 0.5m$ under the presence of unimodal Gaussian noise of variance σ^r .

Figure 4.4 illustrates the performance of SPAWN as a function of the noise variance hindering the pair-wise measurements between communicating nodes. As the noise variance increases so does the MSE in the position estimation; large levels of noise variance prohibit the use of SPAWN.

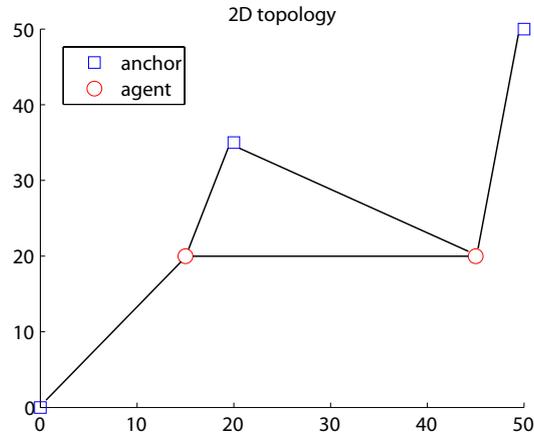


Figure 4.3: Network topology for two dimensional localization through SPAWN.

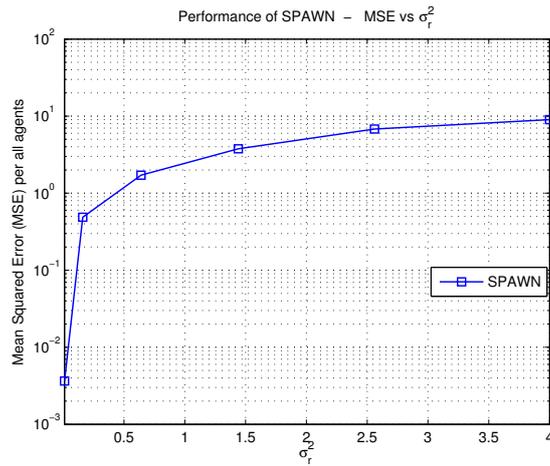


Figure 4.4: MSE as a function of the noise variance for SPAWN

SPAWN requires ample amount of bandwidth. This can be attributed to the fact that SPAWN requires the iterative exchange of distributions. The authors in [47] assumed a UWB infrastructure, ideal for SPAWN operations. However, in cases where bandwidth is constrained SPAWN can not be applied, as is evident in figure 4.5.

Finally, figure 4.6 plots the Cramer-Rao Bound for distributed, two-

dimensional, cooperative localization. As seen, the CRB curve follows the SPAWN MSE curve, corroborating the efficiency of the SPAWN framework. The interested reader will find the full derivation of the CRB in appendix I.

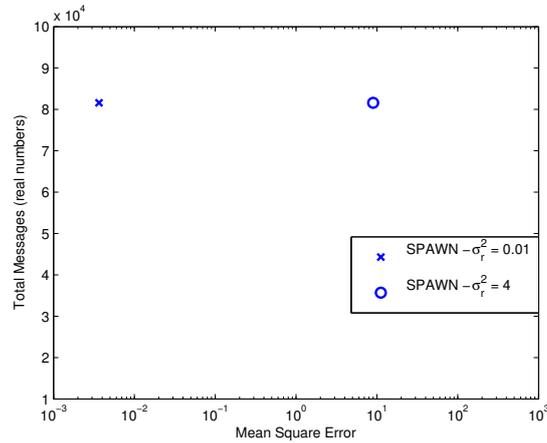


Figure 4.5: Total size of exchanged messages for small and large ranging error noise variance for SPAWN algorithm. Observe the prohibiting order of magnitude.

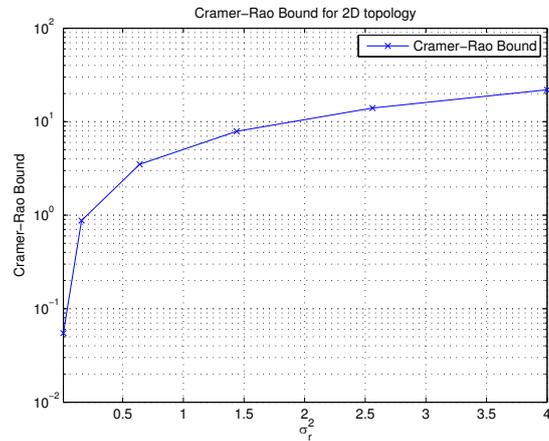


Figure 4.6: Cramer-Rao lower bound (CRB) for the topology of figure [4.3].

Chapter 5

Parameterized Sum Product Algorithm over Wireless Networks (P-SPAWN)

The previous Chapter revealed a very important result regarding localization through application of the Sum Product Algorithm over Wireless Networks. In the absence of ultra-wide-band radios, SPAWN's bandwidth requirements are quite heavy.

Specifically, the belief message $b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})$ of node i regarding its own location $(\mathbf{x}_i^{(t)})$, which is broadcasted at the beginning of iteration $l + 1$, maps $R^D \rightarrow R$ and thus, requires significant amount of communication bandwidth. For example, for 2D localization, area of interest $50m \times 50m$ and grid resolution of $0.5m \times 0.5m$ (i.e. grid resolution parameter $\delta = 0.5m$) at least 10^4 real numbers need to be transmitted per agent per iteration.

The goal of this chapter is to reduce the amount of total transmitted information during the localization process, without compromising localization accuracy. Towards that goal, a variant of the SPAWN algorithm is proposed. For 3D localization ($D = 3$), by exploitation of the independent depth/height measurement at each node, the inter-node measured distances are projected to a common 2D plane. Specifically, the depth/height unbiased, minimum mean squared error (MSE) estimate z_i of each node i is equivalent to its internal measurement, i.e. $z_i = \zeta_i$, according to the Gauss-Markov theorem. The range measurement between nodes i and j is projected on a common plane, according to $\tilde{d}_{ij}^p = \sqrt{\tilde{d}_{ij} - (\hat{z}_i - \hat{z}_j)^2}$, which is no longer Gaussian. The variant of SPAWN, namely Parameterized Sum Product Algorithm over Wireless Networks (P-SPAWN) is then applied. For 2D localization

($D = 2$), the algorithm is directly applicable.

5.1 Message representation

The P-SPAWN approach exploits the fact that the belief message $b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})$ of node i can be carefully expressed via a family of known parametric distributions, using a limited number of parameters. For example, if a node has a single neighbor, which is an anchor, then its belief will form a grommet (i.e. a ring) centered at the anchor's location, with diameters proportional to the measured distance, as well as the ranging error 5.1; the node can be located anywhere on the ring. Even though the possible locations (and their corresponding belief values) are numerous, such belief message can be directly modeled with only three parameters (center, range and range error variance) and thus, the belief to be broadcasted can be significantly compressed through the transmission of those parameters only.

