Abstract:
Deliverable D2.1 summarizes the activities on mobile radio based cooperative positioning during the first half of the project. Numerous message passing algorithms are investigated, and some selected algorithms are compared. The overall goal is to preserve application-compliant accuracy level in very dense and in sparse networks. Therefore, the coordination of cooperative links is mandatory to reduce interference, to alleviate embedded complexity in the mobile nodes or to limit traffic. We present links selection mechanisms and synergetic communication-oriented means supporting the localization function. Specific simulator building blocks have also been developed to assess more realistic performances in the intended heterogeneous and mobile contexts.

Keyword list: cooperative positioning, message passing, link selection, heterogeneous wireless context, decentralized algorithms, indoor environment
Executive Summary

This report summarizes the activities in T2.1 about cooperative and heterogeneous positioning cooperative and heterogeneous positioning. The ongoing research focuses on two theoretical topics. The way how and which information should be exchanged (e.g. among the mobile terminals) in distributed cooperative localization is investigated. The so called message passing algorithms are rather complex compared to time- or signal strength-based ranging algorithms, but the additional information by the local neighbors proofs the additional effort is worth being paid. An accuracy of less than 1m indoors can be achieved by message passing algorithms, and which outperforms the accuracy of the least-squares centralized solution in some deployment and measurements scenarios. Thus, the cooperative localization performance is much better than GPS performance outdoors. However, indoors usually enjoy fine potential in terms of mobile-to-mobile cooperation and heterogeneous access to various radio access technologies (e.g. with respect of multiple access points). With this a coordinated activity is needed to avoid a diminishing performance. Link selection derives out of all available links the best links to reduce the complex calculations in the mobile node to the lowest level according to the accuracy requirements. Different criteria are evaluated to optimize the link selection process. Finally, some first practical implementations are investigated to evaluate how the parametrization of algorithms performs.
# Table of Contents

1 Introduction ................................. 8  

2 Message Passing ............................ 9  
   2.1 State of the art .................................. 9  
   2.2 Criteria, scenarios, commonalities .................. 9  
   2.3 Contributions .................................. 10  
   2.4 Performance of NBP-based Methods in the Presence of Outliers  ... 15  

3 Link Selection ............................. 16  
   3.1 State of the art ................................ 16  
   3.2 Criteria, scenarios, commonalities ................ 16  
   3.3 Achievements .................................. 16  

4 Practical Implementations and Experiments .......... 19  
   4.1 Criteria and scenario ........................... 19  
   4.2 Offline calibration and performance measurement .... 19  
   4.3 Online calibration procedure, ACO ................ 22  

5 Conclusion ................................ 24  

A Appendix ................................ 24  
   A.1 Contribution to D2.1: ACORDE platform: ZigBee modules (distributed location algorithm embedded) ... 26  
   A.2 A New Variant of Nonparametric Belief Propagation for Self-Localization .... 31  
   A.3 Utility Based Node Selection Scheme for Cooperative Localization ........ 38  
   A.4 A Variational Message Passing Algorithm for Sensor Self-Localization in Wireless Networks .... 45  
   A.5 Performance of NBP-based methods in the presence of outliers ........... 52  
   A.6 A Robust Geometric Positioning Algorithm for Heterogeneous Wireless Networks ........ 57  
   A.7 Simulation-based Evaluation of Cooperative Tracking Filters in Realistic Multi-RAT Heterogeneous Contexts .................. 63  
   A.8 A Two-Phases NBP-Based Localization Solution for Ambiguity Mitigation .... 72  
   A.9 Work in progress document for WHERE2 D2.1 Technical Report ........ 85  
   A.10 Contributions on Hybrid Localization Techniques For Heterogeneous Wireless Networks ... 123  
   A.11 A Hybrid Data Fusion Based Cooperative Localization Approach for Cellular Networks .... 138  
   A.12 Accuracy Limits and Mobile Terminal Selection Scheme for Cooperative Localization in Cellular Networks .... 145  
   A.13 Cooperative Localization in a Distributed Base Station Scenario ........ 151  
   A.14 Sensor Localization using Nonparametric Generalized Belief Propagation in Network with Loops .... 157  
   A.15 An Experimental Study of RSS-based Indoor Localization using Nonparametric Belief Propagation based on Spanning Trees .... 166  
   A.17 Optimized Edge Appearance probability for Cooperative Localization based on Tree-Reweighted Nonparametric Belief Propagation .... 182  

References ................................ 187  

References ................................ 187
<table>
<thead>
<tr>
<th>Partner</th>
<th>Name</th>
<th>Phone/Fax/e-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAU</td>
<td>Claus Pedersen</td>
<td>Phone: +45 9940 8615</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>e-mail: <a href="mailto:cpe@es.aau.dk">cpe@es.aau.dk</a></td>
</tr>
<tr>
<td>ACO</td>
<td>Jacobo Domínguez</td>
<td>Phone: +34 942 76 44 00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +34 942 76 44 03</td>
</tr>
<tr>
<td></td>
<td>Lorena de Celis</td>
<td>Phone: +34 942 76 44 00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +34 942 76 44 03</td>
</tr>
<tr>
<td>CEA</td>
<td>Benoit Denis</td>
<td>Phone: +33 4 38 78 18 21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +33 4 38 78 65 86</td>
</tr>
<tr>
<td>DLR</td>
<td>Ronald Raulefs</td>
<td>Phone: +49 8153 282803</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +49 8153 28 1871</td>
</tr>
<tr>
<td>IT</td>
<td>Senka Hadzic</td>
<td>Phone: +351 234 377 900</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +351 234 377 901</td>
</tr>
<tr>
<td>MER</td>
<td>Damien Castelain</td>
<td>Phone: +33 2 234 55 858</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +33 2 23 455 859</td>
</tr>
<tr>
<td></td>
<td>Hadi Noureddine</td>
<td>Phone: +33 2 234 55 858</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +33 2 23 455 859</td>
</tr>
<tr>
<td>UNIS</td>
<td>Na Yi</td>
<td>Phone: +44 1483 68 4703</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +44 1483 68 6011</td>
</tr>
<tr>
<td></td>
<td>Yi Ma</td>
<td>Phone: +44 1483 68 3609</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +44 1483 68 6011</td>
</tr>
<tr>
<td>UPM</td>
<td>Francisco Javier Casajus Quiros</td>
<td>Phone: +34 91 5495700 ext.4006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +34 91 3367350</td>
</tr>
<tr>
<td></td>
<td>Igor Arambasic</td>
<td>Phone: +34 91 5495700 ext.4006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +34 91 3367350</td>
</tr>
<tr>
<td>UR1</td>
<td>Bernard Uguen</td>
<td>Phone: +33 2 23 23 60 33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +33 2 23 23 56 16</td>
</tr>
<tr>
<td></td>
<td>Mohamed Laaraiedh</td>
<td>Phone: +33 2 23 23 56 16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fax: +33 2 23 23 56 16</td>
</tr>
</tbody>
</table>
### List of Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>3(^{rd}) Generation (A mobile communications system of the 3(^{rd}) generation.)</td>
</tr>
<tr>
<td>4G</td>
<td>4(^{th}) Generation (A mobile communications system of the 4(^{th}) generation.)</td>
</tr>
<tr>
<td>AAU</td>
<td>Aalborg University</td>
</tr>
<tr>
<td>ABS</td>
<td>Antilock Brake System</td>
</tr>
<tr>
<td>ACO</td>
<td>Acorde Technologies S.A</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog To Digital Converter</td>
</tr>
<tr>
<td>AOA</td>
<td>Angle Of Arrival</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CBRNE</td>
<td>Chemical, Biological, Radiological, Nuclear Or Explosive</td>
</tr>
<tr>
<td>CEA</td>
<td>Commissariat l’Energie Atomique - Leti</td>
</tr>
<tr>
<td>CoA</td>
<td>Care of Address</td>
</tr>
<tr>
<td>CR</td>
<td>Cognitive Radio</td>
</tr>
<tr>
<td>DBPSK</td>
<td>Differential Binary Phase Shift Keying</td>
</tr>
<tr>
<td>DLR</td>
<td>Deutsches Zentrum Für Luft- Und Raumfahrt E.V.</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>DVB-H</td>
<td>Digital Video Broadcasting Handheld</td>
</tr>
<tr>
<td>ECID</td>
<td>Enhanced cell identification</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically-Erasable Programmable Read-Only Memory</td>
</tr>
<tr>
<td>EPS</td>
<td>Evolved Packet System</td>
</tr>
<tr>
<td>EUR</td>
<td>Eurecom</td>
</tr>
<tr>
<td>FAP</td>
<td>Femto Access Point</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System For Mobile Communications</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HDTV</td>
<td>High Definition Television</td>
</tr>
<tr>
<td>HDF</td>
<td>Hybrid Data Fusion</td>
</tr>
<tr>
<td>HI-FI</td>
<td>High Fidelity</td>
</tr>
<tr>
<td>HKC</td>
<td>City University Of Hong Kong</td>
</tr>
<tr>
<td>HSDPA</td>
<td>High Speed Downlink Packet Access</td>
</tr>
<tr>
<td>HSPA</td>
<td>High-Speed Packet Access</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>ICIC</td>
<td>Inter Cell Interference Coordination</td>
</tr>
<tr>
<td>ICT</td>
<td>Information And Communication Technology</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute Of Electrical And Electronics Engineers</td>
</tr>
<tr>
<td>IF</td>
<td>Intermediate Frequency</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPTV</td>
<td>Internet Protocol Television</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial Science And Medical</td>
</tr>
<tr>
<td>IT</td>
<td>Instituto Telecomunicas</td>
</tr>
<tr>
<td>JTAG</td>
<td>Joint Test Action Group</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LBS</td>
<td>Location Based Services</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>LDP</td>
<td>Location Dependent Parameters</td>
</tr>
<tr>
<td>LDR</td>
<td>Low Data Rate</td>
</tr>
<tr>
<td>LNA</td>
<td>Low Noise Amplifier</td>
</tr>
<tr>
<td>LOS</td>
<td>Line Of Sight</td>
</tr>
<tr>
<td>LPP</td>
<td>LTE Positioning Protocol</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access (Medium Access Layer)</td>
</tr>
<tr>
<td>MER</td>
<td>Mitsubishi Electric R&amp;D Centre Europe MERCE</td>
</tr>
<tr>
<td>MIP</td>
<td>Mobile IP</td>
</tr>
<tr>
<td>MHZ</td>
<td>Megahertz</td>
</tr>
<tr>
<td>MT</td>
<td>Mobile Terminal</td>
</tr>
<tr>
<td>NBP</td>
<td>Nonparametric Belief Propagation</td>
</tr>
<tr>
<td>NBP-ST</td>
<td>Nonparametric belief propagation based on spanning trees</td>
</tr>
<tr>
<td>NGBP-PJT</td>
<td>Nonparametric Belief Propagation based on Pseudo-Junction-Tree</td>
</tr>
<tr>
<td>NLOS</td>
<td>Non Line Of Sight</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency-Division Multiplexing</td>
</tr>
<tr>
<td>OQPSK</td>
<td>Offset Quadrature Phase-Shift Keying</td>
</tr>
<tr>
<td>OTDOA</td>
<td>Observed Time Difference Of Arrival</td>
</tr>
<tr>
<td>OTE</td>
<td>Hellenic Telecommunications Organization</td>
</tr>
<tr>
<td>PAN</td>
<td>Personal Area Network</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>PER</td>
<td>Probable Error In Range</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical (Physical Layer)</td>
</tr>
<tr>
<td>PI</td>
<td>Positioning Information</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase Locked Loop</td>
</tr>
<tr>
<td>QoE</td>
<td>Quality Of Experience</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality Of Service</td>
</tr>
<tr>
<td>RAT</td>
<td>Radio Access Technology</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RGPA</td>
<td>Robust geometric positioning algorithm</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indication</td>
</tr>
<tr>
<td>RT</td>
<td>Ray Tracing</td>
</tr>
<tr>
<td>RTOA</td>
<td>Round Trip Time Of Arrival</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SDRAM</td>
<td>Synchronous Dynamic Random Access Memory</td>
</tr>
<tr>
<td>SIG</td>
<td>Sigint Solutions Ltd</td>
</tr>
<tr>
<td>SINR</td>
<td>Low Signal To Interference And Noise Rate</td>
</tr>
<tr>
<td>SIR</td>
<td>Siradel</td>
</tr>
<tr>
<td>SME</td>
<td>Small-And Medium Enterprises</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal To Noise Ratio</td>
</tr>
<tr>
<td>SOC</td>
<td>System On Chip</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Port Interface</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDOA</td>
<td>Time Difference Of Arrival</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>TOA</td>
<td>Time Of Arrival</td>
</tr>
<tr>
<td>TV</td>
<td>Television</td>
</tr>
<tr>
<td>TP-NBP</td>
<td>Two-phased nonparametric belief propagation</td>
</tr>
<tr>
<td>TWI</td>
<td>Two Wire Interface</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>UNIA</td>
<td>University Of Alberta</td>
</tr>
<tr>
<td>UNIS</td>
<td>University Of Surrey</td>
</tr>
<tr>
<td>UPM</td>
<td>Universidad Politecnica De Madrid</td>
</tr>
<tr>
<td>UI1</td>
<td>University Of Rennes 1</td>
</tr>
<tr>
<td>URW-NBP</td>
<td>Uniformly Reweighted Nonparametric Belief Propagation</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>UWB</td>
<td>Ultra Wide Band</td>
</tr>
<tr>
<td>VMP</td>
<td>Variational message passing</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice Over Internet Protocol</td>
</tr>
<tr>
<td>WHERE</td>
<td>Wireless Hybrid Enhanced Mobile Radio Estimators (Project Acronym Of Phase 1)</td>
</tr>
<tr>
<td>WHERE2</td>
<td>Wireless Hybrid Enhanced Mobile Radio Estimators (Project Acronym Of Phase 2)</td>
</tr>
<tr>
<td>WiFi</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>WiMAX</td>
<td>Worldwide Interoperability For Microwave Access</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WPx</td>
<td>Work Package x</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
1 Introduction

Task 2.1 of the WHERE2 project deals with radio based cooperative positioning for heterogeneous networks. In this deliverable D2.1 we focus on theoretical aspects and practical implementations.

The theoretical aspects investigate how positioning and related information shall be shared successfully in a network with mobile and (static) anchor nodes. The decentralized feature plays a key role in our investigations. Furthermore, we study different aspects such as describing the (non-)Gaussian noise efficiently, how to combine the different messages properly or how we could deal with mobility in turn from both algorithms design and evaluation points of view.

The shared (exchanged) information or message uses an access and packet error free wireless channel. In very dense networks it is a non-realistic and best case assessment. Therefore, we investigate also algorithms that select a priori the most promising nodes to exchange the messages. Criteria to select a priori could be power consumption, geometrical constellation, interference management, dynamic heterogeneous connectivity, theoretical location accuracy, etc. The practical investigations address how the mapping of received signal strength and an online calibration could be performed. The applied scenarios are indoor environments or GPS-denied environments. Investigations cover different density of active nodes in the given area and static and mobile scenarios (D1.1b and D2.3). The summarized work of T2.1 focuses on a scenario that includes links between mobile nodes as well as between mobile nodes and anchors (or base stations).

Section 2 Message Passing summarizes the different message passing concepts that are investigated. Besides the individual concepts some first results are outlined comparing the different algorithms with the same setups. The investigations presume a static setup of all nodes. However, this work will be carried on to integrate mobility.

Section 3 Link Selection intends to improve the allocation of links for the whole network. Different selection criteria, such as the expected accuracy or the geometrical setup are applied.

Section 4 Practical Implementations and Experiments covers experimental results of cooperative nodes. Selected existing algorithms use the parameter received signal strength indicator (RSSI). The parametrization of the propagation model as well as the applied algorithms are based on the results. This work is complemented by a continuous online calibration procedure.

Section A is an Appendix that contains a collection of already/soon to be published articles or reports from the WHERE2 project. This collection contains detailed information to the various sections in the deliverable.
2 Message Passing

In this section we present a number of message passing schemes for localization. These techniques share the properties that they can be deployed on the nodes in a network in a decentralized manner. The network nodes can then cooperate with each other by exchanging messages which describe the individual nodes’ belief about its position. The network nodes use these messages along with observed inter-nodes distances as input for the localization algorithm to update/refine their position estimates.

The message passing localization schemes considered in this deliverable are:

- The Nonparametric belief propagation based on spanning trees (NBP-ST) localization algorithm [1].
- The Generalized nonparametric belief propagation based on pseudo-junction-tree (NGBP-PJT) localization algorithm [2].
- The Uniformly reweighted nonparametric belief propagation (URW-NBP) localization algorithm [3].
- The Two-phases nonparametric belief propagation (TP-NBP) localization algorithm [4].
- The Variational message passing (VMP) localization algorithm [5].
- The Robust geometric positioning algorithm (RGPA) [6].

2.1 State of the art

A well known message passing method for distributed inference is belief propagation [7]. The computational effort is proportional to the number of links inside the graph. Therefore, wireless sensor networks will apply well. However, belief propagation will not suit well in a heterogeneous network with unknown, non-linear and non-Gaussian conditional probabilities. Ihler et al. [8] proposed the nonparametric belief propagation (NBP) by approximating via particles. In dense networks NBP reaches its limits as loops inside the network cause non-convergence. Moreover, lower practical network density could lead to geometrical ambiguities. The asymmetry of the involved links in a realistic heterogeneous context (e.g. different ranges of transmission coverage) and mobility are challenges that are open and addressed in T2.1.

2.2 Criteria, scenarios, commonalities

The presented message passing schemes address scenarios listed in [9] ranging from private areas to professional and public areas. The NBP-ST, NGBP-PJT and URW-NBP algorithms consider static, indoor, single-floor, private scenarios. Professional areas (static, indoor, single-floor) are targeted by TP-NBP and RGPA with 1 mobile terminal within $100 - 1000m^2$ while VMP considers static, indoor, single-floor public areas with 1 mobile terminal within $10 - 100m^2$. We note here, that the majority of these localization algorithms can be deployed in other smaller or larger scenarios than the ones described in this deliverable. Mobility is not yet addressed in this deliverable but under investigation. The scenarios addressed are summarized in Table 1.

2.2.1 Properties of the Message Passing Algorithms

For clarity, we state the properties of the different algorithms with regards to accuracy, latency, complexity, ambiguity and cooperation. These properties are listed in Table 2. Whatever the considered algorithm, the final positioning accuracy obviously depends on the quality of the location-dependent feeding measurements. The algorithms based on nonparametric belief propagation all provide high accuracy, in indoor scenarios as low as $0.8m$ (please check the publications for detailed property descriptions). The VMP algorithm provide moderate to high (< 1m) accuracy depending on the variance of the observation noise. The accuracy of the RGPA depends on the accuracy of the location dependent parameters that are available for the algorithm to use. However, the RGPA usually outperforms the randomly initialized maximum likelihood solution.
The iterations of the nonparametric belief propagation algorithms NBP-ST, NGBP-PJT and URW-NBP scale with the square number of particles user and the number of iterations. These algorithms require a small number of iterations (8 or less). Similarly, TP-NBP requires only a few iterations. The VMP algorithm has a low number of iterations since each network node only needs to broadcast 3 real values in each iteration. This algorithm converges in 10–30 iterations.

Complexity defines the number of operations and the amount of data that needs to be processed. The operations itself are also different for the algorithms, as some apply complex calculations and others perform arithmetic operations which are rather simple. The complexity of all the NBP algorithms and the VMP algorithm is moderate to high due to the calculations of the messages to be exchanged between the network nodes. The RGPA offers a low complexity because it is mainly based on simple arithmetic operations.

To resolve ambiguities in the localization estimation the NBP algorithms utilize the current state of the art approach which takes information on absent connections between network nodes into account (except the TP-NBP). The VMP algorithm does not use any means to mitigate ambiguities in the location estimates. The RGPA can resolve ambiguities when only two TOAs are available. All the algorithms presented in this deliverable assume full cooperation between the nodes in the networks (except RGPA, which will be extended to full cooperative).

<table>
<thead>
<tr>
<th>NBP-ST</th>
<th>NGBP-PJT</th>
<th>URW-NBP</th>
<th>TP-NBP</th>
<th>VMP</th>
<th>RGPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Moderate/High</td>
<td>Moderate/High</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>8</td>
<td>3–8</td>
<td>10–30</td>
<td>n/a</td>
</tr>
<tr>
<td>Moderate/High</td>
<td>Moderate/High</td>
<td>Not resolvable</td>
<td>Moderate/High</td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Resolvable</td>
<td>Resolvable</td>
<td>Not resolvable</td>
<td>Resolvable</td>
<td>Resolvable (for 2 TOAs)</td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 2: Accuracy, iterations, complexity, ambiguity and cooperation properties of the message passing algorithms.

### 2.3 Contributions

#### 2.3.1 The NBP Based on Spanning Trees (NBP-ST) Localization Algorithm

The NBP-ST algorithm represents two (or more) independent runs of the NBP algorithm on spanning trees. The spanning trees are created by using the distributed breadth first search (BFS) method on the original network graph. Examples of spanning trees are shown in Figure. Preliminary results were reported in [1]. Experimental analysis is reported in Section of this deliverable.
2.3.2 The Generalized NBP Based on Pseudo-Junction-Tree (NGBP-PJT) Localization Algorithm

The solution for cooperative localization proposed in [12], NGBP based on junction tree (NGBP-JT) is only tractable for small-scale networks because of the i) high-complexity of the junction tree formation [13], and ii) high-dimensionality of the particles. To alleviate this, the NGBP based on pseudo-junction tree (NGBP-PJT) algorithm is proposed in [2]. The main difference comparing with the standard method is the formation of a pseudo-junction tree (PJT), which represents the approximated junction tree based on thin graphs. The PJT was created using following approximations:

- Number of cliques is limited (e.g., in the order of number of nodes).
- In order to decrease the dimensionality of the problem, each clique includes no more than 3 nodes.
- Since the triangulation is an expensive procedure [13], hence, it is avoided.

In order to satisfy the first two conditions, the number of edges in the graph must be decreased by the formation of a thin graph. This can be easily done using a modified version of the BFS method. Note that this modification will make a graph with a small number of short loops, not a spanning tree. These loops are used to form 3-node cliques, and rest of the pairs of the nodes will form 2-node cliques. Having defined cliques, the cluster graph can be formed by connecting all pairs of cliques with non-empty intersection. The junction-tree, as well as pseudo-junction tree, is the maximum spanning tree of the cluster graph [13]. The original graph and PJT are illustrated in Figure 2. Once the PJT is created, the NGBP method can be applied to the PJT (NGBP-PJT). In order to decrease the number of high-dimensional particles, more informative importance density functions, and dimensionality reduction of the messages are used.

2.3.3 The Uniformly Reweighted NBP (URW-NBP) Localization Algorithm

The two methods (NBP-ST and NGBP-PJT), require some kind of graph transformations before applying message passing methods. This can be avoided using tree-reweighted NBP (TRW-NBP) proposed in [14]. However, finding edge appearance probabilities for each edge in the graph is intractable (especially, for distributed implementation) except for certain types of graphs. See Figure 3 for illustration of a 4-node network. Therefore, the case of uniform ”edge appearance probabilities” are considered [2]. For the novel method, uniformly-reweighted NBP (URW-NBP)\(^1\) empirical values of the edge appearance probabilities are obtained as a function of node degree in the network. Through Monte Carlo simulations, the performance gains in terms of RMSE and KLD w.r.t. the true distribution have been verified.

\(^1\)This part is done in cooperation with Henk Wymeersch (Chalmers University of Technology, Sweden), and Federico Penna (Politecnico di Torino, Italy).
2.3.4 The Two–Phases NBP (TP-NBP) Localization Algorithm

The TP-NBP algorithm [4] is a NBP-based localization algorithm that addresses the problem of flip ambiguities (see Figure 4) in the location estimation by considering the connectivity information between the nodes in the network (Figure 5). That is, two nodes that are not connected are probably far away from each other. Compared to the standard NBP algorithm [8], the algorithm improves the accuracy and reduces the data communication overhead. The TP-NBP scheme performs in two phases: In the first phase, the classical NBP is applied without considering the connectivity information. In the second phase, a new algorithm based on estimation in discrete state space is proposed for solving the flip ambiguities. This solution has the advantage of reducing the amount of communicating particles and improving the accuracy of the computed positions.

In some cases, the presence of a priori information may compensate the flip ambiguity, as for example map constraints (e.g., the location of a femto base station or an access point is constrained to be inside the owner’s apartment). The NBP method is suitable for handling map constraints which are considered in T2.2.
Figure 4: Example of a flip ambiguity. A target node is connected to three anchor nodes which are nearly collinear. Small errors in the ranging measurements result in a flipped position estimate.

Figure 5: Target node 8 is connected to target nodes 6, 9 and 13 which are nearly collinear. Two potential estimates are identified during the first phase of the TP-NBP. The second phase uses the connectivity information to select the most probable estimate.

2.3.5 The Variational Message Passing (VMP) Localization Algorithm

The VMP localization algorithm [5] is based on variational message passing which is an alternative to Bayesian inference in factor graphs [15], [16].

The proposed algorithm aims at minimizing the communication overhead between the network nodes. This is obtained by modelling the positions of the sensors as circular symmetric Gaussian pdfs which are completely described by their means and variances. This modelling yields closed-form expressions for the messages that are passed between the network nodes. Consequently, the communication overhead between the nodes is very low since these messages can be represented by as little as three real values in the 2D case.

In this contribution the algorithm only considers directly communicating node pairs, i.e. ambiguities in the position estimation are not mitigated via the use of connectivity information between the nodes. This is evident from our simulation results in which some nodes (typically at the border of the sensor deployment area) are ambiguously located (see Figure 6), located, and hence might necessitate applying further mitigation approaches like the one described in 2.3.4. The performance of the VMP algorithm is...
verified by Monte Carlo simulations.

Figure 6: Typical result of the VMP localization algorithm after 30 iterations. Clearly, some sensors at the right border are ambiguously located due to the fact that the algorithm has no ambiguity mitigating capabilities.

2.3.6 The Robust Geometric Positioning Algorithm (RGPA)

The RGPA described in [6] is a decentralized scheme that solves the localization problem from a geometric point of view based on constraints which bound a certain region in space. These constraints are obtained from classic location dependent parameters (LDPs) such as TOA, TDOA or RSS. The constraints obtained from these LDPs are shown in Figure 7.

The RGPA can be seen as an engine deployed on each mobile device in order to perform localization in a decentralized manner (Figure 8). The inputs to this engine are:

- LDPs: location dependent parameters are radio observables like TOA, RSSI, TDOA, AOA, etc.
- Messages: information obtained on neighbourhood such as locations of neighbours, connectivity, and pairwise distances, etc.
- Map constraints (from T2.2 (at least)): any information that clip the region where the position will be estimated.

The possible outputs are:

- Position and positioning accuracy of the device represented by a box and its centroid.
- The positions (boxes) of the neighbours to be used as messages by other devices.