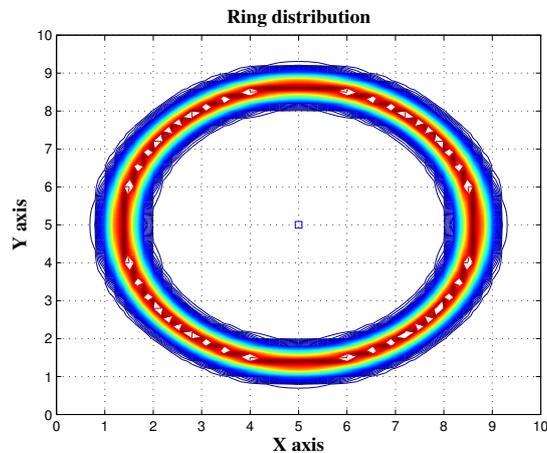


Figure 5.1: An agent measuring noisy distance from an anchor can be located anywhere on a grommet centered at the anchors location.

Therefore, the problem at hand can be viewed as a four step process, followed by each node i :

- successful identification of the appropriate belief type $b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})$,

- its expression to a limited number of parameters,
- broadcasting of those parameters to the neighbors (and not the belief itself),
- reconstruction of the belief at a receiving node through those parameters.

Inline with the computational rules of the Sum Product Algorithm (section 4.4.1), the belief message of node i at iteration l during time slot t is the multiplication outcome of factor graph messages:

$$b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)}) \propto \mu_{h_i^{(t-1)} \rightarrow X_i^{(t)}}(\mathbf{x}_i^{(t)}) \prod_{j \in \mathcal{S}_{j \rightarrow i}^{(t)}} \mu_{g_{j \rightarrow i} \rightarrow X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)}) \quad (5.1)$$

At the simplest case, messages $\mu_{g_{j \rightarrow i} \rightarrow X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})$ can be viewed as circular grommets centered at the neighbors estimated locations (note that these locations may be ambiguous i.e. numerous rings per message). In figure 5.2 for example, an agent receives beliefs from 3 anchors, obtaining as a result a unimodal belief after application of equation (5.1).

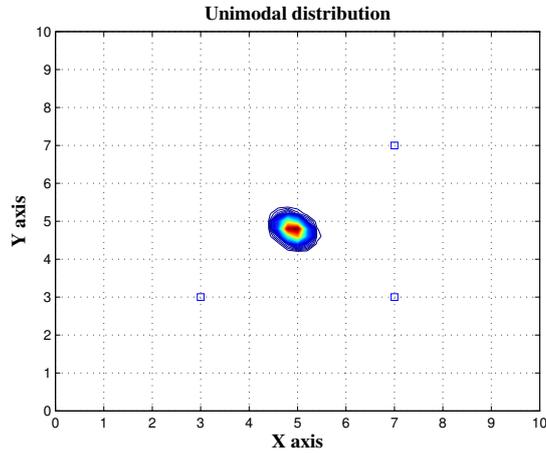


Figure 5.2: Communication with three anchors leads to a unimodal belief on the 2D plane.

The area of each grommet

$$r_{j \rightarrow i} = \{\mathbf{x}^t : \mu_{g_{j \rightarrow i} \rightarrow X_i^{(t)}}(\mathbf{x}_i^{(t)}) \neq 0\} \quad (5.2)$$

quantifies localization accuracy. For the specific example of figure 5.2, the neighbors are anchors and thus, the only source of uncertainty is the range measurement noise. However, if the neighbor is an agent, then there will be two sources of uncertainty: range measurement noise as well as ambiguity regarding the agents location 5.3. This leads to wider areas, which naturally indicate greater uncertainty; the set of possible locations is bigger.

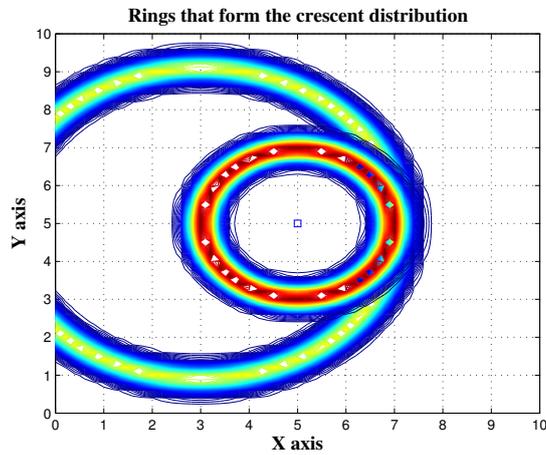


Figure 5.3: The area of received factor graph messages quantifies uncertainty.

Having determined how messages are represented, we are now ready to classify each possible type of resulting belief. Section 5.2 develops a simple classification scheme based on the number of neighbors as well as the type of messages their broadcast.

5.2 Classification of beliefs in a noiseless environment

In order to classify the types of resulting beliefs, we first a noiseless channel in which nodes are able to perform perfect measurements. In the ideal noiseless channel, the grommets of section (5.1) revert to perfect circles centered at the estimated (possibly ambiguous) locations on the transmitting nodes with a radius of δ_{ij} , were δ_{ij} denotes the true Euclidean distance between communicating nodes i and j .

Consider agent node i , updating its location estimate at time t , through iteration l of the SPAWN algorithm. Under the noiseless channel assumption, each message received by node i from neighbor $j \in S_{j \rightarrow i}$ during iteration l is associated with the following set:

$$\mathcal{C}_{j \rightarrow i}^{(l)}(\mathbf{x}_i^{(t)}) = \{\mathbf{x}_i \in R^2 : (x_i - x_j)^2 + (y_i - y_j)^2 = \sqrt{\delta_{ij}} \forall x_j, y_j \text{ such that } b_{X_j}^{(l-1)}(\mathbf{x}_j^{(t)}) > 0\} \quad (5.3)$$

It can easily be seen that the set $\mathcal{C}_{j \rightarrow i}^{(l)}$ is a collection of circles in R^2 . The cardinality of the set, expressed in terms of individual circles is equal to the number of modes in transmitter node j 's belief. For example, if j 's belief is unimodal, the cardinality of set $\mathcal{C}_{j \rightarrow i}^{(l)}$ is equal to one. A transmitter node j with a bimodal belief leads to a set $\mathcal{C}_{j \rightarrow i}^{(l)}$ of cardinality two. In the general case, if the belief of the transmitting node is K -modal, the cardinality of set $\mathcal{C}_{j \rightarrow i}^{(l)}$ is K .

Note that an exception to the above rule occurs when the distribution of the transmitting node is a circle. This leads to a set of infinite circles for $\mathcal{C}_{j \rightarrow i}^{(l)}$. It will be shown that, under the presence of noise, the resulting set $\mathcal{C}_{j \rightarrow i}^{(l)}$ contains only one ring of larger uncertainty (area). Thus, we consider that a circle distribution results in cardinality of one.

The above formulation allows us to express the set $\mathcal{C}_{j \rightarrow i}^{(l)}$ as an enumeration of the circles:

$$\mathcal{C}_{j \rightarrow i}^{(l)} = \{c_{j,1}, c_{j,2}, \dots, c_{j,K}\} \quad (5.4)$$

where K denotes the modality of $b_{X_j^{(t)}}^{(l-1)}$ and $c_{j,m}$, $m = 1, \dots, K$ denotes the circles corresponding to each mode.

As a result, the non-zero points of message $\mu_{g_{j \rightarrow i} \rightarrow X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})$ received by node i from node j are given on the two-dimensional plane, in terms of sets, by the union of the collections of circles in $\mathcal{C}_{j \rightarrow i}^{(l)}$

$$\mu_{g_{j \rightarrow i} \rightarrow X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})_{2D} = \{\mathbf{x}_i^{(t)} \in \bigcup_{m=1}^K c_{m,j} \subset R^2\} \quad (5.5)$$

where the subscript $2D$ indicates that we are interested only on the points at which the message is non-zero (i.e., potential locations) and not on the quality of estimation.

In a similar fashion, in terms of sets, the correction operation of SPAWN given by equation (5.1) can be expressed as the union of the intersections of the sets of $\mathcal{C}_{j \rightarrow i}^{(l)}$ for all nodes $j = 1, \dots, J$ with a communication link to node i :

$$b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})_{2D} = \{\mathbf{x}_i^{(t)} \in \left(\bigcup_{m=1}^K c_{m,1}\right) \cap \left(\bigcup_{m=1}^K c_{m,2}\right) \cap \dots \cap \left(\bigcup_{m=1}^K c_{m,J}\right) \subset R^2\} \quad (5.6)$$

Since the intersections of unions is the union of intersections (the intersection operation is distributive over the union operation) we can now easily determine the maximum possible modality of the resulting belief of node i at iteration l of time slot t . The distribution is given by the non-overlapping union of intersection operations, and thus the maximum number of modes is given by

$$K_{max} = \frac{N!}{(N-2)!} \quad (5.7)$$

where

$$N = \sum_{d=1}^J \sum_{k=1}^{K_d} m_k j_d \quad (5.8)$$

K_d denotes the cardinality of the set $\mathcal{C}_{j \rightarrow i}^{(l)}$ for $j = d$ and J is the cardinality of the set $S_{j \rightarrow i}$, i.e., the set of all neighbors of node i .

It is important to note that K_{max} determines the maximum possible

modality and will, in all likelihood, be different than the actual modality. We can however, determine the actual number of modes through a simple clustering algorithm, such as *K-means*, by clustering $b_{X_i^{(t)}}^{(l-1)} \left(\mathbf{x}_i^{(t)} \right)_{2D}$ for K_{max} clusters and discarding invalid results by taking the point-wise multiplication with the original belief. Clustering efficiently requires an indication regarding the number of clusters so, in a best effort attempt, the maximum number of modes needed to be determined a-priori.

Summarizing, the modality of the new belief of node i , $b_{X_i^{(t)}}^{(l)}$ can be determined in the following steps:

1. Given a set of J neighbors, each with K_j -modal distributions, K_{max} is computed.
2. If node i has only one neighbor with a circle distribution, i.e.:

$$\left(\sum_{n=1}^N - 2 \right) = 0 \quad (5.9)$$

the process is terminated this the distribution of i 's belief has been determined.

3. A set of clusters, C , is determined through application of the K-means algorithm and placed on the topology.
4. Invalid clusters are discarded through point-wise multiplication with the positions in which the original belief is non-zero (potential locations of node i).

$$C^* = C \cdot b_{X_i^{(t)}}^{(l)} \left(\mathbf{x}_i^{(t)} \right)_{2D} \quad (5.10)$$

C^* now contains all valid clusters.

The process then continues in subsequent iterations. The important result is that, given the number of clusters, we have already determined a message parameterization of node i 's new belief. Specifically, each mode (i.e cluster) of node i 's belief, can be expressed by its coordinates on the two dimensional plane. This information is sufficient for constructing node i 's belief without the need for the whole distribution to be broadcasted in the next iteration.

The exceptional case of the circle distribution is handled in the following section where, under a noisy channel, it is shown it results in a ring of larger proportions, expressed by 4 numbers: two-dimensional center, radius and an uncertainty parameter.

5.3 The P-SPAWN algorithm

P-SPAWN operates under practical, realistic environments where communication is impaired by noise, by generalizing the results of the previous section. In the ideal case, messages were a collection of circles. Under noise, messages are a collection of grommets. While the same principles apply, we note the following distinctions:

- Discovered clusters are associated with two-dimensional Gaussian distributions with mean equal to the cluster's coordinates and variance corresponding to uncertainties due to noise (see figure 5.2)
- Grommets are characterized by their width (i.e., uncertainty 5.1). Different nodes have different levels of uncertainty. Therefore, during communication of node i with its neighbors the different levels of uncertainty and their contribution to the resulting belief of node i need to be taken into account.
- When the belief broadcasted is a ring, through line 8 of the SPAWN algorithm, the receiving end obtains a ring of much greater proportions. This quantifies the level of uncertainty caused by the infinite possible locations on the ring's periphery. Using the approach of sets, it is evident that since each of the infinite nodes on the ring corresponds to a ring at the receiving side, their union blends in a larger area. Fortunately this observation allows direct computation of the width of the new "fatter" ring.
- Due to uncertainties, the intersection of two rings will never result in a single point as in the ideal case. This has been taken into consideration while discussing the clusters. However an exceptional case may occur,

as is evident in the case of neighbors lying on a hyperplane. Figure 5.3 depicts this exceptional case. In order to better model the resulting belief, referred to as a *crescent* distribution, clustering will include an additional step to detect this case.

Having addressed the aforementioned distinctions, we are ready to develop a heuristic scheme in which nodes identify their belief type and broadcast only the parameters needed for reconstruction by the receiving end. The following cases may occur:

- Case 1: Agent node i has no neighbors; its belief will be uniform over the area of interest and will not be broadcasted in subsequent iterations (it contains no useful information).
- Case 2: Agent node i has one neighbor j . The following possibilities exist:
 - Case 2.1: Node j has a unimodal belief: Node i 's belief will be of ring-type.
 - Case 2.2: Node j has a ring belief: Node i 's belief will be of ring-type.
 - Case 2.3: Node j has a crescent belief: Node i 's belief can be sufficiently approximated as a ring-type distribution.
 - Case 2.4: Node j has a K -modal belief: Node i 's belief is the union of the collection of rings corresponding to each mode of node j . We characterize the resulting belief as K -modal (see subsection 5.2).
- Case 3: Node i has at least two neighbors; In order to determine node i 's distribution we follow the automated clustering procedure described in subsection 5.2 to determine the mode of node i 's belief.