Generally, the RGPA is applicable in all types of indoor and outdoor scenarios containing either homogeneous or heterogeneous RATs. In this contribution, the RGPA is deployed in static, indoor scenarios with a low density of nodes since these are well suited for message passing localization algorithms.
2.4 Performance of NBP-based Methods in the Presence of Outliers

NBP-based localization algorithms perform inference in probabilistic graphical models, where every node computes an approximation of its posterior marginal distribution from its local a priori information and messages received from neighboring nodes. Probabilistic models for measurements need to be known in order to define potential functions between every two connected nodes in the probabilistic graph. The particle-based implementation of these algorithms makes them capable to handle any measurements model. In we assess the performance of four such algorithms under the occurrence of outlier processes in the ranging measurements: TP-NBP, URW-NBP, NBBP and NBP. An outlier is an observation that is highly deviated from the true distance value. We also performed simulations for the classical centralized least-squares (LS) estimator initialized using SDP method. The details of the comparisons are outlined in \(A5\).
3 Link Selection

In this section, we address the problem of link selection for localization. Link selection procedures are applied for the purpose of both increased accuracy and restrictive use of limited resources, in cases when the target has more than the minimum number of reference nodes available, or, in the cooperative case, a number of cooperative links with different levels of reliability. Increased accuracy results from using most reliable and geometrically useful links, while resource saving comes from the fact that computational complexity is proportional to the number of links taken into account during information fusion. Besides power, one can also refer to the available spectrum as a limited resource. Gain can also be achieved in terms of latency and traffic. In this deliverable, we look at the link selection problem from different perspectives, considering various scenarios and parameters, as summarized in 3.2.

3.1 State of the art

In [17], [18], the geometrical impact has been addressed and compared to the approach of using the closest reference nodes. However, most of these selection schemes have been proposed for non-cooperative localization. More recent approaches address cooperative scenarios, and the selection criteria are mainly based on theoretical localization performance limits such as CRLB or SPEB [19], [20], [21]. In [19], unreliable links are successively discarded based on CRLB analysis during the first phase - connectivity based coarse positioning. In this manner resources are saved as the number of packet exchanges in the second, refined TOA based ranging phase is reduced. The algorithm in [20] includes both transmit and receive censoring. Transmit censoring prevents unreliable position estimates to be broadcasted, while receive censoring eliminates useless links after collecting information. All censoring decisions are distributed and based on a modified CRLB. A node selection least squares algorithm has been presented in [22]. The selection strategy is to use only nodes providing valuable information in terms of RSS measurement quality so that the accuracy is not degraded.

Besides the use of theoretical localization performance limits as selection criteria, concepts from coalitional games and utility functions have also been adopted to the node selection problem. In [23], [24] methods based on minimization of mean square error (MSE) are discussed. Utility of each set consisting of Na nodes is defined as the reciprocal of the mean square error. The work of Ghassemi [25], [26] uses notions from game theory, and utility is defined as information gain from a node, i.e. the mutual information between the prior density of target position and the measurement. Additionally, a price for transmission is included to account for the current energy level in the nodes, and the energy needed for data transmission. An entropy based node selection scheme has been proposed in [27], adopting heuristics instead of the computationally intensive optimal solution. To model the tradeoff between power consumption and localization performance, a scheme based on coalitional game theory has been proposed in [28].

3.2 Criteria, scenarios, commonalities

We consider the link selection problem for various environments, such as indoor professional environment, both static (IT) and including mobility (CEA), outdoor urban canyon in a cellular scenario - single cell and distributed BS (UNIS), as shown in Figure 9 and exploiting different location-dependent radio parameters, e.g., RSS (UNIS, IT) or TOA (DLR). In this sense also the ranging error is modeled by means of lognormal for RSS based ranging, or centered Gaussian (CEA). We mainly consider accuracy as performance indicator, even though latency plays a significant role, especially in tracking applications (CEA).

3.3 Achievements

Part of the work carried out has been devoted to the development of specific simulation architecture in the context of heterogeneity and realistic human mobility patterns (CEA). The tool supports multi-standard protocol stacks at mobile terminals. In terms of MAC layers, Wi-Fi and ZigBee are already available, and regarding PHY, Wi-Fi at 2.4 GHz, ZigBee at 868 MHz and 2.4GHz, and simplified IR-UWB are
available. The simulator also supports realistic environmental features like the presence of walls within a semideterministic simulation approach. An example of typical indoor heterogeneous and cooperative scenario is depicted in Figure 10. Parametric propagation models are adapted over each link, depending on LOS/NLOS conditions, as in Figure 11. Ranging error models are centered Gaussian, while the variance depends on the considered RAT. Several mobility models are supported.

Another Java based simulator has been developed to assess the performance of cooperative localization techniques supporting several decentralized data fusion and filtering algorithms and mobility models such as Random-way-Point, Levy-Flight, and White Noise Acceleration. Here OFDM based systems, such as LTE are considered. For an OFDM signal the ranging variance depends not only on the distance between nodes, but also on transmit power, carrier frequency, bandwidth, etc., all of which can be seen as resources. Hereby multi-user aspects have been addressed. Centralized and decentralized resource allocation schemes have been analyzed, as well as a bidding method that combines both. The greedy performed based on the derived CRLB for TOA. In addition, cooperative peer-to-peer links are assumed, adding uncertainty to the shared information. The goal is to reduce error propagation by weighting the links accordingly, and the method can be applied to dynamic scenarios. Weighting the cooperative links accordingly with an appropriate confidence factor could improve or worsen the performance accuracy, as illustrated in Figure 12.

The concept of cooperative localization has been also introduced to cellular networks. For such a scenario, a link selection scheme has been developed to reduce training overhead, without degrading.
Figure 11: Example of simplified semi-deterministic approach to model

Figure 12: Simulation result - 9 MTs are uniformly located in a 30x30 map, the ranging variances of all the links are set to 0.09m$^2$. The communication range is 7 m.

The accuracy $A_{11}$, $A_{12}$, $A_{13}$ Specifically, a static single cell scenario is addressed, and the physical environment considers urban canyon. The idea is to use several previously located mobile terminals as reference nodes, when the a priori knowledge of reference nodes locations is imperfect. The main result is the derivation of a closed form expression for the squared position error bound (SPEB) taking into account the imperfect anchor locations. In this sense, the effect of imperfect a priori knowledge on SPEB is equivalent to the increase of variance of RSS based distance estimation.

However, this closed form expression, although derived for a cellular scenario, can be applied to any general case of RSS based localization. In this sense, it can be relevant for iterative multilateration, where each newly localized node serves as a virtual anchor for its neighbors $A_{3}$ Virtual anchors have different degrees of uncertainty in their location estimates. By incorporating the virtual anchor uncertainty into the CRLB, and using the fact that in case of LS position calculation is represented by the trace of covariance matrix, all this information can be used for link selection in the iterative multilateration algorithm. Specifically, a utility function based approach has been developed to avoid error propagation. Node selection problem is defined as choosing the set that maximizes the accuracy, subject to constraints given by nodes with limited processing capacity. In that sense we define utility functions that incorporate benefit factors, indicating link reliability and nodes relative geometry, and penalty factors such as uncertainty of anchor node locations and power consumption. Coalition value reflects the achievable localization accuracy, as can be seen in the scatter plot (Figure 13).
4 Practical Implementations and Experiments

In this section, we address the use of hardware platforms to carry out experiments, in order to parameterize our existing localization algorithms based on received signal strength indicator (RSSI) measurements. First we discuss the general criteria that guide our efforts in this topic, as well as the scenario which is considered. Then we proceed to describe an offline procedure, carried out at UPMA, to parameterize the propagation model, as well as the results obtained by using those parameters. Finally, we discuss a continuous online calibration procedure, to be carried out at ACOA.

4.1 Criteria and scenario

We consider practical, hardware implementations of wireless nodes communicating using the ZigBee protocols. These nodes communicate and in the process measure RSSI readings. Out of these readings, the nodes aim to use our proposed localization algorithms to determine their positions.

However, a key step in this process is mapping the RSSI readings to a distance estimate, via a propagation model. In this work we will consider the well-known log-normal propagation model. It takes exactly two parameters: $P_0$, the RSSI at a known, reference distance $d_0$; and $n_p$, the propagation exponent with which RSSI decays with distance.

The scenario we consider is the indoor, single-floor, private scenario, as described in the deliverable D1.1.

4.2 Offline calibration and performance measurement

The simplest form of proceeding is to carry out offline measurements to estimate the parameters $P_0$ and $n_p$, and then plug these values into our localization algorithms to estimate their performance given these static, pre-calculated values and a realistic network setup. A summary of the procedure and the results is given here, and a more detailed treatment is given in the Appendix.

Nonparametric belief propagation (NBP) is a well-known particle-based message passing method for cooperative localization in wireless networks, as already mentioned in Section 2. However, due to the double counting problem, NBP usually provides less accurate estimates, and also its convergence is not guaranteed. One of the methods to solve this problem is NBP based on spanning trees (NBP-ST).

NBP-ST algorithm represents two (or more) independent runnings of the NBP algorithm based on formed spanning trees (see Figure 14). Spanning trees are created by distributed breadth first search (BFS) method which is optimal for unweighted graphs. Preliminary results have already been reported in D2.4 of the WHERE project, so now we focus on an experimental analysis. More details about the algorithm can be found in [10].
We obtained measurements (RSSI) in our lab using Iris wireless sensor nodes equipped with the AT86RF230 transceiver (Figure 14a). The AT86RF230 is a high performance RF-CMOS 2.4 GHz radio transceiver specially targeted to low cost ZigBee/IEEE802.15.4 applications. The transmitter provides programmable output power: -17 dBm up to 3 dBm. The receiver, with -101 dBm sensitivity, generates digital signals with a 3 dB granularity. The data is stored in a 128-byte dual port SRAM, from which 8 bytes are reserved.

In order to estimate the distance between sensors, we placed two sensors, 2m from the floor, in our 5m x 10m lab (Figure 14b) and set the transmission power to 3 dBm. We obtained RSSI measurements at 8 equidistant inter-sensor distances ($k \cdot 1.2m$, $k = 1, ..., 8$). For each of them, we obtained 1000 measurements. Because of the 3 dB granularity of RSSI, we assume that the real power is a random variable uniformly distributed within the interval (RSSI - 1.5 dB, RSSI + 1.5 dB).

Figure 14: (a) Crossbow's Iris wireless sensor node, (b) Illustration of the experiment in our lab

Using these RSSI measurements, our goal is to obtain all the necessary parameters for the indoor model: the path-loss exponent, reliable distance estimation, and the probability of detection.

First, we define a reference point ($P_0(d_0 = 2.4m) = -61$ dBm), and using least-squares fitting we find the optimal value of the path-loss exponent, being $n_p = 2.7$. As we expected, our indoor model is not similar to the ideal one (see Figure 15), so the distance cannot be always reliably estimated using the log-normal model. For instance, the averaged received power of -66 dBm corresponds to three different distances (4.6m, 7m, and 9.6m). This is because the received power is not a monotonically decreasing function of the distance. Therefore, we have to cut out the area below the threshold power (-64 dBm) because this area corresponds to the non-monotonic part of the function. Above the threshold, each received power corresponds to a unique distance, which makes this model reliable for our scenario. In addition, since we excluded data below the threshold, we re-estimate $n_p$ using only the remaining data, i.e., $n_p = 1.2$. Regarding the probability of detection, we have to follow a defined model, so we assume that, if the power is less than the threshold (-64 dBm), there is no communication among nodes. We found that the corresponding distance is 4m, so this will be the maximum value of transmission radius ($R$).

We carried out several simulations, which will show the performance of NBP-ST and NBP methods. We assumed that there are 50 unknowns and 10 anchors in a 5m x 10m area. Unknown nodes are deployed randomly within this area and anchor nodes are fixed (8 along the edges and 2 in the center area). This constraint (usually, realistic for an indoor environment) helps the unknown nodes near the edges which suffer from low connectivity. The number of iteration is set to $N_{iter} = 3$. All simulations are done for $N = 50$ particles with respect to the transmission radius ($2m < R < 4m$). Finally, each point in the simulations represents the average over 30 Monte Carlo trials.

For NBP-ST, we used 2 and 3 spanning trees. As we can see in Figure 16a, NBP-ST performs better than NBP starting from some value of $R$, which controls the connectivity. We can conclude the same for
true distance [m] | received power [dBm] 
--- | ---
real model | 
ideal model | 
threshold power | 
our model | 

Figure 15: Reliable model for distance estimation

the coverage (Figure 16b), which represents the percentage of located nodes with error less than predefined tolerance. Obviously, for higher values of $R$, there is a large number of loops in the network (hundreds, in our case) which decreases the performance of the NBP method. For lower values of $R$, we did not include all information (i.e., we used only 2 or 3 spanning trees) available, so in this case the NBP outperforms NBP-ST.

Figure 16: Comparison of (a) accuracy, and (b) coverage.

To measure the communication cost, we count elementary messages, where one elementary message is defined as simple scalar value. We assumed that this data is represented in single precision floating-point format that occupies 4 bytes in the memory. As we have already mentioned, 8 bytes are already reserved, so the size of elementary message is 12 bytes. According to Figure 17, NBP-ST performs better than NBP for $R > 3.3m$ only if we use 2 spanning trees. In order to explain this we have to remember two main things we have taken into account: i) removing the edges in order to form the spanning trees and ii) running NBP two times in these spanning trees. The former decreases the communication, but the latter increases it. Therefore, in sparsely connected networks the second operation predominates, but in highly connected networks the first one predominates. Furthermore, it is important that the communication cost is nearly constant with respect to the transmission radius. This feature provides us more precise information about battery life. The main conclusion is that NBP-ST (with 2 spanning trees) algorithm performs better than NBP in terms of accuracy and communication cost, for $R > R_{\text{min}}$. In our case $R_{\text{min}} = 3.4m$, but this parameter depends on the density of the network (i.e., average connectivity).
4.3 Online calibration procedure, ACO

In order to improve the performance of localization provided by ACORDE platform, an online calibration procedure has been considered. The main point is the continuous estimation of parameters $P_0$ and $n_p$.

ACO contribution will be focused on the dynamic estimation of these two parameters in order to improve the accuracy of the position estimation [29]. The right selection of the values of parameter $P_0$ and parameter $n_p$ is the best way to improve the current algorithm hosted on the ZigBee platforms, as for example in the final demonstration produced in WP4.

The work that will be carried out for the estimation of these two parameters will be split in several stages:

1. Determine initial conditions.
   a) Scenario selection
   b) Parameter $P_0$ calibration: obtained in a no-obstacle one-meter distance signal strength measurement from the reference nodes.
   c) Parameter $n_p$ characterization: select one point in the scenario (center of the scene) and execute the current Location Algorithm with the ZigBee platform modifying parameter $n_p$.
   d) Initial condition selection: process the data obtained and selects the best pair $(P_0, n_p)$ to be used as initial conditions in the selected scenario.

2. Once the initial conditions have been obtained, the entire scenario will be characterized with different values of the pair $(P_0, n_p)$.
   a) Build a grid in the scenario in order to divide it into smaller areas.
b) Selection of one area.

3. When the scenario is characterized a dynamic selection of these values will be implemented in order to improve the location estimation taking into account the previous position estimated of the mobile node.
5 Conclusion

This intermediate deliverable D2.1 summarizes the numerous activities in T2.1 about cooperative positioning. Different message passing algorithms for different densities of mobile nodes were investigated. Less than 1m of the RMSE could be achieved with one node per every $10^{\phantom{1}2}$ m. Link selection methods allow to reduce complexity at the receiver to perform calculations as well as the interference (censoring at the transmitter) in dense networks. Finally, some of the message passing algorithms we parameterized and evaluated with real measurements. The ongoing work in task T2.1 will be connected to two other activities inside the WHERE2 project. T2.3 (early outcome is D2.3) builds a synthetic environment and could offer via the applied ray-tracing tools a set of measurement data as well as a common motion model to offer the same ground for comparisons. Furthermore, WP4 defined an interface that T2.1 will investigate to select a reduced version of the presented methodologies to integrate for the final demonstration. The access to the hardware platforms in WP4 is considered through a common database that provides the relevant data and metrics that the algorithm could exploit.

A Appendix

The appendix contains a collection of articles and reports with detailed information to the summaries of the different sections in this deliverable, which are results of the WHERE2 project. Table 3 lists titles of the following sections in the appendix.
<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Contribution to D2.1: ACORDE platform: ZigBee modules (distributed location algorithm embedded)</td>
<td>26</td>
</tr>
<tr>
<td>A.2</td>
<td>A New Variant of Nonparametric Belief Propagation for Self-Localization</td>
<td>31</td>
</tr>
<tr>
<td>A.3</td>
<td>Utility Based Node Selection Scheme for Cooperative Localization</td>
<td>38</td>
</tr>
<tr>
<td>A.4</td>
<td>A Variational Message Passing Algorithm for Sensor Self-Localization in Wireless Networks</td>
<td>45</td>
</tr>
<tr>
<td>A.5</td>
<td>Performance of NBP-based methods in the presence of outliers</td>
<td>52</td>
</tr>
<tr>
<td>A.6</td>
<td>A Robust Geometric Positioning Algorithm for Heterogeneous Wireless Networks</td>
<td>57</td>
</tr>
<tr>
<td>A.7</td>
<td>Simulation-based Evaluation of Cooperative Tracking Filters in Realistic Multi-RAT Heterogeneous Contexts</td>
<td>63</td>
</tr>
<tr>
<td>A.8</td>
<td>A Two-Phases NBP-Based Localization Solution for Ambiguity Mitigation</td>
<td>72</td>
</tr>
<tr>
<td>A.9</td>
<td>Work in progress document for WHERE2 D2.1 Technical Report</td>
<td>85</td>
</tr>
<tr>
<td>A.10</td>
<td>Contributions on Hybrid Localization Techniques For Heterogeneous Wireless Networks</td>
<td>123</td>
</tr>
<tr>
<td>A.11</td>
<td>A Hybrid Data Fusion Based Cooperative Localization Approach for Cellular Networks</td>
<td>138</td>
</tr>
<tr>
<td>A.12</td>
<td>Accuracy Limits and Mobile Terminal Selection Scheme for Cooperative Localization in Cellular Networks</td>
<td>145</td>
</tr>
<tr>
<td>A.13</td>
<td>Cooperative Localization in a Distributed Base Station Scenario</td>
<td>151</td>
</tr>
<tr>
<td>A.14</td>
<td>Sensor Localization using Nonparametric Generalized Belief Propagation in Network with Loops</td>
<td>157</td>
</tr>
<tr>
<td>A.15</td>
<td>An Experimental Study of RSS-based Indoor Localization using Nonparametric Belief Propagation based on Spanning Trees</td>
<td>166</td>
</tr>
<tr>
<td>A.17</td>
<td>Optimized Edge Appearance probability for Cooperative Localization based on Tree-Reweighted Nonparametric Belief Propagation</td>
<td>182</td>
</tr>
</tbody>
</table>
A.1 Contribution to D2.1: ACORDE platform: ZigBee modules (distributed location algorithm embedded)

Jacobo Dominguez, Lorena de Celis (ACORDE) Contribution to D2.1: ACORDE platform: ZigBee modules (distributed location algorithm embedded)
Contribution to D.2.1: ACORDE platform: ZigBee modules
(distributed location algorithm embedded)

Jacobo Domínguez, Lorena de Celis (ACORDE)
January 30, 2012

1 ZigBee platform and location engine embedded overview

Acorde platform is based on a ZigBee architecture. It takes advantage of the location engine implemented, on hardware and software, by the chip manufacturer Texas Instruments (TI) [1].

This location algorithm embedded in the platforms follows a distributed approach. It uses the values of the received signal strength indicator (RSSI) from static reliable nodes, also known as anchor nodes. The nodes to be located are referred as blind nodes.

The location algorithm defines a location dongle which is a higher level functionally of a designed node that is in charge of coordinate the retrieve of location information. This location dongle usually is located in the coordinator of the ZigBee network.

Some advantages of a distributed algorithm are the reduction of network traffic and communication delays that are present in a centralized approach.

2 Algorithm optimization

The distributed location algorithm can be summarized in the following steps:

1. The anchor nodes need to be placed in the border of the area where the blind nodes are located.
2. A blind node broadcasts a request to all anchor nodes within range.
3. Anchor nodes respond to the blind node with their X/Y location and the RSSI of the blind node request.
4. The Blind node calculates its position based on the locations and RSSI values from the anchors nodes, and then sends the result to the location dongle.
5. The location dongle send this information to a GUI which is then able to provide the location of the Blind node to the user.

This algorithm can be optimized taking into account known parameters of the environment where the platforms are deployed. The basic information required by the location engine is the reference coordinates of the anchor nodes and a set of measured parameters. These parameters consist mainly of two radio parameters, four search boundary coordinates and 16 RSSI values. These radio parameters are used in the engines algorithm in order to find the estimated location. The radio parameters are the values A and np. The parameters A and np can be adjusted to describe the propagation environment in which a network of devices will operate and are defined as follows:

- **Parameter A (P_0 with d_0 =1meter):** The radio parameter A is defined as the absolute value of the average power in dBm, P_0, received at a close-in reference distance of one meter (d_0) from the transmitter, assuming an omni-directional radiation pattern. For example, if the mean received
power at one meter is -40 dBm, the parameter A is specified as 40. The Location engine expects the parameter A to be in the range [30.0, 50.0] with precision 0.5.

- **Parameter** $n_p$: The radio parameter $n_p$ is defined as the path loss exponent that describes the rate at which the signal power decays with increasing distance from the transmitter. This decay is proportional to $d^{-n_p}$ where $d$ is the distance between transmitter and receiver.

The parameters A and $n_p$ can be estimated empirically by collecting RSSI data (and therefore path loss data) for which the distances between the transmitting and receiving devices are known. Figure shown is a scatter plot of abs (RSSI) data versus log (distance) in meters.

![Figure 1: ZigBee scenario](image)

### 3 Practical Implementations and Experiments

Acorde contribution will be focused in the dynamical estimation of these two parameters in order to improve the accuracy of the position estimation [2]. One limitation of the TI Location Engine is that one pair of $[A, n_p]$ is used for all the communication links. This pair usually is also the same for all the locations. Acorde contribution will focus on dynamically select the best $[A, n_p]$ for each location.

The right selection of the values $[A, n_p]$ is one of the best ways to improve the current algorithm embedded in the ZigBee platforms.

The measurement campaign can be split in three phases:

1. **First stage**: Determine initial conditions. An initial value of $[A, n_p]$ in the selected scenario must be estimated. The following procedure describes the main steps that will be performed:
   a) Indoor scenario selection
   b) Parameter A calibration: It is usually obtained in a no-obstacle one-meter distance signal strength measurement from the reference nodes.
   c) Parameter $n_p$ characterization: Select one point in the scenario (center of the scene) and execute the current location algorithm with the ZigBee platform modifying parameter $n_p$ with the value for A parameter obtained in the previous step.
   d) Initial condition selection: Process the data obtained and selects the best pair $[A, n_p]_0$ to be used as initial conditions in the selected scenario.
2. Second stage: Scenario characterization: Once the initial conditions have been obtained, the entire scenario will be characterized with different values of the pair $[A, n_p]$ for different sections/areas. In a first stage, parameter $A$ will be a fixed value, the one obtained in previous step. Next are listed the actions that will be followed in order to estimate the best $n$ value.

   a) Build a grid in the scenario in order to divide it smaller areas. This grid could have a smaller step in more complicated geometry locations. Different options to perform the partitioning are possible as shown in Figure 2. Also defining different level of detail the accuracy could be improved.

   

   Figure 2: Parameter $A$ calibration (RN = reference node)

   b) Selection of one area, $S_i$

   c) Obtain for each position:
      
      i. Multiple RSSI values from each Anchor node.
      ii. A pair $[A, n_p]_i$ that best describes the radio parameters for this area.

   d) Estimate the best value of $n_p$ for the selected area $S_i$. The selection criteria will be study. A first proposal for this criteria is the one that minimize the distance error vector of the estimated positions and the know position or a mean value of all the obtained values. Therefore several measurements are needed in the same the same area that is being analyzed $S_i$.

   Figure 3: Scenario characterization using different partitioning criteria

3. In case of large scenarios with heterogonous areas different spatial partitions will be analyzed. The possibility of use multiple pairs $[A, n_p]$, to define different levels of detail will also be study.

4. When the scenario areas are characterized a dynamic selection of these values will be implemented in order to improve the location estimation. The dynamic selection will be a iterative process that will start using the pair $[A, n_p]_0$ a for each location estimation will select the best parameters $[A, n_p]$, estimated in step 2.d.
The measurement campaign needed to accomplish step 2, will also be studied using external off-line processing tools.

In order to evaluate deeply the location engine, all the tests will be performed in different environments. Some of them could be:

- Static obstacles conditions.
- Dynamic conditions, e.g. office with working employees.
- With several wireless 802.11 networks.

References


A.2 A New Variant of Nonparametric Belief Propagation for Self-Localization

Hadi Noureddine, Nicolas Gresset, Damien Castelain, Ramesh Pyndiah

A New Variant of Nonparametric Belief Propagation for Self-Localization

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
A New Variant of Nonparametric Belief Propagation for Self-Localization

Hadi Noureddine∗†, Nicolas Gresset∗, Damien Castelain∗, Ramesh Pyndiah†
∗Mitsubishi Electric R&D Centre Europe, France
†Telecom Bretagne, France

Abstract—We consider the problem of relative self-localization of a network of fixed communicating devices that evaluate range measurements between each other. The solution is obtained in two stages: First, a new variant of the Nonparametric Belief Propagation algorithm is used for estimating the beliefs. This variant is based on a Monte-Carlo integration with rejection sampling where a delimited space region is determined for solving the flipping ambiguities resulting from the lack of measurements. This solution has the advantage of reducing the amount of communicating particles and the computation cost.

I. INTRODUCTION

Finding the positions of a set of wireless communicating devices has a lot of practical applications, going from the deployment of ad-hoc networks and its related topics, e.g. communication enhancement and location-based routing, toward a variety of location-based services and applications, e.g. military, environmental and health.

The communicating devices may take several forms, such as sensors, femto cells, access points, etc., with indoor or outdoor deployment. They might be subject to several constraints on their size, power consumption and price. Thus, developing GPS-free localization techniques is capital.

If pairs of nodes perform measurements relevant to their relative positions, they can be localized in a coordinate system. Several names have been attributed to this topic in the literature, such as ‘network calibration’, ‘cooperative’ and ‘self-localization’.

Several centralized and distributed algorithms have been developed in order to solve this localization problem. For centralized algorithms, measurements are collected to a central processor where the overall processing is done. One example of such an algorithm is the ML estimation [1][2], which can be applied if the statistical model of the measurements is known.

In distributed algorithms, all the nodes are involved in the estimation process, and the computation is distributed among them. These algorithms are most useful for large networks. In [3], a node estimates its distance to several reference nodes according to the number of hops of the shortest connection path. The positions are then found by multilateration. In [4][5], successive refinement is processed, where one node refines and sends its estimate to its neighbors at each iteration. The Belief Propagation (BP) algorithm is based on probabilistic graphical models [6][7] where each node calculates the probability density function of its coordinates, based on prior information, measurements and probability densities provided at each iteration by neighboring nodes. This algorithm produces both an estimate of locations and metrics of uncertainties.

In this paper, we are interested in Nonparametric Belief Propagation (NBP)[8], which is a particle-based version of the BP.