Observe that cases 1 through 2.4 do not require computation of K_{max} . This allows computationally efficient clustering. This very process is executed by the *BeliefClassification* algorithm, presented below.

The algorithm at node i receives as input SPAWN belief message $b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})$ at iteration l and outputs its estimated type parameters, that will be transmitted instead.

BeliefClassification $\left(b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})_{2D}, S_{j \rightarrow i}, J, \sigma_{b_i}^2 \right)$ [49]

- 1: Given cardinality of set $\mathcal{C}_{j \rightarrow i}^{(l)} \forall j \in S_{j \rightarrow i}, \mathbf{K}_j$
- 2: **if** $(J == 1) \ \&\& \ (K_j == 1)$ **then**
- 4: parameters \leftarrow *CompressBelief*("ring", Null, $\sigma_{b_i}^2$)
- 5: $\sigma_{\mu_i}^2 \leftarrow g(\sigma_r^2, r_i, \sigma_{b_j}^2)$
- 6: **elseif** $(J == 1) \ \&\& \ (K_j > 1)$ **then**
- 7: **for** $m = 1$ to K_j with step 3 **do**
- 8: parameters[$m : m + 3$] \leftarrow *CompressBelief*("ring", Null, $\sigma_{b_i}^2$)
- 9: **end loop**
- 10: $\sigma_{\mu_i}^2 \leftarrow g_1(\sigma_r^2, r_i, \sigma_{b_j}^2)$
- 11: **else** (i.e. $J > 1 \ \&\& \ K_j > 0$)
- 12: Compute maximum estimated modality through equation (5.7)

$$K_{max} = \frac{N!}{(N-2)!}$$

- 13: Cluster the belief for K_{max}

$$C \leftarrow \text{Kmeans} \left(b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})_{2D}, K_{max} \right)$$

- 14: Discard invalid clusters by (5.10)

$$C^* = C \cdot b_{X_i^{(t)}}^{(l)}(\mathbf{x}_i^{(t)})_{2D}$$

- 15: Trim redundant clusters: $C^{**} \leftarrow \text{trim}(C^*)$
 - 16: **for** $m = 1$ to $\text{size}(C^{**})$ with step 2 **do**
 - 17: parameters[$m : m + 2$] \leftarrow *CompressBelief*("K", $C^{**}(m, m + 1)$, $\sigma_{b_i}^2$)
 - 18: **end loop**
 - 19: $\sigma_{\mu_i}^2 \leftarrow g_1(\sigma_r^2, r_i, \sigma_{b_j}^2)$
 - 20: **endif**
-

Function `BeliefClassification(.)` outputs parameters according to the input string. For the case of a ring distributions the parameters include the two dimensional ring center, the ring radius as well as a uncertainty parameter, to be discussed further down.

It is noted that an automated way to locate the ring center of a ring-type belief stems from the selection of three $[xy]^T$ points on the $2D$ plane that maximize the belief on the periphery of the ring; creating two line segments that connect a pair of those points and taking the perpendicular bisector of each line segment, offer an estimate of the ring center as the intersection of the two bisectors. The ring radius is proportional to the measured range.

K -modal distributions ($K \geq 1$) are modeled through function `BeliefClassification` by a two-dimensional mean (cluster) per mode and an uncertainty parameter. The process of trimming, performed by the trim function, refers to discarding clusters the are close to each other and substituting them with one cluster. This is a necessary step due to over-clustering performed through the K_{max} parameter.

For input string "K" the `BeliefClassification` function also classifies crescent type distributions through comparison of cluster-heads inter-distance. This is performed on the basis that clusters that are not close enough to be grouped (see trim function) in to a single cluster, may also lead to accuracy loss if grouped as two individual clusters under certain conditions (see figure 5.4). In this case, the combination of clusters leads to a crescent distribution, and is modeled as such (Case 2.3). As a rule-of-thumb, this occurs when the cluster inter-distance is between two-to-six times the standard deviation associated with the belief variance uncertainty parameter, discussed further bellow.

P-SPAWN carefully quantifies localization uncertainty using two variables: σ_b^2 and σ_μ^2 . Variable σ_b^2 denotes the localization uncertainty that should be modeled at the parameters of each node's belief. For example, in the unimodal case of figure 5.2 that variable determines the variance of the underlying Gaussian. Variable σ_μ^2 quantifies the uncertainty in the location of node i given the broadcasted belief (and corresponding uncertainty) of neighboring node j . Specifically, σ_μ^2 resulted from the belief broadcasted

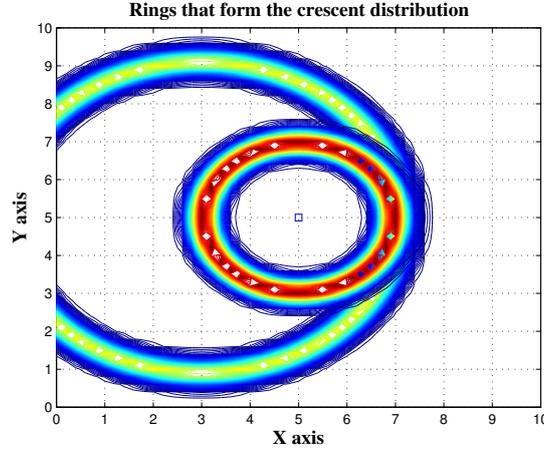


Figure 5.4: The resulting belief is not modeled adequately neither as a single nor a pair of clusters.

from node j to agent i is given by:

$$\sigma_{\mu_{j \rightarrow i}} = g_1(\sigma_r^2, r_i, \sigma_{b_j}^2) = \sigma_r + \sigma_{b_j}^2 \quad (5.11)$$

where $\sigma_{b_j}^2$ is one of the estimated parameters that model belief of node j . When node j is an agent with a ring distribution, the formula reverts to:

$$\sigma_{\mu_{j \rightarrow i}} = g(\sigma_r^2, r_i, \sigma_{b_j}^2) = \frac{2}{3} * r_i + \sigma_r + \sigma_{b_j}^2 \quad (5.12)$$

so that the additional localization uncertainty is taken into account. The results are obtained through careful computation of the width of messages, viewed as ring distributions corresponding to each potential point of the belief, as discussed in previous sections.

The estimated belief variance of node i is given by:

$$\sigma_{b_i}^2 = \min_{j \in S_{j \rightarrow i}} \sigma_{\mu_{j \rightarrow i}} \quad (5.13)$$

since belief of i is the product of the factor graph messages and thus, the one with the smallest area (and thus uncertainty) will determine the area of the produced belief.

In the numerical results section it will be shown that P-SPAWN, as well as a SPAWN variation based on truncation provide a flexible and efficient tradeoff between bandwidth reduction and localization accuracy, even with unimodal, non-Gaussian ranging error distribution.

5.4 Performance of P-SPAWN algorithm

Numerical results are provided for both 2D as well as 3D localization. The $[x \ y \ z]^T$ coordinates and the connectivity of each node are given at Fig. 4.3, where it is shown that each agent is connected with only two anchors. Even for 2D localization, connectivity with only two anchors provides ambiguity between two possible locations and thus, additional information is needed. However, each agent is also connected to another agent (of unknown location) and thus, cooperation among the two provides that necessary extra information. Notice that in classic LBL systems, only two anchors are utilized for a given area. We first describe the results for 2D localization, assuming that all nodes are placed on a common plane, according to their x and y coordinates. 3D localization results follow, alongside with the Cramer-Rao bound (CRB) for the examined problem. The reported mean squared error (MSE) has been calculated after 150 experiments (with the exception of BTB schemes, which have been tested for a larger number of experiments equal to 1000, given their computational simplicity) per reported noise variance. K-SPAWN has utilized $K = 27$ and only non-zero belief value reporting (i.e. a zero belief value is not transmitted).

Fig. 5.6 offers the mean square error (MSE) calculated across all agents, as a function of the ranging noise variance σ_r^2 , for 2D localization. It is shown that the proposed bandwidth versions K-SPAWN and P-SPAWN offer MSE results similar to the original SPAWN, even though they utilize exchanged messages of significantly reduced size (to be quantified and discussed subsequently). Furthermore, all three algorithms above, computed for a grid resolution of $\delta = 0.5$ m, outperform the distributed version of MDS (dwMDS); the latter requires some prior knowledge regarding the initial location of the agents; such knowledge was not available for any of the above

algorithms above.

Fig. 5.7 offers the size of the exchanged messages among all nodes (anchors and agents) and the corresponding MSE across all agents. The size of the messages is expressed as the total real numbers exchanged in the network (e.g. if a message broadcasted at iteration l from agent i consists of n real numbers, it will increase the total number by n). For the classic SPAWN, the considered topology and grid resolution of $\delta = 0.5$ m, the total real numbers required until final localization was on the order of $80K$ numbers. Fig. 5.7 shows that K-SPAWN reduces the total message size to ~ 180 real numbers (two orders of magnitude reduction), while P-SPAWN reduces total numbers to ~ 20 (three-orders of magnitude). There is also an interesting tradeoff between localization accuracy and respective size (depicted by an arrow), since one can sacrifice MSE and choose P-SPAWN instead of K-SPAWN, so that smaller messages are utilized and thus, localization can be performed faster in bandwidth-limited environments (e.g. those with acoustic modems). For larger ranging error noise variance, all algorithms perform poorly with similar MSE but vastly different message size requirements. K- and P-SPAWN will be also shown to perform relatively well below, in non-Gaussian (but unimodal) ranging error environments.

Fig. 5.8 shows that performance of BTB schemes. Specifically, K-, P- or classic SPAWN was first run with a "rough" grid resolution of $\delta = 5$ m (i.e. one order of magnitude smaller than above). Such grid resolution allowed faster calculation of all the double integrals (for 2D) of SPAWN (and its variants) compared to $\delta = 0.5$ and offered an initial estimate of all agent's location. Next, dwMDS was executed, using the outcome of SPAWN or P-SPAWN or K-SPAWN as prior knowledge. It is shown that BTB schemes reduce the MSE and thus, improve localization accuracy compared to prior art. If K-SPAWN or P-SPAWN is used in conjunction with dwMDS, computation- and bandwidth-friendly localization is possible, since initial estimate of agents' location is offered with smaller size of exchanged messages, compared to classic SPAWN. On the other hand, if classic SPAWN is utilized instead, localization can be further improved at the expense of increased message size. Such tradeoff is shown at Fig. 5.9 where it can be seen

that the total number of exchanged of BTB-SPAWN is one order of magnitude larger than BTB-P-SPAWN or BTB-K-SPAWN. Notice again that under large ranging error noise variance, all algorithms achieve the same (relatively) large MSE. Discussion regarding MSE limits (of BTB or other schemes) is offered at the end of this section.

Fig. 5.10 shows the sensitivity of SPAWN and its variants on grid resolution δ . Since the location of anchor nodes and the dimensions of the examined topology are a priori known (or can be approximated), a good balance is achieved by setting the grid resolution equal to the greatest common divisor (GCD) of all anchors' non-zero coordinates. In Fig. 5.10 it is shown that when parameter $\delta = 5$ (i.e the GCD of all nodes' non-zero coordinates for the given topology), MSE is smaller than that for $\delta = 4$; the latter value of δ , even though smaller, results to inaccurate anchor location approximation before SPAWN (or its variants) is executed, resulting to higher MSE.

For 3D localization, the projection of range measurements between any two nodes on a common plane results to a non-Gaussian random variable. Fig. 5.11 depicts localization MSE for all agents, for depth/height measurement noise variance $\sigma_z^2 = 1$ and various range noise variance σ_r^2 values. It can be seen that P-SPAWN (as well as K-SPAWN) offers similar MSE results with classic SPAWN, even though it operates on non-Gaussian ranging noise. Such result can be explained by the fact that range measurements are distributed according to a unimodal distribution. Furthermore, BTB-SPAWN with grid resolution parameter $\delta = 5$ achieves the smallest MSE among the tested algorithms (SPAWN, P-SPAWN and K-SPAWN are tested with $\delta = 0.5$ and BTB-SPAWN with $\delta = 4$ performs inferiorly, for reasons already explained above). The bandwidth-friendly versions of SPAWN reduce the required number of exchanged real numbers one order (K-SPAWN) or two orders (P-SPAWN) of magnitude compared to (BTB-SPAWN), which however achieves the smallest MSE (Fig. 5.9). It is further noted that classic SPAWN in 3D localization would need at least one order of magnitude larger amount of exchanged real numbers, while attaining larger MSE, compared to its BTB counterpart. For stronger range noise variance, all algorithms offer similar, relatively high MSE with different bandwidth requirements.

Finally, the Cramer-Rao bound (CRB) is depicted in Fig. ??, i.e. the MSE lower bound achieved by an *unbiased* estimator of the unknown coordinates across all agents. The CRB has been calculated at the appendix for the 3D long-baseline (LBL) setup of this work, assuming independent noise at both ranging and depth/height measurements. Fig. ?? shows that the MSE bound follows the same trend with the numerical results of the proposed algorithms, i.e. increased noise variance, either at depth/height or range measurement increases the achieved MSE. It is also observed that the achieved MSE for the BTB cases is *smaller* than the above CRB bound. That can be explained by the fact that a) the reported MSE of all algorithms is offered after a bounded number of experiments and thus itself is an estimate (with associated variance) and b) the SPAWN-based algorithms (including the classic SPAWN) offer estimated variance of localization error that does not coincide with estimated MSE, i.e. the offered estimator is *biased*; the estimated belief messages are approximations of the posterior pdf of each agents' true location, providing non-zero average localization error (and thus, biased estimators). However, the calculated and reported CRB values clearly offer an order of magnitude for the tested algorithms' MSE.