In the first phase of our solution, NBP is implemented using a Monte-Carlo integration instead of the stochastic method of [9]. Furthermore, the samples are selected from the beliefs by using rejection sampling. The errors on measurements are supposed to lie in known intervals, which allows for constructing limited space regions for each node and thus reducing the rejection ratio.

In the second phase, we propose an algorithm for mitigating the flipping ambiguities that result from the lack of measurements. This algorithm is based on K-means clustering and estimation in discrete-valued states space, and has the advantage of drastically reducing the computation complexity and the amount of data to be exchanged.

The rest of the paper is organized as follows: In section II, the problem is formulated as a graphical model, and our implemented variant of the NBP is presented. In section III, we present a new method for solving the flipping ambiguities remaining after convergence of the NBP algorithm. Simulation results and conclusions are provided in sections IV and V.

II. PROBABILISTIC GRAPHS APPLICATION TO SELF-LOCALIZATION

In this section, we consider the belief propagation applied to self-localization. We assume that we have $N$ fixed nodes scattered in a planar space, and only consider 2D localization. In addition, we only consider relative localization as no node knows its absolute position.

Each node obtains distance measurements with the set of its neighboring nodes, and these measurements are corrupted by an additive error. Nodes are mutually neighbors, and the relationship between the nodes can be described by an undirected graph.

We assume that neighboring nodes share the same observation on their distance. Let $x_i$ denote the two-dimensional position of node $i$, and $\tilde{d}_{ij}$ the noisy distance measurement with its neighbor $j$. The joint a posteriori probability distribu-
tion factorizes as:

\[ p \left( x_1, \cdots, x_N / \{ \hat{d}_{ij} \} \right) \propto \prod_{i \in V} \Phi_i(x_i) \prod_{j \in \Omega(i), i < j} \Psi_{ij}(x_i, x_j) \]  

(1)

where \( V \) is the nodes set and \( \Omega(i) \) is the set of neighbors of node \( i \). \( \Psi_{ij}(x_i, x_j) = p_i(x_j) \) is a pairwise potential function and \( \Phi_i(x_i) \) is the a priori probability on the location of node \( i \).

Two approaches are possible for estimating the positions of the nodes:

- Find the joint maximum a posteriori (MAP) of all \( x_i \)'s, or in other words, the sequence of states \( \{ x_i \} \) maximizing (1). For example, the Max-Product algorithm finds this most likely sequence of states.
- Find the MAP of each \( x_i \) apart. This can be done by a marginalization of (1) so as to obtain the a posteriori probability distribution at each node. For example, the Sum-Product algorithm is a way for evaluating the marginalization.

In the following, the Sum-Product is considered so as to determine the belief of each node for a given position, and is described in the following subsection.

A. Belief Propagation

The previously described model can be qualified as a probabilistic graphical model, in which a node represents a random variable or a parameter to be estimated, and an edge expresses the existence of a probabilistic relationship, or a compatibility, between two nodes.

Belief Propagation (BP) is an iterative message passing algorithm that calculates the posterior marginalization at each node. At the \( n \)-th iteration, each node computes its belief by taking the product of its local potential and the incoming messages from its neighbors as follows:

\[ \hat{p}^{(n)}(x_i) \propto \Phi_i(x_i) \prod_{k \in \Omega(i)} m^{(n)}_{ki}(x_i) \]  

(2)

The message from node \( j \) to node \( i \), called the update rule, is:

\[ m^{(n)}_{ji}(x_i) \propto \int \Phi_j(x_j) \Psi_{ij}(x_i, x_j) \prod_{k \in \Omega(j) \setminus i} m^{(n-1)}_{kj}(x_j) dx_j \]  

(3)

where \( \Omega(j) \setminus i \) is the set of neighbors of \( j \) except \( i \). All messages are initialized to an arbitrary value, for example 1.

In the case of graphs without loops, it is known that this algorithm perfectly computes the marginal probability distributions, and the needed number of iterations is equal to the graph diameter. If loops occur in the graph, good approximations of the marginal probability distributions are observed under some conditions [10].

The integral equation (3) can be evaluated when the variables are discrete valued or in the case of Gaussian distributions. When these conditions are not fulfilled, the integral equation rarely has tractable analytic solution and must be replaced by an approximation, such as for the Nonparametric Belief Propagation. As relative positioning is considered here, \( \Phi_i(x_i) \) will be dropped from the equations given above.

B. Nonparametric Belief Propagation

Nonparametric Belief Propagation [8] is based on stochastic methods for propagating kernel-based approximations of the continuous messages. In this algorithm, we propagate a set of values \( \{ r_{ji}^{(l)} \}_{l=1}^M \) from node \( j \) to node \( i \), where \( r_{ji}^{(l)} \sim \Psi_{ij}(x_i, s_{ji}^{(l)}) \) is a sample taken from \( \Psi_{ij} \) for a position sample \( s_{ji}^{(l)} \) of node \( j \). The set of position samples \( \{ s_{ji}^{(l)} \}_{l=1}^M \) are drawn from the beliefs with an association of weights. The message \( m_{ji} \) is then formed by placing identical Gaussian kernels about the points \( \{ r_{ji}^{(l)} \} \) which requires an appropriate choice of the kernel covariance matrix. The belief function, computed by taking the product of the incoming messages, becomes a Gaussian mixture with a huge number of components. In the case where the potentials are Gaussian mixtures, [11] proposed to use Monte-Carlo integration for estimating the message equation (3).

In relative positioning, we consider that each node lies in a known limited region of space. This region is obtained by assuming that the measurement error is constrained to a known interval, with a good probability. This allows for the application of rejection sampling in drawing independent samples. Thus, we propose to perform a Monte-Carlo integration of equation (3) without resorting to Gaussian mixtures approximations and kernel covariance matrix choice.

A Monte-Carlo integration of equation (3) yields \( \hat{m}_{ji} \), an approximation of \( m_{ji} \), by drawing \( M \) samples \( \{ s_{ji}^{(l)} \}_{l=1}^M \) from \( p_{ji}^{(n)} \) defined as:

\[ p_{ji}^{(n)}(x_j) \propto \prod_{k \in \Omega(j) \setminus i} m^{(n-1)}_{kj}(x_j) \]  

(4)

which can be considered as a probability density function. In general, we can draw the samples from any density function \( g_{ji}(x_j) \) that does not vanish when \( p_{ji}^{(n)} \) does not. The message \( \hat{m}_{ji} \) then becomes the weighted mixture:

\[ \hat{m}_{ji}^{(n)}(x_j) = \frac{1}{\sum_{k=1}^M \pi_{ji}^k} \sum_{l=1}^M \pi_{ji}^l \Psi_{ij}(x_i, s_{ji}^{(l)}) \]  

(5)

where \( \pi_{ji}^l = p_{ji}^{(n)}(s_{ji}^{(l)})/g_{ji}(s_{ji}^{(l)}) \) is the weight associated to sample \( s_{ji}^{(l)} \). We choose \( g_{ji}(x_j) \) equal to the belief of node \( j \):

\[ g_{ji}^{(n)}(x_j) \propto \prod_{k \in \Omega(j)} \hat{m}_{kj}^{(n-1)}(x_j) \]  

(6)

This function is the same for all \( i \in \Omega(j) \).

When compared to existing relative positioning techniques using NBP [9][12], we propagate the generated samples \( s_{ji}^{(l)} \) from the node \( j \) to all its neighbors and sampling at node \( j \) is done only once. Thus, we don’t have to sample from the different potentials. Furthermore, we don’t have to estimate densities for the relative directions, in order to concentrate the samples in regions of interest, as done by [9] in order to
alleviate the fact that potentials do not contain information on the directions.

As a remark, the associated weights $\pi_{ij} = 1/\tilde{m}_{ij}^{(n-1)}(s_{ij})$ could be calculated locally at node $i$ and not propagated.

III. DEALING WITH AMBIGUITIES

The problem of relative localization can be resolved up to congruence, i.e., translation, rotation or reflection of the whole network. Firstly, we define the node 1 as the origin in order to remove this ambiguity. We attribute the coordinates vector $x_1 = (0, 0)^T$ to this node. Secondly, the node 2 is set on the positive half of the x-axis, $x_2 = (x_2, 0)^T$ and $x_2 > 0$. Finally, the node 3 is set in the half plane with positive y-component, $y_3 > 0$. After verifying these three conditions, the region of the space where each node can lie can be determined based on the hypothesis that measurement errors are bounded.

It is important to understand the conditions under which the problem is solvable. For example, in the case of lack of measurements, the network can be subject to deformation as shown in Fig. 1(a), where any rotation of the two pairs of two points on the extrema around the center points lead to a possible solution. A sufficient condition for obtaining a unique solution is to observe a globally rigid graph of the network [13][14]. In this paper, we only consider rigid graphs, where discontinuous deformations are possible as shown in Fig. 1(b). An efficient algorithm for testing graph rigidity [15], called The Pebble game, is implemented in our simulations. This algorithm can also identify all rigid subgraphs and its complexity is at most $O(N^3)$.

A. Flipping Ambiguities

Discontinuous deformations create a kind of ambiguity on the solution that we call flipping ambiguity. It follows that the beliefs of some nodes occur to be multimodal. In Fig. 4, a network of 7 nodes is plotted altogether with the region of each node. Nodes 5 and 7 can be flipped, resulting in 4 possible solutions for node 7. Thus its belief has four modes as is illustrated in Fig. 5.

In this subsection, we present the state-of-the-art for solving ambiguities, as introduced in [9]. The fact that two nodes are not neighbors gives the additional information that they are probably far one from the other. This information will be exploited for solving ambiguities. We note $P_o(x_i, x_j)$ the probability for two nodes to be neighbors one of the other. This probability is a function of the communication channel quality, and mainly of the distance between the two nodes. We do not investigate thoroughly the communication performance

and consider the following simplified model for $P_o(x_i, x_j)$ [9]:

$$P_o(x_i, x_j) = \exp \left( -0.5 \|x_i - x_j\|^2 / R^2 \right)$$ \hspace{1cm} (7)

In [9][12], $P_o$ is included in (1) and in the NBP exchanged messages that occur with direct and ‘2-step’ neighbors. The ‘2-step’ neighbors of node $i$ are the set of neighbors of neighbors of $i$ (except $i$). For the latter, potentials are taken as $1 - P_o$.

One drawback of this method is the complexity and overhead of exchanged messages. Indeed, if the nodes perform a broadcasting, the number of broadcast operations at some nodes should at least be doubled before the messages reach ‘2-step’ neighbors. For example, the message from node $a$ in Fig. 2 has to be broadcast three times before attaining nodes $b$ and $c$. Furthermore, samples drawing and beliefs computation become more complicated, as the latter are constructed by taking the product of all incoming messages, whether from direct or ‘2-step’ neighbors. Fig. 3 shows the average number of neighbors in rigid networks. Nodes are drawn uniformly in an $L \times L$ square, and the connectivity is constructed according to (7) and independently for each pair of nodes. It shows that the number of neighbors is much increased when considering direct and ‘2-step’ ones, especially when $R/L$ is small.

B. A clustering-based disambiguating algorithm

We propose a new algorithm for solving flipping ambiguities which reduces both the exchanges overhead and computation complexity. It is applied in a second phase after finding the
beliefs with the NBP, during which we considered only direct neighbors. The algorithm is composed of the following steps:

1) We first identify the different beliefs modes. In order to do so, we apply K-means clustering [16] on the samples, which is particularly relevant as the samples tend to be concentrated around the modes. As a remark, the farther we go from the node 1 located on the origin, the higher will be the number of possible flips. We propose to take the number of clusters proportional to the smallest number of hops to node 1. We can also take a constant overestimated number of clusters. Other methods for automatically determining the number of clusters from the samples are described in [17].

2) For each cluster, we retain only the sample that has the maximum belief. For example, in Fig. 6, clustering is done for the samples of node 7, where four clusters are considered.

3) At this point, each node will have a small set of points that include the belief’s modes. We apply a discrete version of the BP to find, again, the beliefs of these retained points, with involving the ‘2-step’ neighbors this time.

- We can use the Sum-Product rules, and in that case the messages are:

\[ m^{(n)}_{ij}(s^q_i) = \sum_{l=1}^{[S_i]} \Psi_{ij}(s^q_i, s^q_j) \prod_{k \in \Omega_2(j)} m^{(n-1)}_{kj}(s^q_j) \]

where \( S_i \) is the set of retained points of node \( i \) after clustering, \( q = 1, \ldots, [S_i] \) and \( \Omega_2(i) \) is the set of ‘2-step’ neighbors of \( i \). We compute \( \Psi_{ij}(s_i, s_j) = p(d_{ij}/s_i, s_j)P_o(s_i, s_j) \) for direct neighbors, and \( \Psi_{ij}(s_i, s_j) = 1 - P_o(s_i, s_j) \) for ‘2-step’ ones.

- If the Max-Product is used instead, the messages are:

\[ m^{(n)}_{ij}(s^q_i) = \max_{l=1}^{[S_i]} \Psi_{ij}(s^q_i, s^q_j) \prod_{k \in \Omega_2(j)} m^{(n-1)}_{kj}(s^q_j) \]

4) The beliefs at node \( i \) are computed with:

\[ \hat{B}^{(n)}(s^q_i) = \prod_{k \in \Omega(i)} m^{(n)}_{ki}(s^q_k) \]

5) The estimated position is taken as the point with the maximum belief:

\[ \hat{x}_i = \arg\max_{s^q_i \in S_i} \hat{B}^{(n)}(s^q_i) \]

With this algorithm, the ‘2-step’ neighbors are implicated in the message exchange process, but the amount of data contained in the message is much smaller than that of the first phase NBP.

For the network of Fig. 4, the estimated positions are plotted in Fig. 7. The crosses represent the estimates from the NBP without applying the second stage algorithm. Circles represent the estimates after the second stage of disambiguating where the Max-Product algorithm is applied.

---

**Fig. 4.** A network of 7 nodes. The region of each node is represented with a color. Nodes 5 and 7 can be flipped causing an ambiguity.

**Fig. 5.** A contour plot of the belief of node 7 after 4 iterations. It has 4 modes.

**Fig. 6.** Clustering of the samples of node 7. A color is attributed to each cluster. These samples are drawn from the belief after 4 iterations.
### IV. SIMULATIONS

To measure the performance of the localization algorithm, we use a metric called Global Energy Ratio (GER) [18], given by (12).

$$\text{GER} = \sqrt{\frac{\text{mean} \left( \sum_{i<j} \tilde{e}_{ij}^2 \right)}{N(N-1)/2}}$$  \hspace{1cm} (12)

where $\tilde{e}_{ij} = (\hat{d}_{ij} - d_{ij})/d_{ij}$ is the normalized error, $d_{ij}$ is the true distance and $\hat{d}_{ij}$ is the distance in the algorithm’s result. This metric measures the performance compared to the true configuration topological properties, by taking into account all the distances, whether measured or not. The method developed in the previous section is compared to the ML estimate. In order to make a fair comparison, the density function $P_o(x_i, x_j)$ is included in the joint probability distribution. We also include $1 - P_o(x_i, x_j)$ for the ‘2-step’ neighbors.

The Pebble Game algorithm is implemented to identify the rigid graphs, for which the localization is done. The measurement errors follow a truncated Gaussian distribution, with variance $\sigma$, and interval $[-a, a]$. The region of space where each node can exist is determined according to the measurements.

Note that truncated Gaussian potentials will cause that the belief given by (6) will have a support different than that of the density function (4). To circumvent this problem, we make a relaxation of the potentials by taking them as Gaussian during the application of the NBP and discrete BP algorithms.

In Fig. 8, the GER is plotted vs the iteration number for rigid networks of 10 nodes. The error interval is $[-7, 7]$ and the variance is 3. The connectivity is established according to (7) with $R/L = 0.2$ and independently for each pair of nodes and $L \times L$ is the total considered area. It shows that the GER is much better in the cases where disambiguiting is applied. It also shows that the Max-Product algorithm performs better than the Sum-Product.

In Fig. 9, the average rejection ratio for rigid networks of 10 nodes is plotted. It shows that this ratio increases with iteration number. This can be justified by the fact that beliefs become more tightened around their modes as the nodes gather more information about their locations. Here, the samples are drawn uniformly from the determined regions before applying the rejection test.

### V. CONCLUSIONS

In this paper, we presented the problem of sensor networks localization with a use of a graphical model. We also presented nonparametric belief propagation (NBP), and applied a new variant of this method, which is based on a Monte-Carlo estimation of the propagated messages. We used rejection sampling to draw samples from the nodes’ beliefs. These
samples are taken from determined regions of the space where the nodes can exist. Other sampling methods can be investigated, such as the Metropolis-Hastings method [19], which is a Markov Chain Monte-Carlo method (MCMC), and the Ziggurat method [20]. If the rejection ratio is very high, rejection sampling can be applied to some of the samples. Then these points are considered as kernels centers, and the remaining samples are drawn from the probability density represented by the kernels.

Due to the fact that rigid graphs are subject to discontinuous deformations, ambiguities on nodes positions may exist. To deal with them, we proposed an algorithm, based on K-means clustering, which reduces both the communication and computation cost. The estimation result is much improved with this algorithm and the result is good in comparison to the ML.

VI. ACKNOWLEDGEMENT

The work presented in this paper has been performed in the framework of the ICT-248894 WHERE2 project funded by the European Union.

REFERENCES

A.3 Utility Based Node Selection Scheme for Cooperative Localization

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Senka Hadzic and Jonathan Rodriguez Utility Based Node Selection Scheme for Cooperative Localization
Utility Based Node Selection Scheme for Cooperative Localization

Senka Hadzic\textsuperscript{1,2} and Jonathan Rodriguez\textsuperscript{1}
\textsuperscript{1}Instituto de Telecomunicações
\textsuperscript{2}Universidade de Aveiro
Aveiro, Portugal
\{senka, jonathan\}@av.it.pt

Abstract— Cooperative positioning is a popular approach for indoor localization, which significantly outperforms conventional positioning techniques. This paper presents a method for iterative node localization, with a relatively small number of initial anchor nodes. In order to reduce error propagation in iterative schemes, it is necessary to use only reliable information among nodes. We introduce a reference node selection strategy based on utility functions. The method is completely distributed and involves only information from local neighborhood. Compared to the nearest neighbor and random node selection scheme, simulation results show that selecting subsets of nodes with highest utility values leads to more accurate position estimates.

Keywords- multilateration; utility; node selection

I. INTRODUCTION

Localization techniques rely on internode measurements and distance estimates to fixed anchor nodes with known coordinates. Cooperation between nodes is used in cases when conventional positioning techniques do not perform well due to lack of existing infrastructure or obstructed indoor environment. Both cooperative and conventional techniques usually consist of two stages: 1) a ranging phase where nodes estimate distances to their neighbors by measuring some distance dependent signal metric, and 2) an algorithm where the ranging information is used for position calculation. The most common ranging techniques are based on received signal strength (RSS) and time of arrival (TOA). Both are prone to noise and therefore there is a need for more advanced algorithms to compensate for the low performance of measurement techniques.

One widely used method for position calculation is least squares optimization. In the ideal case, the coordinates of the unknown node would correspond to the point of intersection of at least three circles with center in anchor node’s coordinates and radius equal to distance to each reference node. Due to erroneous distance estimates these circles do not intersect in one single point, and least squares optimization is applied to minimize the sum of squared residuals. Hence it leads to a nonlinear optimization problem which requires appropriate initial estimates and is considered too expensive [1].

Additionally, if the variance of distance measurements from each anchor node is available, a weighted least squares solution can be attained, using the inverse of the variances as weights. Least squares approach has been adopted in cooperative positioning schemes in [2]-[6]. While all these algorithms are deterministic, i.e., their aim is to find the deterministic location, statistical methods such as belief propagation [7] and factor graphs [8] aim to estimate the maximum a posteriori location using a set of observations and a priori probability distributions of node locations. Several centralized solutions have been proposed such as multidimensional scaling [9] and convex optimization [10].

An alternative approach to nonlinear least squares is to use linearized expressions and calculate the position estimate by means of linear least squares (LLS) approach. It is not an optimal estimator but yields a low complexity solution with reasonable accuracy. Basically, one of the available anchor nodes is selected as a reference, and the nonlinear expressions corresponding to it is subtracted from expressions associated with remaining anchor nodes. By this means the nonlinear term is eliminated, and a simple LS matrix solution provides the location of the unknown node. In [1] it has been shown that LLS localization performance can be enhanced by appropriate selection of linearization reference, instead of choosing it arbitrarily.

In particular, we will consider a distributed localization approach, namely iterative multilateration. Once an unknown node estimates its position, it becomes an anchor and broadcasts its position estimate to all neighboring nodes. The process is repeated until all nodes that can have three or more reference nodes obtain a position estimate. It only involves information within local neighborhood and hence reduces communication cost. However, it suffers from error propagation. As a newly localized node is becoming a new anchor for its neighbors, the estimation error of the first node can propagate to other nodes and eventually get amplified. Excessive iterations could lead to widespread error distribution throughout the network, leading to abundant error in large topologies. The effect may also arise in global methods such as MDS or SDP, but the global constraints are likely to balance against each other and hence make global methods less vulnerable. Hence it is important to choose reference nodes carefully and hereby reduce error accumulation by taking into...
account uncertainties in reference nodes estimates. The procedure is illustrated in Figure 1, where the target can localize itself using neighboring anchor nodes, but also using so called virtual anchors, i.e., nodes that have obtained their location estimates in previous iterations. Virtual anchors have different degrees of uncertainties in their location estimates.

![Figure 1: Iterative multilateration](https://example.com/figure1.png)

The rest of the paper is organized as follows. In Section II we present several approaches to the node selection problem in cooperative positioning. Section III gives an overview on factors that contribute to localization error. In Section IV we describe our node selection method based on utility functions, and in Section V we present simulation results to depict the performance of our proposed method. Conclusions are given in Section VI.

II. RELATED WORK

One critical aspect in cooperative positioning is to select the set of links that will lead to higher accuracy. The use of all available links would increase the computational complexity and positioning latency. The number of cooperating nodes should be kept to a minimum, and therefore an adequate subset has to be chosen, while the others can be ignored. Such a restrictive and selective use of references is crucial in networks with limited resources. Furthermore, using all existing reference node does not necessarily improve the localization performance, as some positional information may be useless and even harmful.

In [2] an algorithm for discarding of unreliable links has been proposed. In order to determine which pairwise distance estimates can be removed, the conditional Cramér Rao Lower Bound (CRLB) of positioning errors is analyzed. Localization is performed in two phases; connectivity-based coarse positioning provides an initial estimate for the following refined procedure, namely Time of Arrival (TOA) based ranging for cooperative positioning. The first phase is performed almost for free during network initialization, and using the obtained coarse estimates a conditional CRLB can be computed. The CRLB is evaluated again while keeping removing several combinations of links, and the combination that provides highest accuracy is chosen for the second phase. In this manner resources are saved as the number of packet exchanges in the second phase is reduced. Nevertheless, links can only be removed after receiving information from neighboring nodes. In [4] the algorithm has been extended in a way that it includes both transmit and receive censoring. Transmit censoring prevents unreliable position estimates to be broadcasted, while receive censoring eliminates useless links after collecting information. Those distributed censoring schemes applied to a cooperative least squares algorithm improve the positioning performance while significantly reducing network traffic.

Besides the use of theoretical localization performance limits, e.g., CRLB, as selection criteria, concepts from coalitional games and utility functions have also been adopted to the node selection problem. Given a set of candidate nodes $S = \{S_1, \ldots, S_n\}$, the problem is to determine the subset $S'$ of $k < N$ nodes, which is referred to as the ‘best set’. The ‘best set’ is the one which achieves a tradeoff between energy constraints and quality of information with respect to positioning accuracy. This tradeoff can be modeled using the notions of utility and cost. Hereby ‘utility’ refers to the accuracy of collected information and its usefulness (i.e., by means of the mean square error), while ‘cost’ is associated with energy spent during the performed task. In [11] and [12], methods based on minimization of mean square error (MSE) are discussed. Localization error bounds based on the node geometry are derived. Basically, selecting nodes close to the target decreases the lower bound, while choosing nodes that provide angular diversity helps the localization error reach the lower bound. Utility of each set consisting of $N_i$ nodes is defined as the reciprocal of the mean square error, and the differential utility (for each node $i$) denotes the utility that the $i$-th node provides to the remaining $N_i - 1$ nodes. In general, the differential utility depends also on the other nodes within the active set, since geometry is considered. It is proved that the differential utility also has a lower bound, and this fact is used to prune the search methods. In this way the global selection procedure is far less computationally intensive than using exhaustive search. The global selection method in [11] requires global knowledge of sensor location, and this limitation is overcome in the autonomous node selection scheme [12]. Both schemes are implemented in an extended Kalman filter, where the predicted target state is used to determine the position of candidate nodes relative to the target.

The work of Ghassemi in [13],[14] uses notions from game theory and the value of each node for localization is modeled using the Shapley value. The level of accuracy depends on the selected nodes, and since this level cannot be known before measurements are obtained, the objective function is the predicted level of accuracy, quantified by the determinant of the posterior CRLB. The localization procedure is represented as a weighted-graph game. This allows an efficient method for computing the Shapley value, which usually involves computational complexity that increases exponentially with number of nodes. The scheme is integrated in a particle filter. At each decision instant the posterior target density is computed, approximated with a Gaussian distribution and used as the prior density at the following instant. In [14] utility is
defined as information gain from a node, i.e. the mutual information between the prior density of target position and the measurement. Additionally, a price for transmission is included to account for the current energy level in the nodes, and the energy needed for data transmission.

An entropy based node selection scheme has been proposed in [15], adopting heuristics instead of the computationally intensive optimal solution. Another possible approach is to solve the sensor selection problem through multiobjective optimization. In [16] the selection is done for a given budget with multiple objectives such as sensor network lifetime, coverage etc.

III. LOCALIZATION ERROR

Localization error is a function of several factors, such as number of anchor nodes, node density, network topology etc. In addition to noisy distance estimates and reference node geometry, the error propagation problem is also resulting from use of erroneous estimates as virtual anchors in subsequent iterations. An unknown node receives information from many neighbors, some of which are virtual anchors with a degree of uncertainty in their estimates. Therefore not all of those links have the same level of “usefulness”, even if localization accuracy increases with the number of used reference nodes, from the information theory perspective. Especially the geometry of used reference nodes has been shown to have a high impact on lateration.

The geometric conditioning on localization accuracy is derived in the GDOP (geometric dilution of precision) metric [17]. A commonly used tool to describe the error bounds on location estimates is the Cramér Rao Lower Bound, which will be explained in detail in ILB. In [18] it was assumed that the a priori knowledge of reference node locations comes from successive localization (e.g., iterative multilateration), and is usually imperfect. It is shown that the effect of inexact location knowledge of reference nodes on error bounds is equivalent to the increase of variance of RSS-based distance estimation.

A. RSS based distance estimation

In case of received signal strength, ranges are first estimated and then LS techniques are applied on these ranges. We use the standard lognormal model for RSS with path loss parameter $n_p$ and shadowing variance $\sigma^2_{RSS}$. Assuming that the received power $P_{ij}$ between nodes $i$ and $j$ is lognormal, the random variable $P_{ij}$ (dBm) = $10 \log P_{ij}$ is Gaussian and the maximum likelihood distance estimate with bias correction is given by [19]:

$$d_{i,j} = d_0 10^{\frac{P_i-P_j}{10n_p}} e^{-\frac{\gamma^2}{2}},$$

(1)

where $\gamma = \frac{\sigma_{RSS} \ln 10}{10n_p}$ and $e^{-\frac{\gamma^2}{2}}$ is a multiplicative bias correction factor. By this means, distance estimates are obtained using a channel model with known parameters.