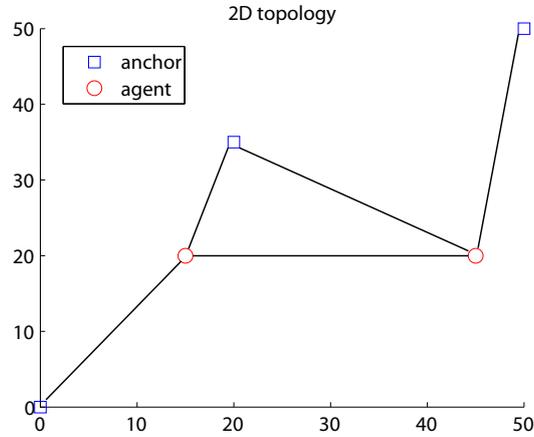


Figure 5.5: **Test topology and node connectivity:** three anchors are placed at $[0 \ 0 \ 0]^T$, $[20 \ 35 \ 0]^T$, $[50 \ 50 \ 0]^T$ and two agents at $[15 \ 20 \ 20]^T$, $[45 \ 20 \ 30]^T$, respectively. Notice that each agent can communicate with only two anchors and another agent. For 2D experiments, nodes are assumed on a common plane according to their x and y coordinates.

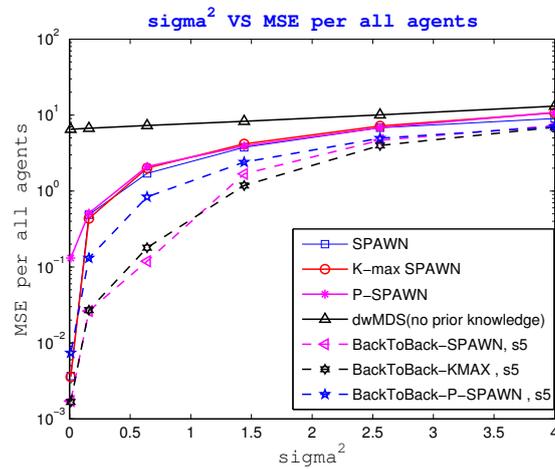


Figure 5.6: Mean squared error (MSE) as a function of ranging noise variance σ_r^2 for 2D localization. All algorithms operate without agents location prior knowledge..

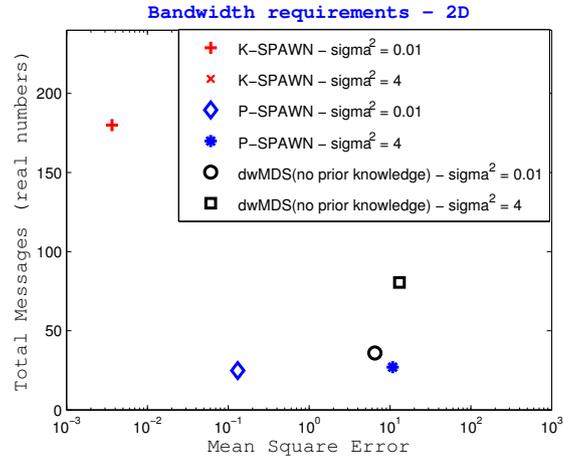


Figure 5.7: K-SPAWN and P-SPAWN offer the best tradeoff between MSE and size of exchanged messages (expressed in total real numbers) and reduces total number of real numbers exchanged by two to three orders of magnitude, compared to classic SPAWN. For large ranging error variance, all algorithms achieve similar MSE with vastly different message size requirements.

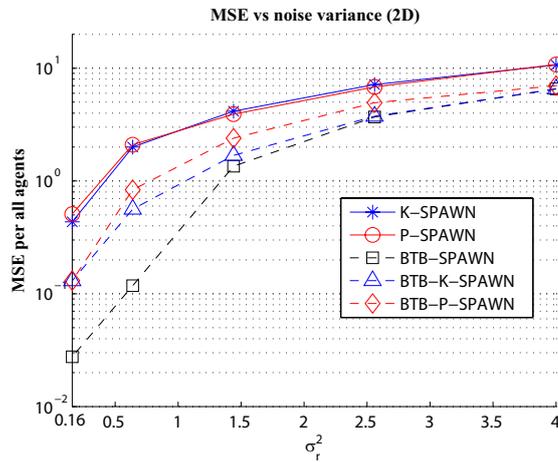


Figure 5.8: Back-to-back (BTB) schemes can outperform state-of-the-art cooperative localization.

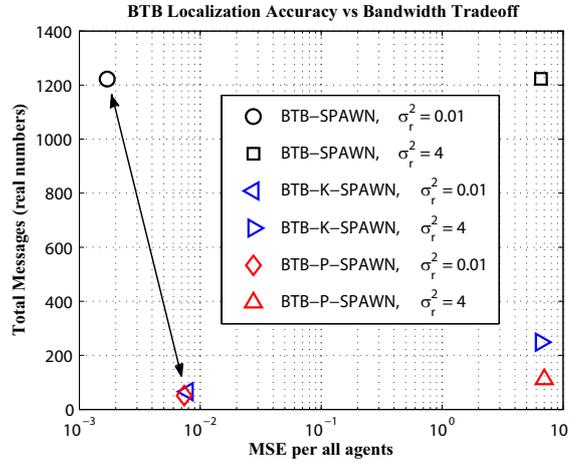


Figure 5.9: BTB-K- and BTB-P-SPAWN require one-of-order of magnitude smaller exchanged messages (in total real numbers) compared to BTB-SPAWN. The latter achieves the smallest MSE. All BTB algorithms achieve similar MSE in relatively large ranging error variance.

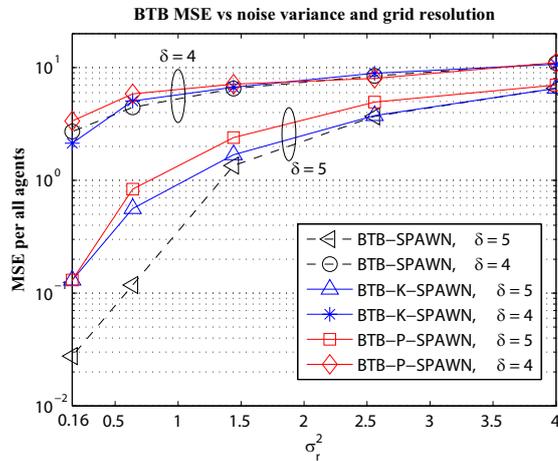


Figure 5.10: Approximating the 2D plane with a grid of area δ^2 regions is necessary for numerical calculations. Parameter $\delta = 5$ m offers better MSE results than $\delta = 4$ m, since the latter value, even though smaller, is not the greatest common divisor of all anchor's non-zero coordinates.

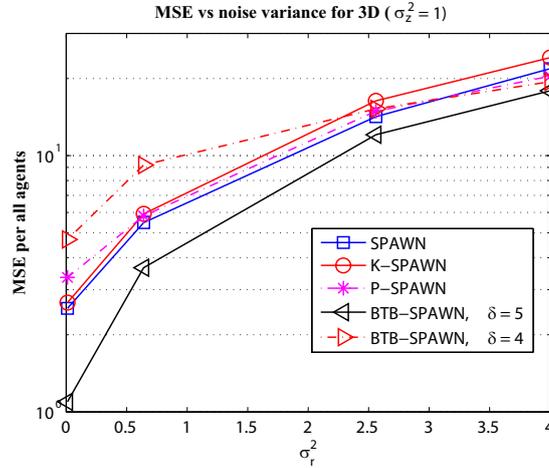


Figure 5.11: MSE across all agents as a function of ranging noise variance σ_r^2 and depth/height measurement noise variance $\sigma_z^2 = 1$ for 3D localization.

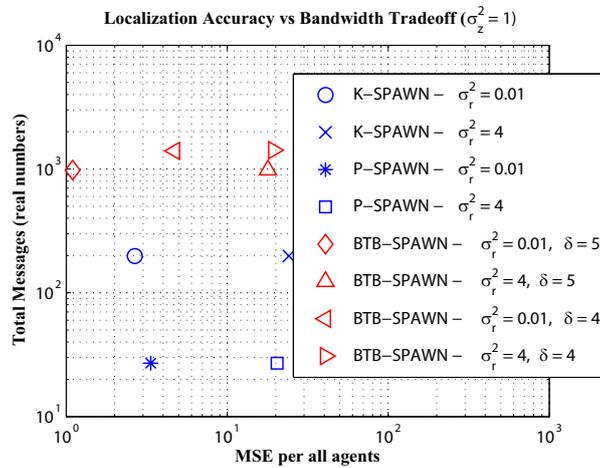


Figure 5.12: BTB-SPAWN further reduces MSE compared to classic SPAWN, while reducing size of exchanged messages one order of magnitude. K-SPAWN and P-SPAWN can further reduce the required bandwidth by one or two orders of magnitude compared to BTB-SPAWN, at the expense of MSE.

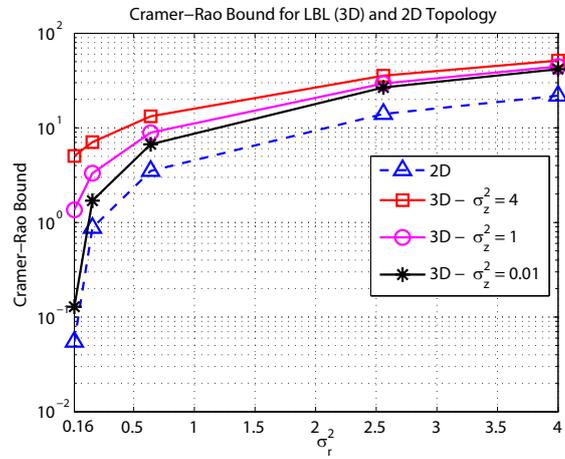


Figure 5.13: The Cramer-Rao bound (CRB) for the 3D long-baseline (LBL) network setup of this work, including ranging noise variance σ_r^2 and depth/height noise variance σ_z^2 . CRB for 2D networks is also depicted as a function of σ_r^2 .

Chapter 6

Conclusion and Future Work

Factor graph-based cooperative localization was extended to bandwidth-friendly versions, tested with Gaussian as well as non-Gaussian, unimodal distance measurement error and compared with cooperative particle filtering, as well as distributed MDS. The tradeoff between localization accuracy and size of total messages exchanged was quantified. It was found that the proposed algorithms can significantly reduce total size of exchanged messages, one- to two- orders of magnitude. In that way, cooperative localization can be offered in narrow-band scenarios (as in underwater or resource constrained sensor networks). Additionally, a computationally- efficient version was found that further improved the localization accuracy, compared to prior-art. Cramer-Rao bound corroborated the efficiency of the proposed algorithms. Future work includes testing with multi-modal ranging error models

Bibliography

- [1] H. Akcan, V. Kriakov, H. Bronnimann, and A. Delis, "GPS-Free node localization in mobile wireless sensor networks", in *Fifth International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE 06)*, June 25, 2006, Chicago, Illinois, USA.
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks (Elsevier) Journal*, vol. 38, no. 4, pp. 393-422, Mar. 2002.
- [3] B. Alavi and K. Pahlavan, "Modeling of the TOA-based distance measurement error using UWB indoor radio measurements," *IEEE Commun. Lett.*, vol. 10, pp. 275-277, Apr. 2006.
- [4] J. N. Ash and L. C. Potter, "Sensor network localization via received signal strength measurements with directional antennas," in *Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing*, 2004
- [5] A. Bahr, J. J. Leonard, and M. F. Fallon, "Cooperative localization for autonomous underwater vehicles," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 714-728, 2009.
- [6] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sensor Networks*, vol. 2, no. 2, pp. 188-220, 2006.
- [7] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360-371, Oct. 2006.

-
- [8] A. Bletsas, A. Vlachaki, E. Kampionakis, G. Sklivanitis, J. Kimionis, K. Tountas, M. Asteris and P. Markopoulos, "Towards Precision Agriculture: Building a Soil Wetness Multi-Hop WSN from First Principles," in *Second International Workshop in Sensing Technologies in Architecture, Forestry and Environment (ECOSENSE) 2011, Belgrade, Serbia*, Apr. 2011.
- [9] N. Bulusu, J. Heideman, and D. Estrin, "Gps-less low cost outdoor localization and tracking system," *IEEE Personal Commun. Mag.*, vol. 7, pp. 28-34, Oct. 2000.
- [10] G. Carter, "Coherence and Time Delay Estimation." Piscataway, NJ: IEEE Press, 1993.
- [11] A. Catovic and Z. Sahinoglu, "The Cramer-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes," *Mitsubishi Electric Research Lab, Tech. Rep. TR-2003-143*, Jan. 2004.
- [12] W. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *J. American Statistical Assoc.* vol. 74, no. 368, pp. 829-836, 1979.
- [13] S. L. Collier and D. K. Wilson, "Performance bounds for passive sensor arrays operating in a turbulent medium: Plane-wave analysis," *The Journal of the Acoustical Society of America*, vol. 113, no. 5, pp. 2704-2718, 2003.
- [14] N. S. Correal and N. Patwari, "Wireless sensor networks: Challenges and opportunities," in *Proceedings of the 2001 Virginia Tech Symposium on Wireless Personal Communications* June 2001, pp. 1-9.
- [15] N. S. Correal, S. Kyperountas, Q. Shi and M. Welborn, "An ultra wide-band relative location system," in *IEEE Conf. on Ultra Wideband Systems and Technologies*, Nov. 2003