B. Cramér Rao Lower Bound

The Cramér Rao Lower Bound provides a lower bound on covariance of any unbiased estimator. It is calculated as the inverse Fisher Information Matrix (FIM). The Fisher information is a way of measuring the amount of information that an observable random variable carries about an unknown parameter upon which the likelihood function depends. Considering the log-likelihood function of random measurements $f_{RSS}$ the FIM is given by

$$FIM = E\left[\nabla f_{RSS} \cdot \nabla f_{RSS}^T\right],$$

where $\nabla f_{RSS}$ is the gradient of the log-likelihood function, and $E[\cdot]$ denotes the expectation operator. It can be shown that the CRLB for RSS based distance estimates for multilateration with $N$ anchor nodes will be of the form [19]:

$$CRLB_{RSS} = \frac{1}{b} \frac{\sum_{i=2}^{N} d_{1,i}^2}{\sum_{i=2}^{N} \sum_{j=i+1}^{N} \left(d_{i,j}^2 \sigma^2_{RSS} \ln 10\right)} ,$$

(3)

where $b = \left(\frac{10n_p}{\sigma_{RSS} \ln 10}\right)^2$.

CRLB captures information about node geometry and channel conditions (ranging quality). For diverse topologies the error bounds will be different. Since the variance of position estimates is associated to the mean error, the lower bound on variance can be seen as the upper bound on accuracy. To illustrate the motive for using CRLB for node selection, we show in Figure 2 a target T trying to select the best three neighbors out of 6 that will be used for positioning. Considering the links individually, anchor nodes A1, A2 and A3 might be the best ones, however with respect to target’s location nodes A2 and A3 cannot contribute any additional information, as they all provide information in one direction. The same holds for anchor nodes A5 and A6.

![Figure 2: Geometric impact on node selection](image-url)
The geometric relation of selected anchor nodes affects localization performance [20].

IV. NODE SELECTION

As concluded in the previous section, the output of a multilateration procedure is varying if different sets of references are chosen. Here we will focus on determining which combination of references results in best performance, in case when alternative reference nodes are available. In order to find a proper model that considers all the factors contributing to localization error, we will adopt some concepts from game theory and utility functions to the node selection problem.

Let us first review some concepts from game theory and see how these concepts can be adapted to the localization problem.

A. Utility function

A node may prefer to collaborate with a subset of nodes to form coalitions. A coalitional game consists of a set of players, an action set (strategy) for each player and a utility (payoff) for each player, measuring its level of satisfaction by assigning a value to a coalition. The players assess the usefulness of their strategies using their utility functions. The utility depends not only on a single node’s strategy, but also on the strategies by other nodes forming a coalition. In [21] a generic approach for coalition formation has been proposed, that has been applied to wireless communications, mostly in resource allocating and cognitive radio.

A cooperative game is the pair \((N, v)\) where \(N\) is a finite set and \(v\) is its utility function. The elements of \(N\) are the players and any non-empty subset \(C \subseteq N\) is a coalition. In particular, \(N\) is called the grand coalition.

The main challenge here is to choose the appropriate utility function, i.e., how a node values different levels of performance. We can formulate the node selection optimization as the one that maximizes the accuracy subject to constraints given by nodes’ limited processing capacity. The following parameters are relevant for reference node selection: number of references, their uncertainty (in case of virtual anchors), quality of range estimates and geometry. We assume that in each coalition exists a data fusion center which acts as coalition head [22]. We choose the closest anchor node, which is also used as reference for linearization [1] to serve as coalition head. Cooperation involves some cost per each anchor node \(c_i\), associated with distance to the data fusion center \(d_{i,f}\), but also yields a benefit \(B(C)\) in terms of improved accuracy when using a particular subset of nodes. Knowing that at least three nodes are needed in order to perform localization, we set the coalition value to zero for all subsets containing less than three elements.

Assuming that the communication range is \(R\), we define the utility function as:

\[
v(C) = \begin{cases} 0, & |C|<3 \\ \frac{1}{\text{CRLB}_{ic}} \sum_{i \in C} \frac{d_{i,f}}{R}, & \text{otherwise} \end{cases},
\]

where the first term is the benefit indicator, while the second term represents the cost function related to the energy consumption required for communication. A lower CRLB indicates higher accuracy, and therefore it is inversely proportional to the benefit. The cost function is formulated using the fact that distant nodes spend more energy for communication, and accordingly the cost is proportional to distance. Since the true position of the unknown nodes is not accessible, the CRLB will be calculated using the estimated positions.

B. Selection procedure

The simplest method is to perform an exhaustive search that evaluates the coalition value (4) for all possible sets of size \(\text{card}(C)\), and then choose the set with the largest coalition value. This method is guaranteed optimal but the search time is exponential and the number of combinations is very large. It may be quite acceptable for small \(N\). Pruned search methods were proposed in [11] to reduce the number of computations. Considering a low density scenario, and thereby a rather small number of candidate nodes, we will illustrate the algorithm performance using exhaustive search. In our future work we aim to consider other, suboptimal search methods that will reduce computational complexity, without significantly degrading the results achieved by the proposed scheme.

V. SIMULATION RESULTS

Let us assume a network consisting of \(N\) nodes, \(M\) of which are initially anchor nodes, and \(K = N - M\) remaining ones are unknown nodes. For evaluation we will consider one snapshot in the iterative algorithm, where a node analyzes its local neighborhood, and exchanges information with \(N_p\) reference nodes within communication range \(R = 30\) m. Among the candidate references the goal is to choose three of them, \(N_p = 3\), which provide best performance. In our simulations, we assume that a node has 10 available candidate nodes, randomly placed within communication range of the unknown node, and 5 of them are assumed to be virtual anchors. Position is calculated using the LLS algorithm, using erroneous distance estimates as in (1), and choosing the closest anchor nodes as reference for linearization. Ranging error is modeled using channel parameters \(n_r = 2.3\) and \(\sigma_{RSS} = 3.92\) dB, as in [19].

In this work we assume a random, independent ranging error and all links have same channel conditions. We model the uncertainty of virtual anchors by associating a variance \(\sigma^2\) to the true position. The number of possible combinations is:

\[
\frac{N_a^!}{N_b^!(N_a - N_b)^!},
\]

(5)
where \( N_a \) is the total number of available reference nodes, and \( N_b \) is the number of nodes to be selected, i.e., the coalition size. In our simulations we assume \( N_a = 10 \) and \( N_b = 3 \).

Since the network density is not high, and having in mind limited communication range, \( N_a \) is relatively small and this approach is applicable. Otherwise we would have to switch to the pruned search methods. Each combination represents a possible coalition, and for all of them the coalition value is calculated based on (4). The subset of nodes with the highest value for utility function is used to estimate the position. Note that the set containing all \( N_a \) available reference nodes represents the grand coalition (the coalition of all nodes). However, since in our case there is cost associated with coalition formation, and we limit the selection to 3 nodes, the grand coalition will not form.

We take into account the virtual anchor uncertainty by means of variance \( \omega^2 = 1 \) m, and adopt the results from [18], according to which imperfect location knowledge corresponds to the increase of variance, in case of RSS-based distance estimates. Figure 3 shows through a scatter plot that higher coalition values lead to more accurate position estimates.

In Figure 4 we compare the error cumulative distribution functions (cdf) of our proposed selection strategy against selection based on closest distances, as well as purely random reference selection. We perform 1000 independent runs. The 90th percentile for utility based selection is 3.5 m, which is an improvement of 39%, compared to 90th percentile of 5.8 m for closest distance, and 51% improvement with respect to the random case.

On the other hand, the computational complexity notably increases for larger coalition sets. Here the computational complexity is defined as the amount of time spent on localization, in this case simulation run. The measurement of computation time is calculated using MATLAB functions tic and toc, which return the elapsed time in seconds. The increase in elapsed time when making coalitions sets of size 3, 4, and 5 elements is shown in Figure 5.

VI. CONCLUSION

In this paper we propose a technique for reference node selection with the objective to improve position accuracy in iterative multilateration based algorithms. Utility functions incorporate all information relevant for the node selection problem with respect to accuracy, and by incorporating a cost function they provide a powerful tool to model the tradeoff between accuracy and cost. The proposed method is viable for distributed algorithms since it only involves information from the local neighborhood. For the considered scenario exhaustive search methods are still applicable; however for a dense deployment it would require too much computation. Therefore our future work will focus on more efficient, heuristic based search methods.

ACKNOWLEDGMENT

This work has been performed in the framework of the ICT project ICT-248894 WHERE2, which is partly funded by the European Union.

REFERENCES


A.4 A Variational Message Passing Algorithm for Sensor Self-Localization in Wireless Networks

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Claus Pedersen, Troels Pedersen, Bernard H. Fleury A Variational Message Passing Algorithm for Sensor Self-Localization in Wireless Networks
©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
A Variational Message Passing Algorithm for Sensor Self-Localization in Wireless Networks

Claus Pedersen, Troels Pedersen, Bernard H. Fleury
Section Navigation and Communications, Department of Electronic Systems
Aalborg University, Aalborg, Denmark
Email: {cpe, troels, bfl}@es.aau.dk

Abstract—We propose a novel algorithm for sensor self-localization in cooperative wireless networks where observations of relative sensor distances are available. The variational message passing (VMP) algorithm is used to implement a mean field solution to the estimation of the posterior probabilities of the sensor positions in an \( \mathbb{R}^2 \) scenario. Extension to \( \mathbb{R}^3 \) is straightforward. Compared to non-parametric methods based on belief propagation, the VMP algorithm features significantly lower communication overhead between sensors. This is supported by performance simulations which show that the estimated mean localization error of the algorithm stabilizes after approximately 30 iterations.

I. INTRODUCTION

Information collected and communicated by a wireless sensor in a wireless sensor network (WSN) is often only valuable if the location of the wireless sensor is known [1], [2]. Manually supplying wireless sensors with their positions is cumbersome or impossible and equipping wireless sensors with a global positioning system (GPS) receiver may be cost and energy prohibitive [1], [3]. Furthermore, GPS signals have poor building penetration properties and receiving these signals indoors or in urban areas surrounded by tall buildings may be difficult or impossible. Consequently, the position information is inadequate or erroneous [2]. To meet the challenge of providing position information in wireless networks, reliable methods for self-localization of wireless sensors are in demand.

In cooperative localization, sensors in a network estimate their own positions by exploiting relative position information obtained from measurements with neighbour sensors and/or absolute reference locations available from anchor sensors [3]. In order to estimate its own position from the information obtained from other sensors, each sensor needs a processing unit and an algorithm for self-localization.

Self-localization algorithms based on geometric and probabilistic methods have been considered previously. In [4] and [5], sensor localization methods based on convex optimization and semidefinite programming are considered. Probabilistic localization methods based on belief propagation (BP) in factor graphs, and its sum-product (SP) implementation, are proposed in [2] and [3]. The BP methods in these contributions yield accurate results at the expense of large communication overhead due to the use of a large number (typically hundreds) of samples (particles) to represent the messages. The authors of [6] propose an expectation-propagation-based localization algorithm which uses Gaussian estimates instead of particles to represent messages. Variational Bayesian methods, and their variational message passing (VMP) implementation, complement BP methods for probabilistic inference on factor graphs [7], [8]. In the particular application context of localization considered in this paper, the VMP algorithm allows for simpler message representations than the SP algorithm. This translates into lower communication overheads between nodes.

In this contribution, we apply the VMP algorithm to distributed, iterative self-localization of sensors in cooperative wireless networks. We present a probabilistic model for the joint probability density of sensor positions and relative sensor distance observations in a WSN. We contrast the structures of the SP and VMP algorithms in this particular application context and show that compared to particle based BP methods the communication overhead of VMP can be drastically reduced by approximating the posterior densities of the sensor positions with circular symmetric Gaussian densities. The resulting scheme features a simple representation of the messages broadcast by the nodes. For instance, in an \( \mathbb{R}^2 \) localization scenario, the mobile nodes need pass only three real values (the mean, and the standard deviation of the Gaussian pdf approximating their position) at each iteration. We investigate the performance of the VMP algorithm in a static scenario containing 100 mobile and 13 anchor sensors by means of Monte Carlo simulations. Finally, we present our concluding remarks.

II. MODELS

Consider a graph defined by a set of vertices \( V \) and a set of edges \( E \) (cf. Figure 1). Each vertex \( v \in V \) represents a wireless sensor placed randomly in the plane and each edge \( (r, t) \in E \) represents a communication link between sensors \( r \) and \( t \), where sensor \( r \) receives a signal transmitted from a neighbouring sensor \( t \). The set \( V \) of sensors is divided into a set of anchor sensors \( V_A \) at known, fixed positions and a set of mobile sensors \( V_M \) at unknown positions. The position of
sensor $v$ is given by the vector $x_v \in \mathbb{R}^2$.

We describe sensor $v$'s prior knowledge of its position by a circular symmetric Gaussian pdf $p_v(x_v)$ in $\mathbb{R}^2$ with mean $\mu_v = E_{p_v}(x_v)$ and variance $\sigma_v^2 = \frac{1}{2}E_{p_v}(x_v) \| x_v - \mu_v \|^2$, where $E_{p_v}$ denotes expectation with respect to the pdf $p_v$, and $\| \|$ is the Euclidean norm. In the special case when $v \in V_A$, $\sigma_v^2 = 0$ and $p_v(x_v)$ reduces to a Dirac’s delta function localized at $\mu_v$ in $\mathbb{R}^2$.

If $(r, t) \in \mathcal{E}$, sensor $r$ can obtain sensor $t$’s current position information and a noisy measurement of the distance $d_{r,t}$ between $r$ and $t$:

$$d_{r,t} = \| x_r - x_t \| + w_{r,t},$$

where $w_{r,t}$ represents observation noise. In this work, $w_{r,t}$ is a zero-mean Gaussian random variable with variance $\sigma_{r,t}^2$.

Given a network of $N$ sensors, let $\mathcal{X} = \{ x_i : i \in V_M \}$ denote the set of unknown sensor positions. The set $\mathcal{E}$ is obtained as follows: for any $r, t \in \mathcal{V}$, $(r, t) \in \mathcal{E}$ if, and only if, $\| x_r - x_t \| \leq R$. Thus, any two sensors in $\mathcal{V}$ are connected if, and only if, their distance is not larger than a given coverage radius $R$. The set $\mathcal{D} = \{ d_{r,t} : (r, t) \in \mathcal{E} \}$ contains the distance observations between the connected sensors. In the considered decentralized scheme each mobile sensor only utilizes the distance measurements from the sensors with which it is connected. Notice that, if each sensor in addition has access to information on the network topology, e.g. to know the positions of the sensors connected to its neighbours with which it is connected, a more sophisticated scheme would also exploit the position information inherent to the knowledge of absence of connection [3], e.g. to these neighbours’ neighbours.

The joint pdf describing the probabilistic model for the considered scenario reads

$$p(\mathcal{X}, \mathcal{D}) = p(\mathcal{D} | \mathcal{X}) p(\mathcal{X})$$

$$= \left( \prod_{(r, t) \in \mathcal{E}} p(d_{r,t} \mid x_r, x_t) \right) \left( \prod_{v \in V_M} p_v(x_v) \right),$$

where $p(d_{r,t} \mid x_r, x_t)$ is the pdf of the observation $d_{r,t}$ conditioned on the positions of sensors $r$ and $t$.

III. MESSAGE PASSING FOR LOCALIZATION

A. Message Passing on Factor Graphs

The joint pdf in (3) is representable by a factor graph [9] with local factors

$$f_v(x_v) = p_v(x_v),$$

$$g_{r,t}(x_r, x_t) = p(d_{r,t} \mid x_r, x_t).$$

We abbreviate the notation as $f_v$ and $g_{r,t}$ for convenience. For each sensor $v$, we draw a variable node, representing the sensor’s position $x_v$. We connect each $x_v, v \in V_M$ to a factor node $f_v$, representing the prior position pdf. For each pair of sensors $(r, t)$, for which a distance observation is available, we draw a factor $g_{r,t}$ and connect the variable nodes $x_r$ and $x_t$ to it. In this step, we make the following two assumptions: a) For sensors $r$ and $t$, $(r, t) \in \mathcal{E} \Leftrightarrow (t, r) \in \mathcal{E}$; b) Anchor sensors have known positions. Thus, the variable node of an anchor sensor $a$ is only connected to the variable node of a mobile sensor $m$ via $g_{m,a}$, because anchors need not estimate their positions and therefore may disregard messages from neighbour sensors. The result is an undirected graph. As an example, the factor graph depicted in Figure 2 corresponds to the WSN topology of Figure 1.

The position posterior pdf $p(x_r \mid \mathcal{D})$ of any mobile sensor $r \in V_M$ can now be estimated via message passing methods [7]–[14]. Two common message passing methods adapted to graphs as depicted in Figure 2 are displayed in Figure 3: the SP algorithm, which implements BP [9], and VMP, which implements the variational Bayesian method [12].

B. The Sum-Product Algorithm

For continuous hidden variables, evaluation of (6), (8) and (9) (see Figure 3a) can become arbitrarily complex [3], [7], [12], [13]. A way to control this is to restrict the messages passed between the nodes to be Gaussian [7]. Nevertheless, Gaussian SP remains unattractive for the problem of localization adressed in this contribution because the nonlinear sensor relationship in the observation model (1) leads to unwieldy integrals in (8). This difficulty can be remedied via the use of particle based methods, e.g. nonparametric belief propagation [3], [12], [13]. In this approach, messages (6) – (8) in Figure 3a are represented by typically hundreds of real-valued samples. Transmission of such messages imposes substantial communication overhead and collecting the incoming messages require that the sensors be equipped with ample memory hardware.

C. The Variational Message Passing Algorithm

Variational methods aim at approximating a complex or intractable pdf by a simpler pdf [7], [12]. That is, using the notation in Section II, given the set $\mathcal{X}$ of unknown positions $\{ \mathcal{X}_i : i \in V_M \}$ and the set $\mathcal{D}$ of distance measurements, the posterior pdf $p(\mathcal{X} | \mathcal{D})$ is approximated by a pdf that belongs to a certain family of pdfs satisfying certain constraints that make their computation tractable. The selected pdf $q(\mathcal{X})$ is the one in the family for which the Kullback-Leibler divergence

$$\text{KL}(q(\mathcal{X}) \| p(\mathcal{X} | \mathcal{D})) = \int_{\mathcal{X}} q(\mathcal{X}) \ln \frac{q(\mathcal{X})}{p(\mathcal{X} | \mathcal{D})} d\mathcal{X}$$

Figure 2. The factor graph that represents the WSN topology in Figure 1. Anchor variable nodes are gray.
a) The sum-product algorithm.

<table>
<thead>
<tr>
<th>Messages from variable ( x_r ) to local factor ( g_{r,t}(x_r, x_t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{x_r \rightarrow g_{r,t}}(x_r) = \prod_{h \in N(x_r) \setminus {g_{r,t}}} m_{h \rightarrow x_r}(x_t) ). ( (6) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Messages from local factors to variable ( x_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{f_r \rightarrow x_r}(x_r) = \int_{x_t} m_{x_r \rightarrow g_{r,t}}(x_t) g_{r,t}(x_r, x_t) , dx_t ). ( (7) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marginal update of the pdf estimate of ( x_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_r(x_r) = \prod_{h \in N(x_r)} m_{h \rightarrow x_r}(x_r) ). ( (9) )</td>
</tr>
</tbody>
</table>

Figure 3. Two message passing algorithms for unconstrained Bayesian inference in a localization factor graph: \( N(x_t) \) denotes the set of factor nodes neighbouring the node \( x_t \) and \( Z \) is the normalization constant defined in (17).

is minimum. A well-known variant of variational methods is the mean field approximation framework from statistical physics where \( q(X) \) is assumed to factorize as \( q(X) = \prod_{i} q_i(x_i) \) \( [7, 8, 12] \). The mean field approximation yields an iterative algorithm that approximates \( p(X|D) \) by separately updating the factors \( q_i(x_i) \) in a sequential manner. Note that this factorization tends to produce overly confident marginals in the approximation of the posterior pdf \([15, \text{Section 10.1.2}] \). A message passing interpretation of this algorithm, which we refer to as variational message passing (VMP), is provided in \([12]\).

Similar to BP, the equations \((10), (12)\) and \((13)\) (see Figure 3b) of unconstrained VMP can become arbitrarily complex for continuous hidden variables. A method to harness the complexity is to restrict the messages passed by variable nodes to belong to the family of exponential pdfs \([7]\). In our work, we restrict these messages to be circular symmetric Gaussian pdfs and demonstrate that, contrary to Gaussian SP, this so-called Gaussian VMP leads to a tractable iterative scheme for distributed localization.

When the message \( m_{x_r \rightarrow g_{r,t}}(x_t) \) in \((12)\) is restricted to be a Gaussian pdf, the \( \ln g_{r,t} \) term in this equation yields the exponent of a Gaussian, and the integral can be computed analytically despite the nonlinear sensor relationships in \((1)\). In this way all messages from factor nodes to variable nodes are given in closed form. Each variable node computes its corresponding product in \((10)\), approximates this product by a circular symmetric Gaussian pdf, and passes the parameters of this pdf as the message to its neighbour factor nodes. For localization in \( \mathbb{R}^2 \), this amounts to three real values which are broadcast to all neighbouring factor nodes \((\text{see } (10))\). Compared to particle based BP methods, Gaussian VMP messages impose significantly smaller requirements on message communication overhead and sensor memory hardware.

IV. THE GAUSSIAN VMP LOCALIZATION ALGORITHM

The unconstrained VMP algorithm listed in Figure 3b imposes no restrictions on the messages passed between the nodes in the factor graph. To develop a tractable mean field localization algorithm, we restrict the messages from variable nodes to factor nodes to be in the family \( G \) of circular symmetric Gaussians with mean \( \hat{x}_i \) and variance \( \sigma_i^2 \) for the \( i^{th} \) node. As a result of this constraint, equations \((10)\) and \((13)\) in Figure 3b must be modified according to (superscript \( G \) indicates Gaussian restriction)

\[
\text{arg min}_{q_r^G(x_r) \in G} KL(q_r^G(x_r) \parallel \tilde{p}_r(x_r)) \quad (15)
\]

with

\[
\tilde{p}_r(x_r) = \frac{1}{Z} \prod_{h \in N(x_r)} m_{h \rightarrow x_r}(x_r), \quad (16)
\]

where \( Z = \int_{x_r} \prod_{h \in N(x_r)} m_{h \rightarrow x_r}(x_r) \, dx_r \) \( (17) \)

and

\[
q_r^G(x_r) = m_{x_r \rightarrow N(x_r)}^G(x_r) \quad (18)
\]

respectively. The solution to \((15)\) is obtained by finding the position and variance estimates \( \hat{x}_i \) and \( \sigma_i^2 \) of \( q^*_r(x_r) \in G \) minimizing \( KL(q^*_r(x_r) \parallel \tilde{p}_r(x_r)) \). These estimates can be obtained by solving the minimization problem using numerical methods.

To compute \( \tilde{p}_r(x_r) \), let \( \mathcal{V}_r = \{ t \in \mathcal{V} : (t,r) \in E \} \) and recast \((16)\) as

\[
\tilde{p}_r(x_r) = \frac{1}{Z} \prod_{t \in \mathcal{V}_r} m_{f_r \rightarrow x_r}(x_r) \prod_{t \in \mathcal{V}_r} m_{g_{r,t} \rightarrow x_r}(x_t). \quad (19)
\]

From \((11)\)

\[
m_{f_r \rightarrow x_r}(x_r) \propto \exp \left(-\frac{\| x_r - \hat{x}_r \|^2}{2\sigma_r^2} \right), \quad (20)
\]

where \( \propto \) denotes proportionality. For \( t \in \mathcal{V}_r \cap \mathcal{V}_h \),

\[
m_{g_{r,t} \rightarrow x_r}(x_r) \propto \exp \left(-\frac{1}{2\sigma_{r,t}^2} \| d_{r,t} - \| x_r - \hat{x}_t \| \|^2 \right). \quad (21)
\]

For \( t \in \mathcal{V}_r \cap \mathcal{V}_h \), we first substitute \((5)\) into \((12)\) to get

\[
m_{g_{r,t} \rightarrow x_r}(x_r) = \exp \left( \int_{x_t} q_r^G(x_r) \ln p(d_{r,t} | x_r, x_t) \, dx_t \right). \quad (22)
\]

b) The variational message passing algorithm.

<table>
<thead>
<tr>
<th>Messages from ( x_t ) to ( g_{r,t}(x_r, x_t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{x_t \rightarrow N(x_t)}(x_t) = \frac{1}{Z} \prod_{h \in N(x_t)} m_{h \rightarrow x_t}(x_t), \quad (10) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Messages from local factors to variable ( x_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{f_r \rightarrow x_r}(x_r) = p_r(x_r) \quad (11) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marginal update of the pdf estimate of ( x_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_r(x_r) = m_{x_r \rightarrow N(x_r)}(x_r) \quad (13) )</td>
</tr>
</tbody>
</table>
Initialization
for all sensors (in parallel) do
I1) Broadcast position information, and collect position information broadcast from neighbouring sensors.
I2) Obtain distance observations \(d_{r,t}\) to neighbouring nodes.
end for

Location estimation
repeat
for all sensors (in parallel) do
L1) Compute \(\hat{p}_r(x_r)\) using (19).
L2) Compute \(q^2_t(x_r) \in \mathcal{G}\) that minimizes \(\text{Kl}(q^2_t(x_r) \parallel \hat{p}_r(x_r))\).
L3) Broadcast \(\hat{x}_r\) and \(\sigma_t^2\) and collect corresponding broadcast from neighbouring sensors.
end for
until stopping criterion is reached.

Inserting \(q^2_t(x_r)\) in (22) yields
\[
m_{x_r \rightarrow x_r}(x_r) \propto \exp\left(-\frac{1}{2\sigma_t^2}\left[2\sigma_t^2 - 2d_{r,t}\sigma_t \sqrt{\frac{\pi}{2}} \times\right.\right.
\]
\[
\left.\left.\Phi(1; 1, -\frac{||x_r-x_r||^2}{2\sigma_t^2}) + ||x_r-\hat{x}_t||^2 + 2\sigma_t^2\right]\right),
\]
where \(\Phi(a; b; x)\) is the confluent hypergeometric function of the first kind. In this particular case we can write
\[
\Phi(1; 1, 1, \frac{x}{2}) = \exp\left(\frac{x}{2}\right) \left[(1-x)\mathcal{I}_0\left(-\frac{x}{2}\right) - x\mathcal{I}_1\left(-\frac{x}{2}\right)\right],
\]
where \(\mathcal{I}_n()\) is the modified Bessel function of order \(n\) [16]. We note that when \(p(d_{r,t} | x_r, x_r)\) is Gaussian, (22) is proportional to the expectation of the exponent of \(p(d_{r,t} | x_r, x_r)\) with respect to \(q^2_t(x_r)\). Equations (15) and (19) define the Gaussian VMP algorithm for sensor self-localization, which we list in Figure 4.

V. SIMULATIONS

We verify the performance of the Gaussian VMP algorithm in a scenario similar to the one described in [2] by means of Monte Carlo simulations. In this scenario, static anchors are positioned in a structured manner as depicted in Figure 5.

Moreover, a sensor has a communication link to all other sensors within a range of 20 m, i.e. \((r, t) \in \mathcal{E}\) and \((t, r) \in \mathcal{E}\) if, and only if, \(||x_r-x_t|| \leq 20\) m. For simplicity, we assume that sensors \(r\) and \(t\) make the same distance observation \(d_{r,t} = d_{t,r}\) and that the observation noise variance is constant and equal for all sensors, i.e. \(\sigma_r^2 = \sigma_t^2\). In each simulation run, 100 static mobiles are uniformly and independently scattered in the area and each mobile sensor estimates its position with the Gaussian VMP localization algorithm depicted in Figure 4.

Figure 6, depicts the estimated mean localization error of the Gaussian VMP algorithm vs. iteration index with \(\sigma_{w}\) as a parameter. We see that depending on \(\sigma_{w}\), the estimated mean localization error does not change significantly after approximately 10–30 iterations. At this point, each sensor has in total broadcast \(3 \cdot 30 = 90\) real values in its messages to the neighbouring sensors. Compared to particle based BP methods (e.g. as proposed in [2] and [3]), the Gaussian VMP algorithm for localization has dramatically lower communication requirements. Furthermore, the plot shows that for \(\sigma_{w} \leq 2.0\) the mean localization error stabilizes at a value that is higher than the standard deviation of the noise. This is caused by sensors for which the position cannot be unambiguously determined, due to the fact that their neighbour sensors are too few and/or the topology of these neighbour sensors does not enable an unambiguous determination of the position. E.g. in the ambiguous case when \(\hat{p}_r(x_r)\) is multimodal with equal-mass modes the VMP algorithm will produce a \(q^2_t(x_r)\) which approximates one of these modes selected at random [7]. When \(\sigma_{w} > 2.0\), the algorithm exploits the network topology to mitigate the noise impact on the distance observations and the mean localization error reaches a value less than the...
In Figure 7, we plot the estimated cumulative probability distribution of the localization error. We see that for all noise standard deviations, at least 65% of the sensors localize with an error less than or equal to the noise standard deviation on average. For an allowable error of 5 m, more than 68% of the sensors are well localized for all the plotted curves. This percentage increases as $\sigma_w$ decreases. At $\sigma_w = 0.1$, nearly 97% of the sensors are localized within 5 m. The fact that the curves in Fig. 7 stabilize to values lower than one results from large errors due to an ambiguity in the estimation of the position of sensors having too few distance observations from their neighbours.

VI. CONCLUSION

We have proposed a novel low-complexity algorithm for sensor self-localization in cooperative wireless networks. The algorithm is a special implementation of the variational message passing (VMP) method, in which messages from variable nodes to factor nodes are approximated by circular symmetric Gaussian probability densities. Note that in the VMP method these messages coincide with the estimated marginal posterior densities of the node positions. The main virtue of the proposed Gaussian VMP algorithm is a low communication overhead when compared to the corresponding requirements for particle based BP localization schemes. The performance of the algorithm is illustrated in a scenario with static sensors by means of Monte Carlo simulations. When the density of sensors is low, some of them cannot be localized unambiguously as they have too few observations from other sensors. This network topology leads to a non-identifiable estimation problem. In a cooperative setting this problem can be alleviated by exchanging additional information on the network topology, e.g. each sensor gets the positions of sensors connected to its neighbours with which it has connection. The cost of this improvement is a larger communication overhead.

An extension of the Gaussian VMP algorithm to scenarios with moving sensors is currently under investigation. Also, work on comparing the proposed algorithm to particle-based methods is in progress. Further theoretical studies shall be conducted to assess the performance of the algorithm versus network characteristics like link attenuation, standard deviation of the distance measurement, and node density.

VII. ACKNOWLEDGMENT

This work was performed in the frameworks of the ICT projects ICT-217033 WHERE and ICT-248894 WHERE2, which are partly funded by the European Union.

REFERENCES

A.5 Performance of NBP-based methods in the presence of outliers

Hadi Noureddine (MERCE), Vladimir Savic (UPM) Performance of NBP-based methods in the presence of outliers
Performance of NBP-based methods in the presence of outliers

November 23, 2011
Hadi Noureddine (MERCE), Vladimir Savic (UPM)

1 Introduction

NBP-based localization algorithms perform inference in probabilistic graphical models, where every node computes an approximation of its posterior marginal distribution from its local a priori information and messages received from neighboring nodes. Probabilistic models for measurements need to be known in order to define potential functions between every two connected nodes in the probabilistic graph. The particle-based implementation of these algorithms makes them capable to handle any measurements model.

Here, we assess the performance of four such algorithms under the occurrence of outlier processes in the ranging measurements: TP-NBP [1], URW-NBP [2], NBBP [3] and NBP [4]. An outlier is an observation that is highly deviated from the true distance value. We also perform simulations for the classical centralized least-squares (LS) estimator initialized using SDP method [5].

In the following sections, the deployment scenario and the ranging measurements model are described, then simulations results are provided, and finally, concluding remarks are drawn.

2 Deployment scenario and measurements model

We consider networks of size 25 nodes consisting of 4 anchor nodes and 21 target nodes deployed in a $100 \times 100m^2$ area. The deployment area is a grid of $5 \times 5$ squares, where each square has a size of $20 \times 20m^2$. Anchor nodes are placed as shown in Figure 1 and target nodes are uniformly drawn inside the remaining squares, one node per square.

The probability that two nodes $i$ and $j$ are connected and perform a ranging measurement between each other is assumed to be a function of their separating distance $d_{i,j}$:

$$p_o(d_{i,j}) = \exp(-d_{i,j}^2/R^2).$$

(1)

We take the communication range $R$ varying from 30 to 100m, and we only consider connected networks (i.e., there is a path between each pair of the nodes).

The measured distance between two neighboring nodes $i$ and $j$ is affected by an additive error:

$$\hat{d}_{i,j} = d_{i,j} + e_{i,j}.$$  

(2)

We model the presence of the outlier by a Gaussian mixture. Thus the error $e_{i,j}$ is drawn from the following Gaussian mixture:

$$e_{i,j} \sim w_1 \mathcal{N}(0, \sigma_1^2) + w_2 \mathcal{N}(e_0, \sigma_2^2),$$

(3)

where we set $w_1 = 0.75$, $w_2 = 0.25$, $\sigma_1 = \sigma_2 = 0.5m$ and $e_0 = 10m$.

We further assume that nodes are mutually neighbors (if $i$ is a neighbor of $j$, then $j$ is a neighbor of $i$) and neighboring nodes share the same measurement (if there are two different measurements, they can be combined in a single one).

Finally, for NBP-based methods, we use 200 particles, and 5 iterations.
3 Simulation results

The performance study is done via Monte-Carlo simulations (we used 50 trials), and the performance indicator used is the accuracy of the estimated positions. In particular, according to D1.1b, we consider root-mean-square error (RMSE), median error, and 90th percentile. Results are shown in Figures 2-4.

During the second phase of TP-NBP, the max-product rule is applied and messages are exchanged between direct neighbors only (max-prod. 1-step).

The RMSE of TP-NBP using the max-product and sum-product rules during the second phase are shown in Figure 5.
Figure 3: Median error w.r.t. communication range $R$

Figure 4: 90th percentile of the error w.r.t. communication range $R$

4 Conclusions

As we can see in the figures, the errors are high at low communication ranges as the probabilities of network rigidity and unique solvability are small.

At high communication ranges, TP-NBP and URW-NBP provides the best estimate in terms of all three metrics. URW-NBP performs the best for high communication range due to the huge number of loops in the graph. On the other hand, NBP performs the worst because small number of particles (200) has been drawn across the whole deployment area. By drawing particles within bounded boxes and filtering out erroneous particles in each iteration (as in NBBP and URW-NBP), the accuracy is significantly increased. TP-NBP has a similar RMSE as URW-NBP at high communication ranges. The second phase of TP-NBP allow to reduce the errors due to the limitation of the number of particles. We can also notice in Figure 5 that max-product rule performs better than sum-product rule.

Finally, as expected, NBP-based methods (except NBP) provides better performance than LS method due to the non-Gaussian measurement noise.
Figure 5: RMSE error w.r.t. communication range $R$

References


A.6 A Robust Geometric Positioning Algorithm for Heterogeneous Wireless Networks

©2012 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

A Robust Geometric Positioning Algorithm for Heterogeneous Wireless Networks N. Amiot, M. Laaraiedh, B. Uguen, and S. Avrillon
A Robust Geometric Positioning Algorithm for Heterogeneous Wireless Networks

N. Amiot, M. Laaraiedh, B. Uguen, and S. Avrillon
IETR, University of Rennes 1, France
{nicolas.amiot, mohamed.laaraiedh, bernard.uguen, stephane.avrillon}@univ-rennes1.fr

Abstract—An original and accurate robust geometrical positioning algorithm (RGPA) is proposed for mobile and wireless network. According to the Monte Carlo simulation RGPA outperforms maximum likelihood using a Nelder-Mead simplex optimizer (ML-NMS) for hybrid and non hybrid localization schemes.

As Classical algebraic maximum likelihood (ML) localization techniques, RGPA is based on location dependent parameters (LDP). However, position estimation in RGPA geometrically processes LDPs. RGPA lied on simple mathematical operations and is computationally efficient and easily embeddable.

Index Terms—Localization, Wireless networks, Heterogeneous networks, TOA, TDOA, RSSI, Interval analysis, Maximum Likelihood.

I. INTRODUCTION

Whereas a single standard unifying wireless networks enabling cooperation scheme is complex to practically implement [1], the use of multiple radio access technologies (RAT) of devices in heterogeneous wireless networks seems to become a spare solution. The architecture of these heterogeneous networks tends to exploit all the available RATs and thus makes possible either to perform a position estimation and thus limit communication between nodes, or to wisely adjust the transmission power. In wireless network, localization can be seen as an unifying concept. Indeed, from wide area network as GNSS [2] or GSM [3], to wireless personal area network Zigbee [4] and through wireless local area network [5], localization is an active domain of research.

Localization algorithms in wireless network are based on location dependent parameters (LDP) observables. Among others, these LDPs can be either time-based such as time of arrival (TOA) or difference time of arrival (TDOA), or power-based such as received signal strength indicator (RSSI) [6]. As in throughput optimization of a network [7], hybrid positioning is convenient, because it aims to use every available data from each available device’s RAT to perform a position estimation. These LDPs observables, are then processed by classical algebraic algorithms as least square or maximum likelihood (ML) [8] thanks to ad-hoc optimizers.

In robotics, where localization is also an important topic, a family of geometrical localization algorithms is based interval analysis [9]. For a few years, both the increasing number of wireless sensors in robots and the accelerometers in mobile phones has reduced the gap between these fields. As a proof, the interval analysis method has even been recently used in GNSS localization [10]. Mindful of these points, this paper proposes to evaluate performances of a geometrical approach for localization. The developed robust geometrical positioning algorithm (RGPA) is a suitable approach to perform localization from LDPs which has not already widely investigated in the context of wireless network. It allows to easily merge heterogeneous LDPs or any extra layout information. Moreover, by solving geometrically the estimation problem, RGPA outperforms classical linearized approaches.

Section II presents the assumed scenario both for algorithm explanations and results computation. Section III presents RGPA. Monte Carlo simulation setup is presented in Section IV-A and results are shown and discussed in Section IV-B. Finally, concluding remarks and further improvements are discussed in Section V.

II. ASSUMED SCENARIO

Hybrid scenarios, mixing different wireless technologies, leads to significant accuracy improvement in localization [11]. Moreover, these schemes are also close to a real scenario with devices embedding several RATs able to reach various networks. For that purpose, Fig. 1 depicts the assumed hybrid scenario both for algorithm explanation and performance evaluation.

![Fig. 1: The assumed hybrid scenario : A wireless device (UE) needs to be located. It obtains 3 TDOA observables from base stations (BS), or/and 4 TOA observables from mobile stations (MS) or/and 4 RSSI observables from access points (AP).](image)
The user equipment (UE) is a blind node assumed seeking for its position. It lies in a $L \times L$ squared region.

Anchor nodes are regularly distributed along this squared region. UE is able to connect to these anchor nodes. Each anchor node provide a unique but different location dependent parameter (LDP) which could be:
- TDOA LDPs represented by Circle shape are “base stations” (BS) anchors,
- TOA LDPs represented by Square shape are “mobile station” (MS) anchors,
- RSSI LDPs represented by Round shape are “access points” (AP) anchors.

Note that the type of anchors is just given as a compliant example with the WHERE2 scenario [12], but RGPA operates in the same way with different anchor configuration.

III. RGPA DESCRIPTION

RGPA solves geometrically the localization problem, by finding the region where all constraints are satisfied. A constraint is defined as a rule which bounds a space region. The nature of the constraint could be either a RF measurement or a building layout. RF constraints are obtained from classical observables as time of arrival (TOA), difference of time of arrival (TDOA), or received signal strength indicator (RSSI). A TOA measurement is represented by an annulus in 2 dimensions (2D) and a thick shell spherical in 3 dimensions (3D); TDOA by a thick shell hyperboloid in 2D and a thick shell hyperboloid of revolution in 3D; RSSI, as TOA, is represented by an annulus or a sphere, designed thanks to a pathloss model to ensure the power to distance association.

In the following, for the ease of representation, only the 2D case is studied.

Table I summarizes the 5 steps operation of RGPA. The detailed operation of the RGPA algorithm is presented step by step in the following.

### TABLE I: Robust Geometric Positioning Algorithm (RGPA)

1. Describe a set of geometrical constraints (Fig. 2),
2. Bound the constraint with its enclosing box (Fig. 3),
3. Merge all enclosing boxes together to obtain a restricted box (Fig. 4),
4. Approximate the restricted box with a quadtree/octree algorithm to obtain an approximated region (Fig. 5),
5. Estimate the position from the approximated region (Fig. 6).

Fig. 2 shows the annulus of both a TOA observables and a RSSI observables and the inter hyperbola region of TDOA observables. At this step, these constraints are initialized with an initial guess based on the observables and will be updated during the merging step. The constraint interval is bounded both by a lower value $c_m - 3\lambda \sigma_m$ and by a higher value $c_m + 3\lambda \sigma_m$ where $c_m$ is the observable value, $\sigma_m$ its standard deviation and $\lambda$ a scale factor. At this step this scale factor is set to 1 and can be modified during the merging step.

Fig. 3 illustrates the bounding step. It consists in properly limiting a space region. For ease of computation, constraints are here assimilated to their bounding box. In interval analysis theory, a box is a set of $n$ real interval [13]. The TOA and RSSI boxes are squares of $c_m + 6\lambda \sigma_m$ side centered is set on their origin. TDOA boxes are cropped using a propriety of TDOA observables $c_m$. Their time observables are intrinsically bounded by $[c_m] \leq (2d_{TDOA})/c$ where $d_{TDOA}$ is the distance between the both anchors involved in the TDOA observable and $c$ is the speed of light.

Fig. 4 illustrates the merging step. It consists in finding the smallest constraints’ boxes intersection. It is only based on simple belonging tests:
- If the intersection exists for all constraints boxes, the scale factor $\lambda$ of each constraint is incrementally decreased in order to find the smallest intersection box.
- If no intersection exists, the scale factor $\lambda$ of each constraint is incrementally increased until an intersection box appears.
- If the intersection exists for few constraints boxes, the scale factor $\lambda$ is incrementally increased only for constraints which don’t intersect.

The result of this merging step is a restricted box which encloses the true UE position.
A cluster of boxes is obtained once all boxes have been processed. This cluster of boxes is composed by *enclosed* and *ambiguous* boxes; *out* boxes are removed.

The next quadtree iteration, is solely applied on *ambiguous* boxes. After several iteration the approximated region is obtained. The number of iteration is fixed by a stopping criterion which is either the box size or the box number. The Robust Set Inverter via Interval Analysis (RSIVIA) algorithm described in [10] proposed a slightly different method to bound the region: The initial box is shrunk until it is not compliant with all the constraints, and thus it is bisected into 2 boxes. In comparison, quadtree has several advantages. First, it allows to easily switch from 2D to 3D. Then the recursive structure of quadtree algorithm can be easily integrated into a parallel computing architecture.

Fig. 6 illustrates the estimation step. It consists in estimating the true position. This position is set as the center of mass of the previously approximated region. Fig. 6b is a screenshot of an estimated UE position using our demonstrator.

IV. RESULTS AND DISCUSSIONS

A. Simulation setup

Position estimation error is displayed using cumulative distribution functions (CDF). RGPA technique is compared both to the maximized likelihood using a Nelder-Mead simplex optimizer (ML-NMS) [17], and to the Cramer Rao lower bound (CRLB) [18]. Multidimensional likelihood functions corresponding to the given scenarios are described in [19]. The optimizer is initialized with the result of a least square estimator. UE position has been estimated in both hybrid and non hybrid scenarios. All permutations using RSSI, TOA, or TDOA have been considered.

Simulations are performed in the 2D environment shown in Fig. 1. All anchors nodes are distributed on a L-by-L square. For all simulations L is taken equal to 20m. Anchors nodes not involved in the LDP test are removed. CDFs have been obtained using 1000 random positions of the UE.

In non hybrid scenario (RSSI, TOA, TDOA), 4 nodes are involved in the position estimation. In hybrid scenario (RSSI + TOA, RSSI + TDOA, TOA + TDOA), 8 nodes are involved in the position estimation. In the “full hybrid”
scenario (RSSI + TOA + TDOA), 12 nodes are involved in the position estimation.

LDPs observables are perturbed with an additive centered Gaussian noise \( \mathcal{N}(0, \sigma^2) \). The standard deviation \( \sigma \) of these perturbations have been set to: \( \sigma_{TOA} = 1.142 \) m for TOA observables, \( \sigma_{TDOA} = 1.85 \) m for TDOA observables, and \( \sigma_{sh} = 3.98 \) dB for RSSI observables. This \( \sigma_{sh} \) represent the shadowing variance of the path loss model which also assumes pathloss at 1 meter, \( P'L_0 = -36 \) dB, and a path loss exponent \( n_p = 2.38 \). The scenario assumes that all LDPs are independent.

B. Simulation results

Although probability theory states that a ML well initialized is supposed to be the most efficient estimator, the fact that RGPA outperforms ML-NMS is consistent with the theory. Here ML-NMS is initialized with a least square solution. However, due to the non convexity of the problem, likelihood function is complex and may present several local minima. This may trap the Nelder-Mead simplex optimizer. RGPA avoid this problem by using a geometrical approach. This explains its better accuracy.

The RGPA outperforms the ML-NMS estimator for all configurations except a similar accuracy for non hybrid TOA. The highest gains are observed for TDOA schemes.

Fig. 7 a-c shows the CDFs for non hybrid localization techniques. For TDOA, Fig 7a, RGPA outperforms ML-NMS. Indeed, TDOA problems are generally not well conditioned problems and the linearization imposed by algebraic techniques could produce very large errors. Thanks to RGPA, all UE are positioned within a precision of 5 m. For TOA, Fig. 7b, reveals that both methods present similar results. TOA problems generally leads to a well conditioned problem [20] for a correct topology of sensors in order to limit GDOP effects [21]. For RSSI, Fig. 7c states that RGPA outperforms ML-NMS by limiting large errors thanks to the bounding constraint method.

Fig. 8 a-d shows the CDFs for hybrid localization techniques. In hybrid schemes, best performances involves TDOA observables as observed in Fig. 8a, and Fig. 8c. Compared to the TOA + TDOA scheme, the full hybrid scheme (Fig 8d) don’t really take advantage of the RSSI observables. Due to the geometrical approach, if the RSSI observables are accurate enough to provide a proper positioning which could help a time based observable, they can slightly decrease the global accuracy when the time based accuracy is already high. The relevant use of RSSI observables in these schemes should be a thorough study.

Fig. 9 shows CDFs of all localization techniques using RGPA. In non hybrid schemes, time based LDPs (TOA or TDOA) give better accuracy than the power based LDPs (RSSI). As in algebraic algorithms, the relation between distance and the RSSI observable relies on path loss models which give an imperfect statistical representation of the radio channel. Moreover, the variation of shadowing is generally high and makes the distance estimation very inaccurate. The RSSI information alone is generally not accurate enough to ensure precise positioning.

In hybrid schemes, the addition of a time based LDP (TOA, TDOA) to a power based (RSSI) drastically improves accuracy of positioning. As expected, the time based observables gives the best accuracy.
V. CONCLUSION

This paper has proposed an algorithm to geometrically solve localization problem: RGPA. Based on interval analysis, it uses a set of constraints which are rules restricting a region of space. This paper has shown that RGPA outperforms ML-NMS without prior information both in hybrid and non-hybrid schemes. The best accuracy improvements compared to ML-NMS is observed for TDOA schemes. The five steps of RGPA lie on simple mathematical operations on set membership. The reachable complexity may be adjusted by the quadtree approximation. This makes RGPA easily embeddable and computationally efficient. Another advantage of RGPA is that the use of geometrical constraints allows to unify readily any type of observables (fingerprinting information, building layout information,...).

A further work will focus on cooperative schemes of RGPA and the ease to share geometrical constraints between RF devices. The proper use of including RSSI observables will also be studied in order to improve performances of hybrid schemes. It also might be interesting to consider hybridization between geometric and algebraic techniques by computing RGPA as an accurate initial guess of the ML optimizer. Finally, a dynamic version of the RGPA algorithm is under development.

ACKNOWLEDGMENT

The work presented in this paper has been performed in the framework of the FP7 project ICT-248894 WHERE2 (Wireless Hybrid Enhanced Mobile Radio Estimators - Phase 2) which is funded by the European Union.

REFERENCES


A.7 Simulation-based Evaluation of Cooperative Tracking Filters in Realistic Multi-RAT Heterogeneous Contexts

Simulation-based Evaluation of Cooperative Tracking Filters in Realistic Multi-RAT Heterogeneous Contexts
Elyes Ben Hamida, Benot Denis
Simulation-based Evaluation of Cooperative Tracking Filters in Realistic Multi-RAT Heterogeneous Contexts

Elyes Ben Hamida, Benoît Denis
CEA-Leti Minatec Campus - 17 rue des Martyrs, 38054 Grenoble Cedex 09, France

I. INTRODUCTION

THREE main technical challenges have been identified with future mobile, location-enabled, heterogeneous and cooperative wireless networks (See Fig. 1), namely the exploitation of dynamic (unprecedented) multi-standard connectivity conditions; the contextual selection of the most relevant links or measurements accordingly (e.g. preventing from uninformative exchanges/data while benefiting from maximum diversity and redundancy) and finally, the design of efficient cooperative (and possibly decentralized) algorithms incorporating the selected peer-to-peer measurements/messages to track mobile terminals over their long-term trajectories. Preferably, the access to costly centralized servers or fixed elements of infrastructure shall be also limited, hence sharing computational complexity and homogenizing location-specific traffic over the multiple mobile users disseminated in the environment.

In this context, we account herein for preliminary investigations that aim at addressing jointly the previous key points. First of all, one significant part of the efforts was devoted to the development of a specific simulation architecture. This tool accounts for multi-standard protocol stacks at mobile terminals, coupling a packet-oriented event-driven simulation core that relies on the WSNet formalism [1], which includes RAT-specific abstractions for both the PHYsical (PHY) and Medium Access Control (MAC) layers, with the Mobisim2 [2] traces generator, which comprises various realistic mobility models including the Levy-Flight one [3]. For the sake of evaluating the performance of cooperative tracking filters with advanced links mitigation (and/or in turn, links selection) strategies, the overall architecture can generate and replay exploitable outputs, such as history profiles of Location Dependent (radio) Parameters (LDPs), e.g. Signal to Interference and Noise Ratio (SINR), Received Signal Strength (RSS), Time Of Arrival (TOA), under realistic dynamic connectivity and network deployment conditions (i.e. depending on instantaneous sets of mobile neighbors). As a very first step of these investigations in the heterogeneous radio context, we simply considered here combining a narrow-band radio at 2.4 GHz with Impulse Radio - Ultra Wideband (IR-UWB) at the multi-standard Mobile Terminals (MT).

Preliminary simulations regarding the number of available links versus the precision of the corresponding LDPs illustrate practical operating trade-offs for such heterogeneous networks. For instance, when assuming long timeout periods to maintain neighbors tables in each mobile terminal, following a beaconing strategy to update those tables, the number of available links can be very high at the adverse price of unreliable distance or position estimates. Then in the previous evaluation frame, the decentralized and asynchronous tracking filter solution initially presented in [4] has been considered, which takes into account the reliability of neighboring mobile terminals (seen as virtual anchors). This tracking solution has been compared with more classical solutions, including alternative cooperative positioning and tracking solutions, in an indoor professional scenario. The latter is compliant with the maximum speed and terminals/anchors deployment densities specified in [5]. Hence, the presence of other critical trade-offs for timeout periods depending on the current terminal velocity has been pointed out, disclosing possible compromises between cooperation diversity and link quality, while anticipating on the context-dependent and adaptive update strategies currently under investigation.

The report is structured as follows. In Section II, we present the developed specific simulation architecture. In Section III, we describe positioning and tracking algorithms that are further evaluated and compared under realistic network deployment and mobility conditions in Section IV. Finally, V concludes the paper, introducing current works on links selection in the very cooperative, heterogeneous and mobile context.

II. DYNAMIC HETEROGENEOUS WIRELESS NETWORKS SIMULATOR

One specific simulator was developed, accounting for realistic terminals mobility and deployment conditions, as well as realistic physical and medium access abstractions, in the heterogeneous context depicted on Fig. 1. On this example, BS1 – 3 can represent some outdoor base stations (e.g. 3GPP-LTE, etc.), whereas AP1 – 2 stand for indoor access points (e.g. femto-cells, WiFi Access Points) and MT1 – 3 are multi-standard mobile terminals to be positioned. In the following, without loss of generality, we assume that the fixed elements...
Fig. 1. Example of indoor heterogeneous and cooperative scenario. (e.g. AP1 − AP2) are geo-referenced and can be used as anchors in the positioning and tracking problem.

A. Mobility Modeling

One first main feature of the simulator consists in generating realistic mobility patterns for multiple cooperative terminals moving according to the maximum pedestrian speeds specified in [5]. This is handled by a Mobimsim2 [2] block, which supports so-called behavioral models accounting for realistic human activity such as the Levy Flight or Walk [3] or much simpler linear piecewise models such as the Billiard one (See e.g. Fig. 2 and Fig. 3 respectively). Other available models are the Manhattan, Reference Point Group Mobility, Markov, Persue models.

Fig. 2. Example of Levy Walk terminal trajectory supported in the dynamic heterogeneous wireless networks simulator.

Fig. 3. Example of Billiard terminal trajectory supported in the dynamic heterogeneous wireless networks simulator.