-
- [16] J.A Costa, N. Patwari and A.O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Networks*, vol.2, no. 1, pp 39-64, Feb. 2006.
- [17] A.J. Coulson, A.G. Williamson and R.G. Vaughan 1998, "A statistic basis for lognormal shadowing effects in multipath fading channels," *IEEE Trans. on Veh. Tech: 46*, 4 (April), 494-502.
- [18] T. F. Cox and M. A. Cox, *Multidimensional Scaling*. Boca Raton, Florida: Chapman & Hall/CRC, 2001.
- [19] D. Dardari, A. Conti, J. Lien, and M. Z. Win, "The effect of cooperation on localization systems using UWB experimental data," *EURASIP J. Appl. Signal Process. (Special Issue on Wireless Cooperative Networks)*, vol. 2008, pp. 1-11, 2008.
- [20] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, "Convex position estimation in wireless sensor networks," in *Proc. IEEE Int. Conf. on Computer Communications (Infocom)*, Anchorage, AK, 2001, pp. 1655-1663.
- [21] T.Fulford-Jones, D. Malan, M. Welsh, and S. Moulton, "CodeBlue: An ad hoc sensor network infrastructure for emergency medical care," in *International Workshop on Wearable and Implantable Body Sensor Networks*, London, UK, 2004
- [22] P. Groenen, *The majorization approach to multidimensional scaling: some problems and extensions*. DSWO Press, Leiden University, 1993.
- [23] Hashemim H. 1993. "The Indoor Radio Propagation Channel." *Proceedings of the IEEE 81*, 7 (July), 943-968.
- [24] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proc. IEEE Int. Conf. on Computer Communications (Infocom)*, Mar. 2004, pp. 2652-2661.
- [25] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy efficient computing for wildlife tracking: Design tradeoffs and

-
- early experiences with ZebraNet,” in *Conf. Architectural Support for Programming Lang. and Operating Systems (ASPLOS)* October 2002.
- [26] U. A. Khan, S. Kar, , and J. M. F. Moura, ”Distributed sensor localization in random environments using minimal number of anchor nodes,” *IEEE Trans. Signal Processing*, vol. 57, pp. 2000-2016, May 2009.
- [27] E.G. Larsson, ”Cramer-Rao bound analysis of distributed positioning in sensor networks,” *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 334-337, March 2004.
- [28] E. Liao, G. Hollinger, J. Djugash, and S. Singh, ”Preliminary results in tracking mobile targets using range sensors from multiple robots,” in *The 8th International Symposium on Distributed Autonomous Robotic Systems*, Minneapolis, MN, June 2006, pp. 125-134.
- [29] K. W. K. Lui, W.-K. Ma, H. C. So, and F. K. W. Chan, ”Semidefinite programming algorithms for sensor network node localization with uncertainties in anchor positions and/or propagation speed,” *IEEE Trans. Signal Processing*, vol. 57, no. 2, pp. 752-763, Feb. 2009.
- [30] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, ”Wireless sensor networks for habitat monitoring,” in *First ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, Sept. 2002.
- [31] D. Niculescu and B. Nath, ”Ad hoc positioning system (APS),” in *Proc. IEEE Global Commun. Conf. (Globecom)*, vol. 5, San Antonio, AZ, Nov. 2001, pp. 2926-2931.
- [32] B. Ottersten, M. Viberg, P. Stoica, and A. Nehorai, ”Exact and large sample ML techniques for parameter estimation and detection in array processing,” in *Radar Array Processing*, S. S. Haykin, J. Litva, and T. Shepherd, Eds. Springer-Verlag, 1993, pp. 99-151.

-
- [33] N. Patwari, A.O Hero III, M. Perkins, N. Correal and R.J. O’Dea, ”Relative location estimation in wireless sensor networks,” *IEEE Trans. Sig. Proc.*, vol. 51, no. 8, pp. 2137-2148, Aug. 2003
- [34] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, ”The cricket location-support system,” in *Proc. ACM Int. Conf. on Mobile Computing and Networking (Mobicom)*, Boston, M.A., Aug. 2000, pp. 32-43.
- [35] Daniele Puccineli and Martin Haenggi, ”Wireless Sensor Networks: Applications and Challenges of Ubiquitous Sensing,” in *IEEE Circuits and Systems Magazine*, 2005
- [36] Rappaport, T.S. 1996. ”Wireless Communications: Principles and Practice.” Prentice-Hall Inc., New Jersey.
- [37] I. Rekleitis, ”Particle filter tutorial for mobile robots,” in *Technical Report TR-CIM-04-02*, McGill University, 2002.
- [38] E. Robinson and A. Quazi, ”Effect of sound-speed profile on differential time-delay estimation,” *Journal of the Acoustical Society of America*, vol. 77, no. 3, pp. 1086-1090, 1985
- [39] A. Savvides, H. Park, and M. B. Srivastava ”The n-hop multilateration primitive for node localization problems,” *Mobile Networks and Applications*, vol. 8, pp. 443-451, 2003.
- [40] Y. Shang and W. Ruml, ”Improved MDS-based localization,” in *Proc. IEEE Int. Conf. on Computer Communications (Infocom)*, Mar. 2004, pp. 2640-2651.
- [41] Junyang Shen and Andreas F. Molisch, ”Passive location estimation using toa measurements” in *IEEE International Conference on Ultra-Wideband (ICUWB)*,2011
- [42] D.-H. Shin and T.-K. Sung, ”Comparisons of error characteristics between toa and tdoa positioning,” *IEEE Trans. Aerospace Electr. Systems*, vol. 38, no. 1, pp. 307-311, Jan. 2002.

-
- [43] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis* New Jersey: Prentice-Hall, 1997.
- [44] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, Cambridge, MA: The MIT Press, 2006.
- [45] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4-24, Apr. 1988.
- [46] R. Vyas, V. Lakafohis, Z. Konstas, and M. M. Tentzeris, "Design and characterization of a novel battery-less, solar powered wireless tag for enhanced-range remote tracking applications," in *Proc. European Microwave Conference* Rome, Italy, Sept. 2009.
- [47] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427-450, Feb. 2009.
- [48] P. Alevizos, "Factor Graphs: Theory and Applications," Sept. 2012
- [49] N. Fasarakis-Hilliard, P. Alevizos, and A. Bletsas, "Cooperative Localization in Wireless Networks under Bandwidth Constraints", submitted to *46th Asilomar Signals, Systems & Computers Conference*, (to appear) 2013