B. Multi-RAT Protocols

The original WSNet [1] formalism, as represented in Fig. 4, relies on an event-driven approach to represent large-scale mobile wireless networks. It enables a quite exhaustive simulation of wireless nodes, including intrinsic capabilities and features (e.g. battery lifetime, data traffic patterns), an application layer, routing functions, a MAC layer, and realistic PHY modeling, including abstracted Packet/Bit Error Rates (P/BER) depending on the considered modulation schemes, antenna gains, pathloss, fading, shadowing, interference levels, etc. This initial version has been extended herein to enable the simulation of multi-standard MTs within yet non-covered heterogeneous and dynamic wireless contexts. A new WSNet mobility module compliant with the Mobisim2 generator presented in the previous section has also been developed in parallel. This new frame provides support to multiple lower MAC/Radio interfaces. In terms of MAC layers, IEEE 802.11 DCF (WiFi), IEEE 802.15.4 (Zigbee) are already available. As for radio layers, WiFi at 2.4GHz, Zigbee at 868MHz and 2.4GHz and simplified IEEE 802.15.4a (IR-UWB) are represented. Dynamic multi-standard coexistence is hence naturally supported through so-called multi-channel communications. In the following, we consider simple beaconing procedures (i.e. the periodic emission of ranging beacons/request) for neighbors discovery, neighbors table management, link quality estimation, etc. as a support to ranging and tracking. More precisely, depending on the received/lost beacons, each MT maintains a local table containing the available neighbours/links and their related information (e.g. link quality, estimated distance, neighbor id and position, etc.). Dynamic connectivity tables can then be exploited further by higher-level positioning and tracking algorithms. One link is said to be active if the ranging request is answered within a pre-convinced period of time called timeout.

Fig. 4. Nodes in the initial WSNet [1] representation.
C. Radio Channel and Link Modeling

Realistic environmental and physical layer features are also jointly taken into account in our new simulator, following a gradually semi-deterministic approach, with the possibility to include walls, as well as location-dependent spatial correlations (e.g. shadowing) and/or (Non) Line Of Sight ((N)LOS) channel conditions (See e.g. Fig. 7). Accordingly, accounting for the presence of physical walls, parametric propagation models and parameters are adapted over each link (i.e. depending on the \{LOS, NLOS, NLOS^2\} condition experienced on this link) to the channel index/RAT (in the multi-channel communications framework shown in Fig. 6).

Regarding radio propagation, average path-loss models (i.e. Free Space, Log-distance, Two-ray Ground), log-normal centered Gaussian shadowing, and finally small-scale fading (i.e. Rice, Rayleigh) are also available in the simulator. Then, conditioned on the true relative locations (i.e. MTs and anchors) and on the data traffic assumed a priori, it is possible at each time to determine realistic SINR conditions, and hence, realistic Packet Reception Rate (PRR). More precisely, for each of the involved RAT, an abstraction of the form $PRR = f_{RAT}(SINR)$, which also depends on the packet length, can be drawn. In the following, this PRR is compared to an application-dependent threshold to evaluate the instantaneous connectivity of each MT while moving (i.e. with respect to both neighboring MTs and fixed anchors).

Fig. 8 shows an example of instantaneous Packet Reception Rate (PRR) 2D coverage for one particular MT under different physical model assumptions for 500 nodes in a 300mx300m indoor environment, with a MT transmitter located in (125m, 125m), a CBR data traffic, standardized IEEE 802.15.4 PHY/MAC layers and a transmit power of 0dBm.

D. Sensor Error Models

Finally, in case of successful ranging transactions (out of the packets exchanged either between two MTs of between one MT and an anchor), conditioned on the link PRR computed according to the previous Section, further random sensor errors $n$ are added to synthetic range measurements $\tilde{d} = d + n$, which
are viewed in first approximation as centered Gaussian terms and whose standard deviation $\sigma$ depends on the considered RAT (e.g. on the order or $1/n$ over IR-UWB links). The simulator also enables additional uniformly distributed additive biases $b$, which can be included into the measurements in NLOS situations, as $d = d + b + n$ (though not included for the simulation results presented in the following).

E. Overall Simulator Architecture

The overall Dynamic Heterogeneous Wireless Networks (DHWN) simulator architecture is shown on Fig. 9. This flexible framework can take into account the scenarios specified in [5] as inputs for the evaluation of tracking filters and/or links selection strategies under realistic multi-agent terminals mobility.

III. TESTED COOPERATIVE POSITIONING AND TRACKING FILTERS

Based on the realistic range measurements and connectivity profiles generated before, different positioning and tracking algorithms have been applied, including cooperative and decentralized embodiments.

In our context, one critical issue indeed consists in properly coping with heterogeneity in opportunistically cooperative - and hence sparse- networks [6], [7]. As an example, the likelihood of opportunistic interactions, as well as the prior location and ranging precisions have been considered within Weighted Centroid (WC) and the Linear Matrix Inequality (LMI) positioning algorithms. As regards to cooperative tracking more specifically, a few solutions have been proposed in the recent literature, such as generic decentralized filter formalism from the field of cooperative multi-robots localization [8], centralized cooperative Extended Kalman Filter (EKF) for terminal tracking in heterogeneous wireless networks [9], decentralized asynchronous EKFs with virtual anchors in mobile wireless sensor networks (WSN) [4]. But none of the solutions above really copes simultaneously with realistic heterogeneous conditions (e.g. with large discrepancies between the precision levels of LDP measurements) and decentralized embodiments so far.

A. Positioning Algorithms

We consider a first standard non-cooperative Linear Least Squares (LS) solution, as represented on Fig. 10, with $X_{ij}^2$, $Y_{ij}^2$ and $Z_{ij}^2$ some vectors containing the known coordinates of the anchors in $MT_i$’s neighborhood, $d_{ij}$ a range measurement with respect to the $j$-th anchor and $(\hat{x}^{(i)}, \hat{y}^{(i)}, \hat{z}^{(i)})_t$ the estimated coordinates of $MT_j$ at time $t$. Accordingly, at each time instant, each MT estimates its current position out of measurements with respect to the three closest anchor nodes. For comparison purposes, a cooperative Weighted Least Squares (WLS) positioning algorithm is also considered (See Fig. 11), for which each MT estimates its current position according to all available range information (i.e. MT-to-ANs and MT-to-MTs) and estimated neighbors locations around (i.e. viewing already positioned MTs as virtual anchors). The inverse of the a priori known standard deviations of the range measurements (i.e. $\sigma_{ij}$ and $\sigma_{ik}$ with respect to neighboring anchors and MTs, respectively) is used to compute the coefficients weighting the quadratic error terms in the cost function.

![Fig. 10. Non-cooperative linear Least Squares positioning algorithm at each $MT_i$, $i = 1..N_m$.](image1)

![Fig. 11. Cooperative Weighted Least Squares positioning algorithm at each $MT_i$, $i = 1..N_m$.](image2)

B. Tracking Algorithms

Two more tracking filters are also proposed to cope with mobility. The first one is a well-known non-cooperative solution, as shown in Fig. 12. Then, the decentralized asynchronous EKF solution initially proposed in [4] has been slightly adjusted into the heterogeneous context (e.g. considering the MAC layers and beaconing procedures implemented in the simulator for ranging purposes, instead of the prioritized access initially suited to WSN). The latter algorithm provides a fully decentralized framework to opportunistically benefit from cooperation, as well as rather convenient means to accommodate the reliability of the peer-to-peer links and neighbors’ locations through the exchange of location estimates covariance matrices $P^{k}_{t-1}$ on Fig. 13. The algorithm is detailed further on Fig. 18 in the Appendix Section hereafter, under the same notations as in [4]. It also authorizes asynchronous updates even in the lack of new measurements through local state predictions. The inputs required at each time step by each mobile are a) non-cooperative range measurements with respect to available anchors and peer-to-peer (P2P) cooperative range measurements with respect to neighboring MTs, forming the current observation on the one hand, and b) the latest location estimate locally available at each mobile neighbor serving as
IV. SIMULATION RESULTS

A. Simulation Parameters

To evaluate the previous tracking solutions, we consider the Indoor Professional operating environment defined in [5] as the application scenario. Accordingly, we use the simulation parameters reported on Fig. 14.

Multi-standard mobile MTs endowed with IR-UWB and a 2.4GHz narrow-band radio, move independently according to a billiard mobility model, with maximum velocities at 1, 2, 3 and 10 m/s. The physical environment is a 71m x 71m indoor floor divided into 10mx10mx offices, which play a role in the semi-deterministic channel models, as mentioned previously. Regarding the physical layer and propagation channel modeling, we consider a log-distance path-loss exponent $\alpha = \{2, 3.3, 3.3\}$ associated with a rice fading factor $K = \{9, 5, 1\}$, respectively for \{LOS, NLOS, NLOS\} link conditions. Note that the latter parameters impact mostly connectivity and interference conditions. Finally, each simulation trial is run over a duration of 400s, with a beaconing period equal to 1s and a link timeout in \{1s, 2s, 3s\}. As for ranging models, we simply consider in first approximation centered Gaussian noise terms with a standard deviation $\sigma = \{1, 10\}$ m respectively for logical channels 0 (i.e. IR-UWB) and 1 (i.e. Narrowband at 2.4GHz). Note that IR-UWB is only intended with shorter-range cooperative peer-to-peer links, whereas medium-range narrow-band links are used uniquely for non-cooperative links with respect to anchors.

B. Results and Discussions

1) Dynamic Connectivity Conditions: For the previous simulation parameters, we show on Fig. 15 the average number of MT-to-ANs and MT-to-MTs links, averaged over random simulation runs and over all the MTs, for various maximum speeds and beacon timeout values.

At any time instant, it is obvious that the number of available range measurements depends on the experienced location-dependent channel conditions (i.e. path-loss, shadowing, \{LOS, NLOS, NLOS\}, mutual interferences, etc.). Under higher mobility velocities (i.e. 10m/s) and large timeouts (e.g. 3s), MTs are more likely to discover each other at least once within the timeout period, the resulting available number of MT-to-MTs links is thus higher than in the case of lower mobility velocities (e.g. 1m/s). When assuming higher mobility velocity and high timeout periods, the number of available links neighbors in the tables is larger at the price however of less reliable distances/positions estimates. Hence, a trade-off for timeout periods could be reached depending on current MT velocity, so as to provide the best compromise between cooperation diversity and links quality.

2) Location Performances: Fig. 16 shows the average location error over random simulation runs and over all
Fig. 15. Average instantaneous number of links/range measurements (including both MT-to-MTs and MT-to-Anchors) as a function of the elapsed time, with different [timeout, maximum speed] couples: [1s, 1m/s] (a), [3s, 1m/s] (b), [1s, 10m/s] (c), [3s, 10m/s] (d)

the MTs, as a function of the elapsed time (i.e. so-called "users-averaged", "network realizations-averaged" and "noise realizations-averaged" error, according to the notations of [5]), for different speeds and timeout values.

Fig. 16. Average location error (in m) as a function of the elapsed time, with different [timeout, maximum speed] couples: [1s, 1m/s] (a), [1s, 3m/s] (b), [1s, 10m/s] (c), [3s, 10m/s] (d)

Rather trivially, it is clear out of the previous simulation results that cooperative localization/tracking algorithms globally outperform non-cooperative approaches and that tracking algorithms could hardly achieve significant gains in comparison with static localization under too large terminal speed (i.e. 10m/s) and/or too large beacon timeout (no more filtering effects). But more interestingly, location precision is clearly degraded due to ineffective management of the neighbor tables at mobile terminals, as already pointed out (i.e. setting too high timeout periods with respect to the actual terminal velocity). Hence, this results tend to suggest possible solutions, e.g. by adapting at the network layer level the timeout period to fit the estimated MT velocity (i.e. increase/decrease the beaconing period for high/low mobility speed, respectively) or, at the location application level, by censoring the most unreliable/erratic links if the estimated speed exceeds a certain threshold.

The table presented in Fig. 17 summarizes the obtained location error results, averaged over time.

As expected, localization performances are degraded with an increase of the MT velocity and/or the timeout period at the network layer. More specifically, for the cooperative and decentralized EKF scheme, an increase of the MTs velocities does not necessarily induce an increase of the localization error. Indeed, under too low mobility speed (i.e. 1m/s), it seems that the number of distinct cooperative neighbors (MT-to-MTs) asynchronously contacted over the trajectory is too low, hence inducing less cooperation and higher localization errors in comparison with medium speed regimes (e.g. 2m/s).

One final but more general comment is that achieving very fine tracking precision (say on the order of 1m-precision, as aimed in [5]) in such tested dynamic heterogeneous environments looks challenging, most probably because of a still too ineffective management of the instantaneous connectivity conditions at MTs, hence pushing to figure out new algorithmic and protocol solutions in that direction in the near future. However, to mitigate these comments, it is also worth noting that rather conservative ranging error levels were considered herein as a function of the involved RATs (e.g. $\sigma = 10m$ with the narrow-band radio technology). Hence, specific coordinated efforts and actions are currently made out in the frame of WHERE2-WP2 so as to incorporate in the short term more realistic and common sensor error models.

V. CONCLUSION

In this paper, we have presented a global simulation framework specifically designed in the frame of WHERE2-T2.1 to be suited to heterogeneous, cooperative and mobile contexts. More precisely, this framework comprises a mobile traces generator, coupled with a multi-RATs cross-layer packet-oriented simulator generating realistic connectivity and location-dependent measurement profiles under terminals mobility and multi-users interference. This framework was considered to evaluate the performance of various positioning and tracking algorithms, including decentralized and cooperative embodiments, under realistic deployment scenarios in part compatible with [5]. On this occasion, interesting trade-offs have also been highlighted regarding optimal timeout values for ranging answers depending on local terminal velocity, hence disclosing interesting possibilities of improvement. The
<table>
<thead>
<tr>
<th>MTs &amp; Protocol Stack Parameters</th>
<th>Link Timeout (s)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>10</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Cooperative LS</td>
<td>201.68 (± 301.99)</td>
<td>201.62 (± 317.36)</td>
<td>179.05 (± 269.07)</td>
<td>187.37 (± 224.32)</td>
<td>160.40 (± 222.54)</td>
<td>186.29 (± 259.61)</td>
<td>157.95 (± 226.41)</td>
<td>147.04 (± 192.28)</td>
<td></td>
</tr>
<tr>
<td>Cooperative WLS</td>
<td>9.55 (± 0.31)</td>
<td>9.58 (± 0.30)</td>
<td>9.59 (± 0.29)</td>
<td>10.09 (± 0.31)</td>
<td>8.40 (± 0.24)</td>
<td>8.44 (± 0.26)</td>
<td>8.59 (± 0.25)</td>
<td>10.98 (± 0.29)</td>
<td></td>
</tr>
<tr>
<td>Non-Cooperative EKF</td>
<td>4.33 (± 0.90)</td>
<td>5.03 (± 0.56)</td>
<td>5.53 (± 0.51)</td>
<td>9.51 (± 0.42)</td>
<td>4.48 (± 0.57)</td>
<td>5.05 (± 0.56)</td>
<td>5.47 (± 0.50)</td>
<td>9.00 (± 0.38)</td>
<td></td>
</tr>
<tr>
<td>Cooperative EKF</td>
<td>3.07 (± 1.56)</td>
<td>2.57 (± 1.40)</td>
<td>2.78 (± 1.33)</td>
<td>6.06 (± 0.96)</td>
<td>2.66 (± 1.40)</td>
<td>2.55 (± 1.37)</td>
<td>3.04 (± 1.12)</td>
<td>8.09 (± 0.76)</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 17. Time-average location performances for the various positioning and tracking tested in the dynamic heterogeneous environment, for different [timeout, maximum speed] couples.

aim will be a more efficient management of cooperative neighborhoods, and hence better location performances accordingly.

Besides tracking considerations, as regards to the selection/censoring of the best links, neighbors and/or messages, most of the selection schemes proposed so far have been concentrating on non-cooperative localization, although more recent approaches coupled with decentralized static positioning apply selection to peer-to-peer messages and measurements as well. At this point, various selection criteria may be considered, e.g. based on theoretical localization performances/limits, e.g. approximated (Generalized) Cramer Rao Lower Bound ((G)CRLB) or Square Position-Error Bound (SPEB) [10]-[11], Game theory and coalitional games, e.g. [12]-[13], estimated group uncertainty via equivalent continuous-time steady-state covariance matrix [14]. Unfortunately, since mobility is definitely crucial in the very context of interest, new censorship strategies more adapted to the chosen decentralized and cooperative tracking filter still have to be proposed. They may include Tx censoring (e.g. Broadcast or censor local positional information according to cooperation level, etc.) and/or Rx censoring (e.g. Censor P2P links according to Link Quality Indicator (LQI), CRLB, Geometric Dilution of Precision (GDOP), latency, etc.). They also represent interesting perspectives to the work presented in this report, which will be also investigated in the frame of WHERE2-T2.1 in the upcoming months.

**ACKNOWLEDGMENT**

This work has been performed in the framework of the WHERE2 (ICT-248894) project, which is partly funded by the European Union.

**APPENDIX**

Fig. 18 shows the detailed synopsis (at the network level) of the preferred cooperative, decentralized and asynchronous tracking EKF solution, as a function of the time step \( k = 1..K \) and with \( N_m \) MTs to be located, under the same notations as in [4].

<table>
<thead>
<tr>
<th>REFERENCES</th>
</tr>
</thead>
</table>

For \( k = 1..K \)

For \( i = 1..N_m \)

\[
\begin{align*}
\dot{S}^{(i)} &= P_{S_i}^{(i)} \\
C_{i}^{(i)} &= P_{C_i}^{(i)} + P_{F_i}^{(i)} + P_{T_i}^{(i)} \\
\end{align*}
\]

end

\[
\begin{align*}
T_i &= S_i^{(i)} + W_i^{(i)} \\
R_i &= Q_i^{(i)} + K_i^{(i)} \\
S_i &= S_i^{(i)} + W_i^{(i)} + K_i^{(i)} \\
C_i &= C_i^{(i)} + K_i^{(i)} \\
\end{align*}
\]

end


A.8 A Two-Phases NBP-Based Localization Solution for Ambiguity Mitigation

A Two-Phases NBP-Based Localization Solution for Ambiguity Mitigation Hadi Noureddine (MERCE)
A Two-Phases NBP-Based Localization Solution for Ambiguity Mitigation

Hadi Noureddine (MERCE)

November 24, 2011

1 Introduction

Flip ambiguity in cooperative network localization, which corresponds to erroneous geometrical realizations, is a fundamental problem that can result in high errors in the location estimates. It is due either to the lack of measurements necessary to achieve a unique solution or the topology of the network and noises in measurements.

Figure 1(a) depicts a network consisting of three anchors nodes and two target nodes. The nodes $d$ and $e$ have two and four solutions verifying the distance constraints and their exact belief distributions have two and four equiprobable modes, respectively. Graph rigidity theory [1, 2] and semidefinite programming [3, 4] address the unique solvability conditions.

Figure 1(b) depicts a network consisting of three anchor nodes that are nearly collinear. Small errors in the measurements can result in a flip ambiguity as shown in Figure 2. The Cramér-Rao bound (CRB) and the mean square error (MSE) of the maximum likelihood (ML) estimator are plotted in Figure 3(a), where we can notice that at high flip probabilities, the MSE is much higher than the CRB since the flips correspond to discontinuous deformations and the CRB corresponds to unbiased estimators.

Figure 1: (a): The target nodes have ambiguities on their position solutions. (b): The target node is uniquely solvable but the anchors are nearly collinear.

Figure 2: Contour plot of the likelihood function for randomly generated measurements.

Several approaches can be applied to deal with the flip ambiguity.
Figure 3: (a): CRB and MSE of the ML estimate of the network of Figure 1(b). (b): Flip probability.

One approach is to only localize the nodes with low flip probability as in [5, 6]. In the incremental solution of [6], the computed positions are promoted to anchors in the subsequent iterations. A flip ambiguity can degrade the remaining position estimates and propagate in an avalanche fashion. In order to avoid this situation, a test is applied for identifying the nodes with high flip probability and removing them from the localization procedure. While this approach can guarantee a high robustness of the solution, it can leave an important portion of the nodes without being localized.

A second approach is to exploit the connectivity information between the different nodes. The fact that two nodes are not neighbors gives the additional information that they are probably far from each other. This approach has been first considered in [7] using the NBP technique.

Here, our focus is on exploiting the connectivity information using the NBP technique. We present a solution that improves the accuracy and reduces the data communication overhead. This solution performs in two phases: In the first phase, the classical NBP is applied without considering the connectivity information, then, a new algorithm based on estimation in discrete states space is proposed for solving the flip ambiguities. This solution has the advantage of reducing the amount of communicating particles and improving the accuracy of the computed positions.

In some cases, the presence of a priori information may compensate the flip ambiguity, as for example map constraints (e.g., the location of a femto base station or an access point is constrained to be inside the owner’s apartment). The NBP method is suitable for handling map constraints.

Another solution is to perform additional measurements either with a mobile terminal (e.g., Figure 4), or by increasing the transmit power in order to acquire measurements from the far nodes [8]. The feasibility of this solution depends on the deployment scenario and the limitations of the maximum allowed power.

Figure 4: The ambiguity on target node a is eliminated after performing measurements with the mobile target node b which is moving on the dashed line.

NBP performs inference on undirected probabilistic graphs, thus probabilistic models for measurements and connectivity are assumed.
2 Probabilistic modeling

We consider \( N \) fixed nodes scattered in a \( d \)-dimensional space. Each node obtains distance measurements with the set of its neighboring nodes. Nodes are mutually neighbors and the connectivity between the nodes can be described by an undirected graph. Neighboring nodes share the same information (if the observations are different, they can be combined in one sufficient statistic). Let \( x_i \) denote the \( d \)-dimensional position of node \( i \), and \( \hat{d}_{i,j} \) the noisy distance measurement with its neighbor \( j \). If a probabilistic model is assumed for the measurements, the joint a posteriori probability distribution (without considering the connectivity information) factorizes as

\[
p(x_1, \cdots, x_N | \{\hat{d}_{i,j}\}) \propto \prod_{i \in V} \Phi_i(x_i) \prod_{(i,j) \in E} \Psi_{i,j}(x_i, x_j),
\]

where \( V \) is the nodes set and \( E \) is the edges set of the graph describing the network. \( \Psi_{i,j}(x_i, x_j) = p(\hat{d}_{i,j}|x_i, x_j) \) is a pairwise potential function and \( \Phi_i(x_i) = p_i(x_i) \) is the a priori probability on the location of node \( i \).

2.1 Connectivity model based on received power

Two nodes \( i \) and \( j \) are connected if the power \( P_{i,j} \) (in decibel (dB)) at node \( i \) transmitted by node \( j \) (or vice versa) is above a threshold value \( P_{th} \). A widely accepted model for the received power is

\[
P_{i,j} = P_0 - 10n_p \log_{10}(d_{i,j}/d_0) + X_{i,j},
\]

where \( P_0 \) is the average received power at a distance \( d_0 \) (in meters (m)), \( d_{i,j} \) is the true distance between nodes \( i \) and \( j \), \( n_p \) is the path loss exponent and \( X_{i,j} \) is a centered normal random variable of variance \( \sigma^2_{sh} \) representing the shadowing.

The probability that two nodes are connected is a function of their separating distance. By assuming that \( X_{i,j} = X_{j,i} \), this probability is equal to

\[
p_o(x_i, x_j) = \text{probability}(P_0 - 10n_p \log_{10}(d_{i,j}/d_0) + X_{i,j} \geq P_{th})
\]
\[
= \text{probability}(X_{i,j} \geq f(x_i, x_j))
\]
\[
= Q \left( \frac{10n_p \log_{10}(d_{i,j}/R)}{\sigma_{sh}} \right),
\]

where \( Q \) is the Q-function, \( R = 10^{\frac{n_p - P_{th}}{10}} \). Notice that \( p_o = 1/2 \) for \( d_{i,j} = R \).

The power value is available by default for two connected nodes and we can write

\[
p(P_{i,j} = p_{i,j}, \text{nodes } i \text{ and } j \text{ are neighbors}|x_i, x_j) = p(P_{i,j} = p_{i,j}, P_{i,j} \geq P_{th}|x_i, x_j)
\]
\[
= p(P_{i,j} = p_{i,j}|x_i, x_j),
\]

where \( p_{i,j} \) is a particular value taken by \( P_{i,j} \). On the other hand

\[
p(\text{nodes } i \text{ and } j \text{ are not neighbors}|x_i, x_j) = p(P_{i,j} < P_{th}|x_i, x_j)
\]
\[
= 1 - p_o(x_i, x_j).
\]

Let \( E^c = \{(i, j) \in V \times V | i \neq j \text{ and } (i, j) \notin E \} \). The joint likelihood of the power measurements and the connectivity information is

\[
p(\{P_{i,j} = p_{i,j}\}_{(i,j) \in E}; \{P_{i,j} < P_{th}\}_{(i,j) \in E^c} | \{x_i\}).
\]

In general, the shadowing affecting the different links is correlated. The joint distribution of the shadowing values is multivariate Gaussian with a non-diagonal covariance matrix, and equation (6) is not suitable for the application of the NBP. A correlation model is proposed in [9] but is very complex to
apply to localization. Instead, if we assume that the shadowing is independent, then the joint a posteriori
distribution becomes

\[ p \left( \mathbf{x}_1, \cdots, \mathbf{x}_N \left| \{ \tilde{d}_{i,j} \} \right. \right) \propto \prod_{i \in V} \Phi_i(\mathbf{x}_i), \prod_{(i,j) \in E \cup E'} \Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \]  

(7)

with \( \Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = p(\tilde{d}_{i,j}, P_{i,j} = p_{i,j} | \mathbf{x}_i, \mathbf{x}_j) \) if \( (i, j) \in E \) and \( \Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = 1 - p_0(\mathbf{x}_i, \mathbf{x}_j) \) otherwise.

In [10, 11, 12], the exponential function (8) is assumed as an approximation of (3).

\[ p_o(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\log2 \ d_{i,j}^2 / R^2). \]  

(8)

We mention that site specific models that take into account the layout of the deployment area (e.g.,
Keenan-Motley model) can be considered and easily handled by the NBP method.

3 Two-phases NBP implementation

The probabilistic undirected graph associated to equation (7) is fully connected, and the application of
the NBP to it is very complex as it requires an exchange of messages between all the pairs of nodes. To
reduce this complexity, messages exchange is limited to the \( k \)-step neighbors, where two nodes are \( k \)-step
neighbors if their shortest connecting path has a length of \( k \) edges.

To reach all the \( k \)-step neighbors, a message needs to be broadcast several times. Indeed, to reach
the 2-step neighbors, a message need not to be broadcast by all the direct neighbors as several 2-step
neighbors can be connected to the same direct neighbor. Multipoint relaying techniques can be used to
reduce the number of redundant retransmissions while diffusing the broadcast messages [13].

We propose a solution for reducing the exchange overhead. It is applied in a second phase after finding
the beliefs with the NBP, during which we consider only direct neighbors. The solution is composed of
the following steps:

1. For each node, we identify the belief’s modes and construct a small set of points consisting of the
   modes and few points around each mode. This step is performed locally and is described in more
details in the following paragraph.

2. At this point, each node will have a small set of points. Let \( S_i \) denote the set associated to node \( i \).
   We apply the BP to find, again, the beliefs where we assume that the positions are discrete random
variables mapping outcomes to the constructed sets.

   • We can use the sum-product rule, and in that case the message from node \( j \) to node \( i \) at
     iteration \( n \) is

   \[ m_{i,j}^{(n)}(s_j^q) = \sum_{l=1}^{|S_i|} \Psi_{i,j}(s_j^q, s_j^l) \prod_{k \in \Omega(j) \cup \Omega_k(j) \setminus i} m_{k,j}^{(n-1)}(s_j^l), \]  

   (9)

   where \( s_j^q \in S_i, q = 1, \cdots, |S_i|, \Omega(i) \) is the set of direct neighbors of \( i \), and \( \Omega_k(i) \) is the set of
   indirect neighbors up to the order \( k \). We compute \( \Psi_{i,j}(s_i, s_j) = p(\tilde{d}_{i,j} | s_i, s_j)p_o(s_i, s_j) \) for direct
   neighbors, and \( \Psi_{i,j}(s_i, s_j) = 1 - p_0(s_i, s_j) \) for indirect ones.

   • If the max-product is used instead, the message is

   \[ m_{i,j}^{(n)}(s_j^q) = \max_{l=1}^{|S_i|} \Psi_{i,j}(s_j^q, s_j^l) \prod_{k \in \Omega(j) \cup \Omega_k(j) \setminus i} m_{k,j}^{(n-1)}(s_j^l). \]  

   (10)

3. The belief at node \( i \) is computed with

   \[ \hat{B}^{(n)}(s_i^q) = \prod_{k \in \Omega(i) \cup \Omega_k(i)} m_{k,i}^{(n)}(s_i^q) \]  

(11)
4. The estimated position is taken as the point with the maximum belief:

\[ \hat{x}_i = \arg \max_{\mathbf{s}_i \in S_i} \hat{B}^{(n)}(\mathbf{s}_i) \].

(12)

With this algorithm, the k-step neighbors are implicated in the message exchange process, but the amount of data contained in the message is much smaller than that of the first phase NBP.

3.1 Modes finding and construction of the sets

The belief probability densities, computed by the NBP, are represented by mixtures of Gaussian distributions. Let the belief of node i be the following summation of M weighted components:

\[ b_i(x) = \sum_{l=1}^{M} w_{i}^{l} \mathcal{N}(x; \mathbf{x}_i^l, \Sigma_i), \]

(13)

where \( w_{i}^{l} \) are non-negative weighting coefficients having the sum equal to one, the points \( \mathbf{x}_i^l \) are the components centroids, and the covariance matrix \( \Sigma_i \) is the same for all the components.

To find all the modes, no direct method exists and iterative numerical algorithms are necessary. The number of the modes is in general less than the number of components, and a hill-climbing algorithm starting from every centroid will not miss any mode. Two such algorithms adapted to the Gaussian components are described in [14].

After finding the modes, the sets are constructed as follows: For each mode \( \mathbf{m}_i^l \), we select \( n \) points on the ellipse

\[ (x - m_i^l)^T (\Sigma_i^l)^{-1} (x - m_i^l) = a, \]

(14)

where \( a \) is a real positive value and \( \Sigma_i^l \) is the covariance of the cluster of samples associated to \( \mathbf{m}_i^l \). In the simulations, we set \( a = 1.5 \) and \( n = 8 \) points. The number of elements of \( S_i \) is \( n \times \) number of modes. This number can be reduced by rejecting the modes with probability smaller than a threshold value.

3.2 Computational complexity

The computational complexity of the NBP method is moderate to high compared to other distributed methods such as distributed EKF [15] and dwMDS [16]. This complexity is mainly due to sampling the product of messages which is a product of Gaussian mixtures. However, the number of iterations needed by the NBP is small.

The computational complexity of the fixed-point algorithm for modes finding is \( O(M^2) \), \( M \) being the number of samples. In the localization problem, the beliefs have only few modes. When \( M \) is large, applying iterative searches that start at every centroid results in a high complexity and many searches will converge to a same mode. Several solutions can be applied to reduce this complexity. One solution is to randomly select a subset of starting points form the \( M \) centroids. Another solution is to approximate the Gaussian mixture with one containing fewer components [17], and this solution allow to reduce the communication but with an additional complexity of finding the components at each iteration.

The construction of the sets is processed at each node locally. The implementation of this step on a desktop PC took on average 2 seconds per node when \( M = 200 \) samples.

The discrete BP step takes less than one second for four iterations in a network consisting of 20 nodes.

4 Numerical results

We consider networks consisting of 4 anchor nodes and 16 target nodes deployed in a \( 20m \times 15m \) area. The nodes are uniformly drawn inside squares of size \( 2m \times 2m \) having the centers on the grid of Figure 5. The shortest distance between two centers is \( 4m \). We divide the target nodes into two categories: the inner nodes nodes that are inside the convex hull of the 4 anchor nodes and the peripheral nodes that are outside this convex hull. The peripheral nodes have higher flip probabilities than the inner ones.
For the connectivity between the nodes, we consider the probabilistic model of equation (3) with range \( R \) varying from 4 to 10 m, and consider only rigid networks (i.e., networks that cannot have a continuous deformation).

![Figure 5: Grid of the centers of squares in which the nodes are uniformly drawn. The size of squares is 2m × 2m.]

We assume that distance measurements are affected by additive Gaussian errors of the same variance \( \sigma^2 \). The centralized benchmark solution considered is the minimization of the weighted least squares described in [18], where the starting point of the steepest descent optimization is the SDP solution. In order to have a fair comparison between the NBP and the centralized solution, the potential function between two neighboring nodes is set equal to the likelihood of the distance measurement (i.e., \( \Psi_{i,j}(x_i, x_j) = p(\tilde{d}_{i,j}|x_i, x_j) \)).

### 4.1 Communication cost

We assume a perfect medium access (no connections drop, packets loss, interference, etc.). The number of messages broadcast by a node depends on the number of iterations, and the number of messages received by a node depends on the number of neighbors and the number of iterations. Figure 6(a) shows the average number of neighbors of a node as function of \( R \).

For the classical NBP, we assume that a node broadcasts a message if it has already computed its belief either from local a priori information or after receiving messages from neighbors. Resampling is performed and all the samples have the same weight, and thus, the exchanged data consists of the samples, and each sample consists of 2 real values, as the space dimension is 2.

Figure 6(b) shows the variation with \( R \) of the average total number of messages broadcast and received per target node when the number of iterations is four, and Figure 6(c) shows the corresponding amount of real data values sent and collected by a target node when \( M = 200 \) samples.

When messages are exchanged between \( k \)-step neighbors, multipoint relays (MPRs) are considered in order to to minimize the number of messages retransmissions. We choose the MPRs using the heuristic algorithm described in [13]. Figure 7(a) shows the probability that a node has 2-step and 3-step neighbors. The average number of retransmissions (or number of MPRs) of a message to reach the 2-step and 3-step neighbors is plotted in Figure 7(b). We can notice that for \( R > 6 \) m, the number of MPRs is decreasing as the number of direct neighbors is increasing.

The message sent by a node to its \( k \)-step neighbors can be either transmitted by dedicated transmissions in which case every node broadcasts its message then MPRs broadcast the received messages and so on, or it can be concatenated to the message sent to direct neighbors in which case the messages from \( k \)-step neighbors are deferred \( k - 1 \) iterations.

In Figure 8(a) we plot the average total number of messages broadcast and received per node for the classical NBP and the two-phases NBP with messages exchange up to the 2-step neighbors. The amounts of exchanged data per node are plotted in Figure 8(b). The number of iterations is fixed to 4.
Figure 6: (a): Average number of neighbors of a node. (b): Average number of transmitted and received messages. (c): Average number of transmitted and received real data values. The number of samples is $M = 200$ and the number of iterations is four.

Figure 7: (a): Probability that a node has a $k$-step neighbor. (b): Average number of retransmissions per node to reach all $k$-step neighbors.

We can notice that the number of messages is increased in the two-phases solution as each phase runs four iterations. Nevertheless, this number can be reduced by considering less iterations in the second phase as the beliefs are already computed and the messages from the 2-step neighbors are used to eliminate the ambiguity. On the other hand, the overall amount of exchanged data is decreased as the exchanges occur between the direct neighbors in the first phase with $M = 200$ samples and fewer samples are exchanged in the second phase.

4.2 Localization accuracy

Figures 9 and 10 show the variation of the root mean square error $\bar{\epsilon}$ with the range $R$ for error standard deviations $\sigma = 0.5m$ and $\sigma = 1.5m$. $\bar{\epsilon}$ is obtained by averaging 100 noise and network realizations. Several methods are compared. NBP-MMSE and NBP-MAP solutions consider the mean of the belief samples and the most probable mode of the Gaussian mixture, respectively, after applying the classical NBP with messages exchange between direct neighbors only. NBP $k$-step corresponds to the application of the two-phases solution, where at the second phase the discrete version of the BP is applied using the sum-product rule (9). 1-step neighbors are the direct neighbors.

Several points can be deduced from these plots:

- The discrete BP improves the accuracy even when messages are exchanged between direct neighbors only: This is due to the randomness in the NBP and the limitation of the number of samples used in approximating the messages.
- The nodes inside the convex hull have smaller errors and less ambiguities than the peripheral nodes.
Figure 8: (a): Average number of transmitted and received messages. (b): Average number of transmitted and received real data values. Messages are exchanged up to the 2-step neighbors

- Messages exchange up to 2-step and 3-step neighbors brings more improvement for the peripheral nodes. No improvement is observed at high ranges $R$ since the flip probability is small, except for the peripheral nodes when $\sigma = 1.5$ (Figure 10).

The sum-product (9) and max-product (10) rules are compared in Figure 11, where notice that the sum-product performs slightly better than the max-product for this scenario.

The median and the 90% errors variations corresponding to messages exchange between the 2-step neighbors are plotted in Figures 12 and 13.

4.3 Effect of the number of iterations

The latency is monotonically increasing with the number of iterations. To assess the effect of the number of iterations on the accuracy, we consider the network of Figure 14. For the two-phases solution, the same number of iterations is performed during the first and second phase. Node 6 is three hops far from the anchor nodes and starts receiving messages after three iterations, and node 7 is two hops far from the anchor nodes and starts receiving messages after two iterations. It can be noticed from Figure 15 that the estimation accuracy does not improve after four iterations.

References


Figure 9: Variation of root mean square error with $R$. $\sigma = 0.5$.

Figure 10: Variation of root mean square error with $R$. $\sigma = 1.5$.

Figure 11: Variation of root mean square error with $R$ for the sum-product(SP) and max-product(MP) rules. $\sigma = 0.5$. 


Figure 12: Variation of median error $\epsilon_{50}$ with $R$. Messages are exchanged between 2-step neighbors.

Figure 13: Variation of worst case error at 90% error $\epsilon_{90}$ with $R$. Messages are exchanged between 2-step neighbors.

Figure 14: Network of 5 anchors nodes and 9 target nodes.
Figure 15: Variation of the root mean square error with the iteration number of (a): node 6, (b): node 7 and (c): node 8. $\sigma = 0.5$. 
A.9 Work in progress document for WHERE2 D2.1 Technical Report

Work in progress document for WHERE2 D2.1 Technical Report Siwei Zhang, Ronald Raulefs
1 Link Evaluation

For cooperative positioning the MTs’ position estimates are exchanged, which leads to the effect of error propagation. There are ways to reduce this effect, e.g. belief propagation [1], [2], information discarding [3], etc. Another approach is to weight the ranging links differently. In [4] a link is weighted by its type. In this section, we first introduce a weighting scheme similar as in [4], which differs the links by their types. Then a rational link evaluation scheme with low complexity is proposed.

1.1 Weighting by Type

The error is only propagated through the peer-to-peer links. Therefore, it is reasonable to trust the cooperative ranging less than the terrestrial ranging. We define a confidence factor $\beta \in (0, 1]$. For the cooperative link $MT_j \rightarrow MT_i$, instead of using the true ranging variance $(\sigma_{i,j})^2$ for estimation, we use a weighted variance $(\tilde{\sigma}_{i,j})^2$:

$$ (\tilde{\sigma}_{i,j})^2 = \beta \cdot (\sigma_{i,j})^2. $$

(1)

$\beta = 1$ denotes trusting both the cooperative and terrestrial link equally, whereas $\beta \rightarrow 0$ means totally distrusting the cooperative links (equivalent to non-cooperative positioning). A simulation result can be found in Figure 1. We can see that the performance is sensitive to the confidence factor. Weighting the cooperative links with an appropriate confidence factor can bring some improvement, however assigning an inappropriate one may even decrease the accuracy. A reasonable confidence factor could be obtained from experience or some online learning algorithm.
Figure 1: Simulation result of using the link type to weight. 9 MTs are uniformly located in a $30 \times 30$ map, the ranging variances of all the links are set to 0.09 m$^2$. The communication range is 7 m

1.2 Rational Weighting

If we additionally know the accuracy of the neighbors’ estimations, a more rational weighting scheme can be found. For peer-to-peer link $MT_j \rightarrow MT_i$, we assume the ranging measurement error and the $MT_j$’s estimation error in each dimension are normally distributed with known variances, i.e.:

$$
\hat{x}_j = x_j + \epsilon_{x_j}, \quad \epsilon_{x_j} \sim \mathcal{N}(0, \sigma_{x_j}^2),
$$

$$
\hat{y}_j = y_j + \epsilon_{y_j}, \quad \epsilon_{y_j} \sim \mathcal{N}(0, \sigma_{y_j}^2),
$$

$$
\rho_{i,j} = d_{i,j} + \eta_{i,j}, \quad \eta_{i,j} \sim \mathcal{N}(0, \sigma_{\eta_{i,j}}^2).
$$

(2)

We combine $M_j$’s estimation variance with the ranging variance to get an equivalent ranging variance, i.e.:

$$
\tilde{\sigma}_{i,j}^2 = f(\sigma_{x_j}^2, \sigma_{y_j}^2, \sigma_{\rho_{i,j}}^2).
$$

(3)

The equivalent ranging variances can be used to replace the true ones in the positioning algorithms. By doing so, we can consider the $M_j$’s estimation as the position without noise, but the ranging measurements are less reliable. It transfers the cooperative positioning problem into an equivalent non-cooperative positioning one without losing the information of the estimations’ inaccuracy of neighboring MTs. The equivalent ranging variance $(\tilde{\sigma}_{i,j})^2$ can be derived as
follow:

\[ \rho_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \eta_{i,j}} \]

\[ = \sqrt{(x_i - \hat{x}_j + \epsilon_{jx})^2 + (y_i - \hat{y}_j + \epsilon_{jy})^2 + \eta_{i,j}} \]  (4)

and

\[ \Delta \hat{x}_{i,j} \triangleq x_i - \hat{x}_j, \quad \Delta \hat{y}_{i,j} \triangleq y_i - \hat{y}_j, \quad \hat{d}_{i,j} \triangleq \sqrt{(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2} \]  (5)

The equivalent ranging variance states:

\[ (\hat{\sigma}_{i,j})^2 = \mathcal{E} \left[ (\rho_{i,j} - \hat{d}_{i,j})^2 \right] \]

\[ = \mathcal{E} \left[ (\Delta \hat{x}_{i,j} + \epsilon_{jx})^2 + (\Delta \hat{y}_{i,j} + \epsilon_{jy})^2 + \eta_{i,j} \right] - \mathcal{E} \left[ 2\hat{d}_{i,j}\eta_{i,j} \right] \]

\[ = \mathcal{E} \left[ g(\epsilon_{jx}, \epsilon_{jy}) \right] \]

\[ = \sigma_{sx}^2 + \sigma_{sy}^2 + (\sigma_{s})^2 + 2\mathcal{E} \left[ \Delta \hat{x}_{i,j}\epsilon_{jx} \right] + 2\mathcal{E} \left[ \Delta \hat{y}_{i,j}\epsilon_{jy} \right] \]

\[ - \mathcal{E} \left[ g(\epsilon_{jx}, \epsilon_{jy}) \right] \]  (6)

\[ g(\epsilon_{jx}, \epsilon_{jy}) \] can be polynomialized by taking the two variables Maclaurin expansion:

\[ g(\epsilon_{jx}, \epsilon_{jy}) = g(0, 0) + \epsilon_{jx} \frac{\partial g(0, 0)}{\partial \epsilon_{jx}} + \epsilon_{jy} \frac{\partial g(0, 0)}{\partial \epsilon_{jy}} \]

\[ + \frac{1}{2!} \left[ \epsilon_{jx} \frac{\partial^2 g(0, 0)}{\partial \epsilon_{jx}^2} + 2\epsilon_{jx}\epsilon_{jy} \frac{\partial^2 g(0, 0)}{\partial \epsilon_{jx} \partial \epsilon_{jy}} + \epsilon_{jy} \frac{\partial^2 g(0, 0)}{\partial \epsilon_{jy}^2} \right] + \cdots \]  (7)

\[ g(\epsilon_{jx}, \epsilon_{jy}) \] can be approximated by the first order expansion:

\[ g(\epsilon_{jx}, \epsilon_{jy}) \approx g(0, 0) + \epsilon_{jx} \frac{\partial g(0, 0)}{\partial \epsilon_{jx}} + \epsilon_{jy} \frac{\partial g(0, 0)}{\partial \epsilon_{jy}} \]

\[ = 2\hat{d}_{i,j}^2 + 2 \Delta \hat{x}_{i,j}\epsilon_{jx} + 2 \Delta \hat{y}_{i,j}\epsilon_{jy} \]  (8)

Inserting Equation (8) into Equation (6):

\[ (\hat{\sigma}_{i,j})^2 \approx (\sigma_{i,j})^2 + \sigma_{sx}^2 + \sigma_{sy}^2 \]  (9)
If we approximate $g(\epsilon_jx, \epsilon_jy)$ with the second order expansion, the equivalent ranging variance becomes:

$$(\tilde{\sigma}_{i,j})^2 \approx (\hat{\sigma}_{i,j})^2_{\text{2nd}}$$

$$= (\sigma_{i,j})^2 + \sigma_{i,j}^2 + \sigma_{i,j}^2 - \mathcal{E} \left[ \frac{1}{2!} (\epsilon_j^2 \frac{\partial^2 g(0,0)}{\partial \epsilon_j^2} + 2\epsilon_j \epsilon_j \frac{\partial^2 g(0,0)}{\partial \epsilon_j \partial \epsilon_j} + \epsilon_j^2 \frac{\partial^2 g(0,0)}{\partial \epsilon_j^2}) \right]$$

$$= \sigma_{i,j}^2 + \sigma_{i,j}^2 + \sigma_{i,j}^2 - \mathcal{E} \left[ \epsilon_j^2 \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} + \epsilon_j^2 \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} - 2\epsilon_j \epsilon_j \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} \right].$$

With $\sin \hat{\theta}_{i,j} \triangleq \hat{\theta}_{i,j}/a$, the last term of Equation (10) can be reformulated as:

$$\mathcal{E} \left[ \epsilon_j^2 \Delta \hat{y}_{i,j}^2 + \epsilon_j^2 \Delta \hat{y}_{i,j}^2 - 2\epsilon_j \epsilon_j \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} \right] = \mathcal{E} \left[ \epsilon_j^2 \sin^2 \hat{\theta}_{i,j} + \epsilon_j^2 \cos^2 \hat{\theta}_{i,j} - 2\epsilon_j \epsilon_j \sin \hat{\theta}_{i,j} \cos \hat{\theta}_{i,j} \right]$$

(11)

If we consider $\hat{\theta}_{i,j}$ as the observation, Equation (11) can be written as:

$$\mathcal{E} \left[ \epsilon_j^2 \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} + \epsilon_j^2 \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} - 2\epsilon_j \epsilon_j \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} \right]$$

$$= \sin^2 \hat{\theta}_{i,j} \mathcal{E} \left[ \epsilon_j^2 \right] + \cos^2 \hat{\theta}_{i,j} \mathcal{E} \left[ \epsilon_j^2 \right] - 2\sin \hat{\theta}_{i,j} \cos \hat{\theta}_{i,j} \mathcal{E} \left[ \epsilon_j \epsilon_j \right]$$

$$= \sin^2 \hat{\theta}_{i,j} \sigma^2_j + \cos^2 \hat{\theta}_{i,j} \sigma^2_j - 2\sin \hat{\theta}_{i,j} \cos \hat{\theta}_{i,j} \mathcal{E} \left[ \epsilon_j \epsilon_j \right]$$

$$= \sin^2 \hat{\theta}_{i,j} \sigma^2_j + \cos^2 \hat{\theta}_{i,j} \sigma^2_j.$$

(12)

The equivalent ranging variance can be approximated as:

$$(\hat{\sigma}_{i,j})^2 \approx (\sigma_{i,j})^2 + \cos^2 \hat{\theta}_{i,j} \sigma^2_j + \sin^2 \hat{\theta}_{i,j} \sigma^2_j. \quad \text{(13)}$$

In a cooperative positioning system, $\hat{\theta}_{i,j}$ can be estimated by the old location estimate. Another way to estimate $\hat{\theta}_{i,j}$ is getting a rough location estimate at first.

If there is no a priori knowledge of $\hat{\theta}_{i,j}$, we further define $\cos \phi \triangleq \frac{\epsilon_j}{\sqrt{\epsilon_j^2 + \epsilon_j^2}}$. Equation (11) can then be reformulated as:

$$\mathcal{E} \left[ \epsilon_j^2 \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} + \epsilon_j^2 \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} - 2\epsilon_j \epsilon_j \frac{\Delta \hat{y}_{i,j}^2}{\Delta \epsilon_{i,j}^2} \right]$$

$$= \mathcal{E} \left[ (\epsilon_j \sin \hat{\theta}_{i,j} - \epsilon_j \cos \hat{\theta}_{i,j})^2 \right]$$

$$= \mathcal{E} \left[ (\epsilon_j \cos \hat{\theta}_{i,j} + \epsilon_j \sin \hat{\theta}_{i,j})^2 \right]$$

$$= \mathcal{E} \left[ \epsilon_j^2 + \epsilon_j^2 \sin^2(\hat{\theta}_{i,j} - \phi) \right]$$

$$\leq \mathcal{E} \left[ \epsilon_j^2 + \epsilon_j^2 \right]$$

$$= \sigma^2_j + \sigma^2_j.$$

(14)
Because \( E \left[ \left( \epsilon_j x \sin \hat{\theta}_{i,j} - \epsilon_j y \cos \hat{\theta}_{i,j} \right)^2 \right] \geq 0 \) we can obtain:

\[
E \left[ \epsilon_j^2 \frac{\Delta \hat{y}_{i,j}^2}{d_{i,j}^2} + \epsilon_j^2 \frac{\Delta \hat{x}_{i,j}^2}{d_{i,j}^2} - 2\epsilon_j x \epsilon_j y \frac{\Delta \hat{x}_{i,j} \Delta \hat{y}_{i,j}}{d_{i,j}^2} \right] \geq 0. \tag{15}
\]

Combining Equation (10), (14) and (15), the second order expanded equivalent ranging variance can be constrained by:

\[
\left( \tilde{\sigma}_{i,j,\text{low}} \right)^2 \leq \left( \tilde{\sigma}_{i,j} \right)^2 \leq \left( \tilde{\sigma}_{i,j,\text{up}} \right)^2, \tag{16}
\]

where

\[
\left( \tilde{\sigma}_{i,j,\text{low}} \right)^2 = \sigma_{i,j}^2 \tag{17}
\]

and

\[
\left( \tilde{\sigma}_{i,j,\text{up}} \right)^2 = \sigma_{i,j}^2 + \sigma_{j_x}^2 + \sigma_{j_y}^2. \tag{18}
\]

One possible choice of \( \left( \tilde{\sigma}_{i,j} \right)^2 \) could be simply:

\[
\left( \tilde{\sigma}_{i,j} \right)^2 \approx \frac{\left( \tilde{\sigma}_{i,j,\text{low}} \right)^2 + \left( \tilde{\sigma}_{i,j,\text{up}} \right)^2}{2} = \left( \sigma_{i,j} \right)^2 + \frac{1}{2} \left( \sigma_{j_x}^2 + \sigma_{j_y}^2 \right). \tag{19}
\]

A comparison of the true variance, the first order expansion, the second order expansions with and without the angle information can be found in Figure 2 - 4. It shows that, for the small location variances, the performances are similar. When the location variances are large, the second order expansions are much closer to the real variance. When the location variances of different dimensions are changing, the second order expansion with the angle information out performs the other two.

With the equivalent ranging variance, MT is able to work with only local information. The error propagation effect is reduced which improves the performance of distributed cooperative positioning algorithms. It also can be used to fuse the mobility information, i.e. if the variance of neighbors’ mobility is also known, it is straightforward to get an equivalent ranging variance combining the uncertainty of ranging, neighbors’ estimate and neighbors’ movement. The link evaluation scheme raised in this section can be also used in other place. In Chapter 2 the link evaluation scheme helps MT approximating a lower bound of the estimation variance locally with only a limited amount of communications. In Chapter 6, the link evaluation scheme is one of the essential factors to control the resource allocation schemes. The derivation of the equivalent variance is based on the assumption of knowing the neighbors’ position estimation variance and the ranging variance which are statistic features and cannot be extracted without massive samples. Meanwhile, online learning the variances by massive observations may not fulfill the real time requirement. In next chapter, we will introduce a theoretical bound for these variances which can be used to substitute the real variances.
Figure 2: The comparison of the true variance (black net), the first order expansion (green), the second order expansions without the angle information (blue) and the second order expansions with the angle information (red). $\sigma_{jx} = 1$ m, $\sigma_{jy} = 1$ m

Figure 3: The comparison of the true variance (black net), the first order expansion (green), the second order expansions without the angle information (blue) and the second order expansions with the angle information (red). $\sigma_{jx} = 3$ m, $\sigma_{jy} = 5$ m
In this section, we investigate a mathematical tool named Cramér-Rao lower bound (CRLB). The CRLB expresses the lower bound of the variance of any estimation. It was firstly derived by Harald Cramér [5] and Calyampudi Radhakrishna Rao [6]. In the first section, the fundamentals of the CRLB are presented. Then the CRLBs for both non-cooperative and cooperative positioning with independent ranging noise are derived in Section 4. We look into the real system deriving the CRLB of delay estimation using the OFDM signal. Then the new positioning CRLBs with the OFDM signals are stated. At the end of this chapter, a local CRLB approximation is introduced with the link evaluation scheme from the previous chapter. For positioning, the CRLB is often used to evaluate the performance of location estimators. In next chapter we propose several novel resource allocation schemes which use the CRLBs introduced in this chapter as a part of the utility functions.

3 The Fundamentals of CRLB

In [7] Kay derived several kinds of CRLBs:

For a scalar parameter: For an unbiased estimator, the parameter $\alpha$ is estimated with the observation $g$, i.e. $\alpha \approx \hat{\alpha}(g)$. The CRLB theorem states:

$$\text{var}[\hat{\alpha}(g)] \geq \text{CRLB}[\alpha] = \frac{1}{E \left[ \left| \frac{1}{g} \ln p(g|\alpha) \right|^2 \right]}.$$  \hspace{1cm} (20)
An alternative expression states:

$$\text{var}[\hat{\alpha}(g)] \geq \text{CRLB}[\alpha] = \frac{1}{-E\left[ \frac{\partial^2}{\partial \alpha^2} \ln p(g|\alpha) \right]}.$$  \hspace{1cm} (21)

For a scalar parameter and the complex observations with Gaussian noise: If a scalar parameter $\alpha$ is estimated by $m$ complex observations $g_i$ with Gaussian noise:

$$g_i = \mu_i(\alpha) + n_i, \quad i \in [1, \ldots, m], \quad E[|n_i|^2] = \sigma^2,$$  \hspace{1cm} (22)

where $| \cdot |$ denotes the magnitude of a complex number. the CRLB states:

$$\text{var}[\hat{\alpha}(g)] \geq \text{CRLB}[\alpha] = \frac{\sigma^2}{2 \sum_{i=1}^{m} \left| \frac{\partial \mu_i(\alpha)}{\partial \alpha} \right|^2},$$  \hspace{1cm} (23)

For multiple parameters: For multiple parameters $\alpha = [\alpha_1, \ldots, \alpha_N]^T$, the CRLB can be extended as a matrix:

$$\text{CRLB}[\alpha] = J[\alpha]^{-1},$$  \hspace{1cm} (24)

where $J[\alpha]$ is called Fischer Information Matrix (FIM). The entity of $J[\alpha]$ is defined as:

$$J[\alpha]_{i,j} = -E\left[ \frac{\partial^2}{\partial \alpha_i \partial \alpha_j} \ln p(g|\alpha) \right].$$  \hspace{1cm} (25)

The variance of the parameter estimate is lower bounded by the diagonal element of CRLB matrix:

$$\text{var} [\hat{\alpha}_i] \geq \text{CRLB}[\alpha]_{(i,i)}.$$  \hspace{1cm} (26)

For multiple parameters and the observations with Gaussian noise: If the observation noises are Gaussian distributed:

$$g \sim \mathcal{N}(\mu(\alpha), C(\alpha))$$  \hspace{1cm} (27)

then

$$J[\alpha]_{(i,j)} = \left( \frac{\partial \mu(\alpha)}{\partial \alpha_i} \right)^T C^{-1}(\alpha) \left[ \frac{\partial \mu(\alpha)}{\partial \alpha_j} \right] + \frac{1}{2} \text{Trace} \left[ C^{-1}(\alpha) \frac{\partial C(\alpha)}{\partial \alpha_i} C^{-1}(\alpha) \frac{\partial C(\alpha)}{\partial \alpha_j} \right].$$  \hspace{1cm} (28)

4 Positioning CRLB with Independent Ranging Noise

We first assume the measurement noise is independent of the location. The positioning CRLBs can be derived as follows:
4.1 CRLB of Non-Cooperative Position Estimation

In non-cooperative positioning, a MT only ranges with its neighboring BSs. We assume the terrestrial range \( \rho_i^k \) is the distance measurement with location independent Gaussian noise:

\[
\rho_i^k = d_i^k + \eta_i^k \quad \eta_i^k \sim \mathcal{N}(0, (\sigma_i^k)^2),
\]

(29)

The marginal log-likelihood function of \( MT_i \) states:

\[
\ln(p(\rho_i^{\text{BS}} | \bar{r}_i)) = \ln \left( \prod_{k=1}^{K_i(i)} \frac{1}{\sqrt{2\pi(\sigma_i^k)^2}} e^{-(d_i^k - \rho_i^k)^2/2(\sigma_i^k)^2} \right)
\]

\[
= \ln \left( \prod_{k=1}^{K_i(i)} \frac{1}{\sqrt{2\pi(\sigma_i^k)^2}} \right) - \sum_{k=1}^{K_i(i)} (d_i^k - \rho_i^k)^2/2(\sigma_i^k)^2,
\]

(30)

where \( \rho_i^{\text{BS}} = [\rho_i^{(1)(i)} \cdots \rho_i^{K_i(i)}]^T \) is the vector of terrestrial ranging measurements. The FIM of \( \bar{r}_i \) (in our case 2-dimensional, but it is in principle straightforward for 3-dimensional extension) is:

\[
J_{nc}[\bar{r}_i] = -E \left[ \frac{\partial^2 \ln(p(\rho_i^{\text{BS}} | \bar{r}_i))}{\partial(\bar{r}_i)^2} \right] = \sum_{k=1}^{K_i(i)} 1/(\sigma_i^k)^2 \left( \begin{array}{c}
\frac{\cos^2 \theta_i^k}{\cos \theta_i^k \sin \theta_i^k} \\
\frac{\cos \theta_i^k \sin \theta_i^k}{\sin^2 \theta_i^k}
\end{array} \right)
\]

\[
= H_{nc,i} C_{nc,i}^{-1} H_{nc,i}^T,
\]

(31)

where \( \theta_i^k = \arcsin((y_i - y^k)/d_i^k) \) is the angle of the \( BS^k \rightarrow MT_i \) link, \( C_{nc,i} \) and \( H_{nc,i} \) are as defined in Equation (??), (??). We can find that \( (J_{nc}[\bar{r}_i])^{-1} \) is the same as WDOP we defined in Section ??.

For an \( M \) MTs non-cooperative positioning system, the FIM of the global parameter vector \( \bar{r}_i = [(\bar{r}_1)^T \cdots (\bar{r}_M)^T]^T \) is a block diagonal matrix with the FIM of each MT along the diagonals:

\[
J_{nc}[\bar{r}_M] = -E \left[ \frac{\partial^2 \ln(p(\rho_i^{\text{BS}} | \bar{r}_M))}{\partial(\bar{r}_M)^2} \right] = \begin{pmatrix}
J_{nc}[\bar{r}_1] & \cdots & 0 \\
0 & \ddots & 0 \\
0 & \cdots & J_{nc}[\bar{r}_M]
\end{pmatrix},
\]

(32)

where the \( \rho_i^{\text{BS}} \) denotes all the terrestrial ranges from all MTs. The global CRLB matrix can be obtained by inverting \( J_{nc}[\bar{r}_M] \), which is equivalent to inverting the diagonal blocks respectively:

\[
\text{CRLB}_{nc}[\bar{r}_M] = (J_{nc}[\bar{r}_M])^{-1} = \begin{pmatrix}
(J_{nc}[\bar{r}_1])^{-1} & \cdots & 0 \\
0 & \ddots & 0 \\
0 & \cdots & (J_{nc}[\bar{r}_M])^{-1}
\end{pmatrix}.
\]

(33)
The variances of the estimated coordinates without cooperation are lower bounded by the corresponding diagonal elements of CRLB_{nc}[\tilde{r}_{MT}]:

$$\text{var}[\hat{\tilde{r}}_{MT,nc}(i)] \geq \text{CRLB}_{nc}[\tilde{r}_{MT}]_{(i,i)}, \quad \forall i \in (1, \ldots, M). \quad (34)$$

### 4.2 Centralized CRLB of Cooperative Position Estimation

In cooperative positioning, besides the terrestrial ranging (BS-to-MT, Equation (29)), MTs also measure the distance from neighboring MTs via peer-to-peer links:

$$\rho_{i,j} = d_{i,j} + \eta_{i,j}, \quad \eta_{i,j} \sim \mathcal{N}(0, (\sigma_{i,j})^2), \quad \text{Equation (35)}$$

where \((\sigma_{i,j})^2\) is the measurement variance of MT\(_j\) → MT\(_i\) link. With this cooperative ranging and the shared (estimated) location from neighbors, MTs are able to enhance their own positioning performance. However, due to the interaction between MTs’ estimates, it is complicated to derive the CRLB for a single MT. In [8], Penna et al. also derived a distributed cooperative positioning CRLB. They calculated the FIM for each user based on its own measurements’ marginal likelihood function and then constructed the global FIM with all the local ones. In this subsection, we derive the centralized CRLB of cooperative positioning directly from the likelihood of the global measurements to find the theoretic lower bound of the estimator performance. With the assumption that all the ranging measurements are mutual independent and Gaussian distributed, the marginal log-likelihood function of the whole system can be written as:

$$\ln (p(\rho_{MT} | \tilde{r}_{MT})) = \ln \left( \prod_{i=1}^{M} \left( \prod_{k \in B_i} \frac{1}{\sqrt{2\pi(\sigma_{i})^2}} e^{-\frac{(d_{i,k}^k - \rho_{k,i})^2}{2(\sigma_{i})^2}} \right) \right)$$

$$= - \frac{1}{2} \sum_{i=1}^{M} \left( \sum_{k \in B_i} \ln (2\pi(\sigma_{i})^2) + \sum_{j \in M_i} \ln (2\pi(\sigma_{i,j})^2) \right)$$

$$- \frac{1}{2} \sum_{i=1}^{M} \left( \sum_{k \in B_i} \frac{(d_{i,k}^k - \rho_{k,i})^2}{2(\sigma_{i})^2} + \sum_{j \in M_i} \frac{(d_{i,j} - \rho_{i,j})^2}{2(\sigma_{i,j})^2} \right), \quad \text{Equation (36)}$$

where \(\rho_{MT}\) is a vector containing the ranges (terrestrial and cooperative) of all the MTs. Similarly as Subsection 4.1, we further assume the ranging variance does not depend on location. i.e. The first term of Equation (36) is a constant with respect to location vector \(\tilde{r}_{MT}\). Therefore, the FIM for the whole system
can be reformulated as:

\[ J[\vec{r}_{MT}] = -E \left[ \frac{\partial^2 \ln (p(\rho_{MT} | \vec{r}_{MT}))}{\partial (\vec{r}_{MT})^2} \right] \]

\[ = \sum_{i=1}^{M} E \left[ \frac{\partial^2}{\partial (\vec{r}_{MT})^2} \left( \sum_{k \in B_i} \frac{(d_{k,i}^2 - \rho_{i,j})^2}{2(\sigma_{i,j})^2} \right) + \sum_{j \in M_i} \frac{(d_{i,j} - \rho_{i,j})^2}{2(\sigma_{i,j})^2} \right] \]

\[ = \sum_{i=1}^{M} E \left( \frac{\partial^2}{\partial (\vec{r}_{MT})^2} \sum_{j \in M_i} \frac{(d_{i,j} - \rho_{i,j})^2}{2(\sigma_{i,j})^2} \right) \]

\[ = J_{nc}[\vec{r}_{MT}] + J_c[\vec{r}_{MT}], \quad \text{(37)} \]

with \( J_{nc}[\vec{r}_{MT}] \) is the cooperative part defined as:

\[ J_{nc}[\vec{r}_{MT}] \triangleq \sum_{i=1}^{M} E \left[ \frac{\partial^2}{\partial (\vec{r}_{MT})^2} \sum_{j \in M_i} \frac{(d_{i,j} - \rho_{i,j})^2}{2(\sigma_{i,j})^2} \right]. \quad \text{(38)} \]

Expanding Equation (38), we can get:

\[ J_c[\vec{r}_{MT}] = \sum_{i=1}^{M} E \left[ \frac{\partial^2}{\partial (\vec{r}_{MT})^2} \sum_{j \in M_i} \frac{(d_{i,j} - \rho_{i,j})^2}{2(\sigma_{i,j})^2} \right] \left( \begin{array}{cccc}
\frac{1}{\delta_{i,j}} & \cdots & \frac{1}{\delta_{i,M}} \\
\cdots & \ddots & \cdots \\
\frac{1}{\delta_{M,j}} & \cdots & \frac{1}{\delta_{M,M}} 
\end{array} \right) \]

\[ \triangleq \left( \begin{array}{cccc}
J_{c,(1,1)} & \cdots & J_{c,(1,M)} \\
\vdots & \ddots & \vdots \\
J_{c,(M,1)} & \cdots & J_{c,(M,M)} 
\end{array} \right), \quad \text{(39)} \]

where \( J_{c,(i,j)} \) is a 2x2 matrix \( \forall i, j \in (1, \ldots, M) \). The block diagonal element \( J_{c,(i,i)} \) can be rewritten as:

\[ J_{c,(i,i)} = \sum_{j \in M_i \setminus i} E \left[ \frac{\partial^2}{\partial (\vec{r}_{MT})^2} \frac{(d_{i,j} - \rho_{i,j})^2}{2(\sigma_{i,j})^2} \right] + \sum_{i \in M_i \setminus i} E \left[ \frac{\partial^2}{\partial (\vec{r}_{MT})^2} \frac{(d_{i,j} - \rho_{i,j})^2}{2(\sigma_{i,j})^2} \right] \]

\[ = \sum_{j \in \{1, \ldots, M\} \setminus i} \left( \delta_{i,j} \sigma_{i,j}^2 \right) \left( \begin{array}{cc}
\cos^2 \theta_{i,j} & \cos \theta_{i,j} \sin \theta_{i,j} \\
\cos \theta_{i,j} \sin \theta_{i,j} & \sin^2 \theta_{i,j} 
\end{array} \right) \]

\[ = \sum_{j \in \{1, \ldots, M\} \setminus i} \frac{\cos^2 \theta_{i,j}}{\sin \theta_{i,j}^2} \left( \begin{array}{cc}
\cos^2 \theta_{i,j} & \cos \theta_{i,j} \sin \theta_{i,j} \\
\cos \theta_{i,j} \sin \theta_{i,j} & \sin^2 \theta_{i,j} 
\end{array} \right) \]

\[ = H^T_{c,i} C_{c,i}^{-1} H_{c,i} \quad \text{(40)} \]

where \( \theta_{i,j} = \arcsin((y_i - y_j)/d_{i,j}) \) is the angle of \( MT_j \rightarrow MT_i \) link, \( \delta_{i,j} \) is the link selection factor:

\[ \delta_{i,j} = \begin{cases} 
1 & \text{if } MT_j \rightarrow MT_i \text{ link is available;} \\
0 & \text{if } MT_j \rightarrow MT_i \text{ link is not available,} \end{cases} \quad \text{(41)} \]
\[ c_{bi,i,j} = \frac{\delta_{i,j}}{(\sigma_{i,j})^2} + \frac{\delta_{j,i}}{(\sigma_{j,i})^2} \quad \text{for simplicity, } C^{-1}_{c,bi,i-} \text{ is the cooperative range weight matrix:} \]

\[
C^{-1}_{c,bi,i-} = \begin{pmatrix}
    c_{bi,i,1} & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & c_{bi,i,M}
\end{pmatrix}
\]  

(42)

and \( H_{c,i} \) is the cooperative geometry matrix:

\[
H_{c,i} = \begin{pmatrix}
    \frac{(\vec{r}_i - \vec{r}_j)^T}{d_{i,j}} \\
    \vdots \\
    \frac{(\vec{r}_i - \vec{r}_{j,k})^T}{d_{i,j,k}} \\
    \vdots \\
    \frac{(\vec{r}_i - \vec{r}_{j,M})^T}{d_{i,j,M}}
\end{pmatrix} = \begin{pmatrix}
    (\vec{e}_{i,1})^T \\
    \vdots \\
    (\vec{e}_{i,j-1})^T \\
    \vdots \\
    (\vec{e}_{i,M})^T
\end{pmatrix}.
\]

(43)

Similarly, the non-block-diagonal elements (\( \forall J_{c,(i,j)} \) with \( i \neq j \)) are symmetric and can be derived as:

\[ J_{c,(i,j)} = J_{c,(j,i)} = \delta_{i,j} \cdot E \left[ \frac{\partial^2 (d_{i,j} - \rho_{i,j})^2 / 2(\sigma_{i,j})^2}{\partial \vec{r}_j \partial \vec{r}_i} + \delta_{j,i} \cdot E \left[ \frac{\partial^2 (d_{j,i} - \rho_{j,i})^2 / 2(\sigma_{j,i})^2}{\partial \vec{r}_j \partial \vec{r}_i} \right] \right] \]

\[ = -c_{bi,i,j}^{-1} \begin{pmatrix}
    \cos^2 \theta_{i,j} & \cos \theta_{i,j} \sin \theta_{i,j} \\
    \cos \theta_{i,j} \sin \theta_{i,j} & \sin^2 \theta_{i,j}
\end{pmatrix} \]

\[ = -\vec{e}_{i,j} \cdot c_{bi,i,j}^{-1} \cdot \vec{e}_{i,j}^T. \]

(44)

If we assume a MT is considered as a neighbor only if both of the bi-directional links are available (mutual neighbor), i.e.:

\[ i \in M_j \Leftrightarrow 1/j \in M_i, \quad \forall i,j \in (1, \cdots, M) \text{ and } i \neq j. \]

(45)

It is reasonable for the peer-to-peer links in the mobile radio networks unlike in GNSS systems, and necessarily be true for RTL measurement. Then Equation (40) can be simplified as:

\[ J_{c,(i,i)} = \sum_{j \in M_i} E \left[ \frac{\partial^2 ((d_{i,j} - \rho_{i,j})^2 / 2(\sigma_{i,j})^2)}{\partial (\vec{r}_j)^2} + \frac{1}{(\sigma_{i,j})^2} \begin{pmatrix}
    \cos^2 \theta_{i,j} & \cos \theta_{i,j} \sin \theta_{i,j} \\
    \cos \theta_{i,j} \sin \theta_{i,j} & \sin^2 \theta_{i,j}
\end{pmatrix} \right] \]

\[ = \sum_{j \in M_i} \left( \frac{1}{(\sigma_{i,j})^2} + \frac{1}{(\sigma_{j,i})^2} \right), \]

(46)

where

\[
\tilde{C}^{-1}_{c,bi} = \begin{pmatrix}
    \frac{1}{(\sigma_{i,1})^2} & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & \frac{1}{(\sigma_{i,M})^2}
\end{pmatrix}
\]

(47)

\( \Leftrightarrow \) is read as 'is equivalent to'.

12
and

\[ \tilde{H}_{c,i} = \begin{pmatrix}
\frac{(\vec{r}_i - \vec{r}_{1(i)})^T}{d_{i,1(i)}} \\
\vdots \\
\frac{(\vec{r}_i - \vec{r}_{M,(i)})^T}{d_{i,M,(i)}}
\end{pmatrix} = 
\begin{pmatrix}
(\vec{e}_{i,1(i)})^T \\
\vdots \\
(\vec{e}_{i,M,(i)})^T
\end{pmatrix}, \tag{48}
\]

Equation (44) can be rewritten as:

\[ J_{c,(i,j)} = J_{c,(j,i)} = \begin{cases}
-\vec{e}_{i,j} \cdot \left( \frac{1}{(\sigma_{i,j})^2} + \frac{1}{(\sigma_{j,i})^2} \right) \cdot (\vec{e}_{i,j})^T & \text{if } j \in M_i \\
0 & \text{if } j \notin M_i \end{cases} \tag{49}
\]

The cooperative contribution can be obtained by inserting Equations (40), (44) (or (46), (49)) into Equation (39), and then we can get the overall FIM \( J[\vec{r}_{MT}] \) by Equation (37). By inverting \( J[\vec{r}_{MT}] \), we can have the global CRLB matrix for a cooperative positioning system:

\[ \text{CRLB}_c[\vec{r}_{MT}] = J[\vec{r}_{MT}]^{-1}. \tag{50} \]

The variances of estimate coordinates with cooperation are lower bounded by the corresponding diagonal elements of \( \text{CRLB}_c[\vec{r}_{MT}] \):

\[ \text{var}[\hat{\vec{r}}_{MT,c}(i)] \geq \text{CRLB}_c[\vec{r}_{MT}]_{(i,i)}, \quad \forall i \in (1, \ldots, M). \tag{51} \]

For bi-directional peer-to-peer links (\( MT_j \rightarrow MT_i \) and \( MT_i \rightarrow MT_j \)), the measurement variances are normally not identical:

\[ (\sigma_{i,j})^2 \not\approx (\sigma_{j,i})^2, \tag{52} \]

because of the different channel state information (CSI) and the adaptive resource allocation scheme. The CRLBs for non-cooperative and cooperative positioning as well as the simulated positioning estimation error are shown in Figure ???. We can see for non-cooperative positioning, the estimation error can achieve the CRLB and is quite stable. For cooperative positioning, the estimation accuracy is also close to the bound. The non-cooperative positioning performances better than the cooperative case because the bandwidth per MT for the latter one is only a half of the former bandwidth, which makes it an unfair comparison.

### 4.3 Approximate the Local Cooperative Positioning CRLB

From the previous derivation, it is clear that for cooperative positioning, the global FIM is normally not a block diagonal matrix. When calculating the global CRLB matrix, by the inverting operation, the entities interact with each other. That means the estimate error of one MT can directly affect the neighboring MTs who use this estimate as their reference. Furthermore, this error may even affect some non-neighboring MTs through some intermediate MTs. Therefore, calculating the positioning CRLB for a specific MT is difficult. On the other hand, for a distributed system there is no central unit collecting the global information. In this condition, how can a MT know its own estimate accuracy? Instead of using the global CRLB matrix, we present an approach
to approximate the lower bound of its own estimate variance only by local information. If the neighbor’s estimate variance is known, recalling Section 1, the link evaluation scheme we introduced allows us transferring the neighbor’s location estimation inaccuracy to an equivalent ranging variance. It likes assuming the neighbor’s estimate is true, but the ranging measurement is less reliable. With this transformation, we are able to calculate an equivalent non-cooperative CRLB just by replacing the true ranging variances by the equivalent ones. At the end of each time step, besides the location estimate, a MT also broadcasts its own local CRLB approximation so that others can use it as its position estimate variance to evaluate this link’s quality. Sequentially, each MT can approximate its positioning CRLB and the extra communication effort is negligible (only one or two more values to share). In figure 6 - 8 the (non-)cooperative positioning CRLB and the local cooperative CRLB approximation snapshots are compared. For this simulation, the second order expansion with the angle information is used. The variances of all the links are set to be the same (64 m) and the neighbors’ initial approximated CRLBs are set to infinity. We can see that the approximated CRLB is close to the true one. Also it is shown that the CRLB decreases when there are more cooperative nodes. Figure 9 shows the iterative approximation procedure averaging from 1000 simulations. It can be seen that the approximated local CRLB converges to a level which slightly higher than the true CRLB in several iterations. Recalling Section 1,
the second order expansion with angle information is slightly higher than the true variance. It could be the explanation why the local CRLB approximation is also slightly higher than the true one.

Figure 6: The comparison of the (non-)cooperative positioning CRLB and the local cooperative CRLB approximation, 2 MTs

Figure 7: The comparison of the (non-)cooperative positioning CRLB and the local cooperative CRLB approximation, 10 MTs
Figure 8: The comparison of the (non-)cooperative positioning CRLB and the local cooperative CRLB approximation, 50 MTs.

Figure 9: The comparison of the (non-)cooperative positioning CRLB and the local cooperative CRLB approximation.

From the simulation results, we can see that the resource allocation schemes we proposed works well.

5 Positioning CRLB with OFDM signal

OFDM is one of the candidate techniques for the next generation mobile radio system due to some key advantages like the spectrum efficiency, the orthog-
nality, the resource flexibility, etc. In this section, we look into a OFDM cooperative positioning system. We first introduce the CRLB of delay estimation (TOA measurement). Then use it as the measurement variance to recalculate the (non-)cooperative positioning CRLB as the bound of the real system performance.

5.1 Ranging CRLB without Pathloss Dependency

An OFDM signal is formulated as [9]:

$$s(t) = \frac{1}{\sqrt{N}} \sum_{n=\left\lfloor -\frac{N-1}{2} \right\rfloor}^{\left\lfloor \frac{N-1}{2} \right\rfloor} S_n e^{j2\pi fnct}$$

(53)

\(f_{sc}\) is the subcarrier spacing, \(N\) is the number of subcarriers and \(S_n\) is the information symbol carried by each subcarrier. Dammann derived the CRLB of the delay estimation in [9] which states:

$$\text{var}[\hat{\tau}] \geq \text{CRLB}[\tau] = \frac{1}{8\pi^2 f_{sc}^2 \sum_{n=\left\lfloor -\frac{N-1}{2} \right\rfloor}^{\left\lfloor \frac{N-1}{2} \right\rfloor} n^2 |S_n|^2 / \sigma_0^2}$$

(54)

$$\text{var}[\hat{\rho}] \geq \text{CRLB}[\rho] = \frac{c^2}{8\pi^2 f_{sc}^2 \sum_{n=\left\lfloor -\frac{N-1}{2} \right\rfloor}^{\left\lfloor \frac{N-1}{2} \right\rfloor} n^2 \text{SNR}_n}.$$  

(55)

From Equation (55) we can see for a given bandwidth and subcarrier spacing, \(\text{var}[\hat{\tau}]\) can be reduced by

1. Increasing the SNR;
2. Increasing the number of used subcarriers;
3. Using the edge subcarriers (ones with higher index) instead of the central ones.

Moreover, the SNR (dB) of a single subcarrier can be formulated according to the pathloss (PL) and thermal noise (\(N_{\text{th,dBm}}\)) [10]:

$$\text{PL}_{\text{dB}} = 10\log(d^2) + 20\log(f_c) + 20\log\left(\frac{4\pi}{c}\right),$$

(56)

$$N_{\text{th,dBm}} = -174\text{dBm/Hz} + 10\log(f_{sc}) + 7\text{dB},$$

the rule of thumb

(57)

$$\text{SNR}_{n,\text{dB}} = P_{\text{tx},n} - \text{PL}_{\text{dB}} - N_{\text{th,dBm}},$$

(58)

where \(P_{\text{tx},n}\) is the transmit power of subcarrier \(n\), \(f_c\) is the carrier frequency. Normally the value of the carrier frequency is much larger than the value of
the bandwidth, i.e. $f_c \gg N f_{sc}$. Therefore, the subcarriers can be assumed to experience the same pathloss. The SNR is inverse proportional to $d^2$:

$$\text{SNR} \propto \frac{1}{d^2}. \quad (59)$$

According to some recent researches [11], the CRLB of TOA estimation with OFDM signal is achievable especially when the SNR is not too low. It is reasonable to assume there is a ranging estimator which can always achieve the CRLB within certain communication range. By this assumption, we are able to focus on the main issues while avoiding an investigation of the details of ranging technique. We model the ranging measurement as follow:

$$\rho = d + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2_{\text{ofdm}}), \quad \text{if } d \leq R \quad (60)$$

where $\sigma^2_{\text{ofdm}} = \text{CRLB}[\rho]$. For writing simplicity, in the remaining part of the thesis, all the variables are referring to the OFDM system without explicitly specified (e.g. $\sigma^2 \equiv \sigma^2_{\text{ofdm}}$), unless stated otherwise.

5.2 Ranging CRLB with Pathloss Dependency

If we take the dependency between pathloss and delay into account, for the same time domain transmit signal as Equation (53), the sampled received signal is:

$$r(iT - \tau) = \frac{1}{\sqrt{N}} \sum_{n=\lfloor -N/2 \rfloor}^{\lfloor N/2 \rfloor} \vartheta(\tau) S_n e^{j2\pi nf_{sc}(iT - \tau)}, \quad (61)$$

where $\vartheta$ is the attenuation factor $N$ is the number of subcarriers and $T = \frac{1}{N f_{sc}}$ is the sampling period. As shown in Equation 23, we need to calculate the derivative to get the CRLB.

$$\frac{d}{d\tau} r(iT - \tau) = \frac{1}{\sqrt{N}} \sum_{n=\lfloor -N/2 \rfloor}^{\lfloor N/2 \rfloor} \left( \frac{d\vartheta}{d\tau} S_n + \vartheta S_n (-j2\pi nf_{sc}) \right) e^{j2\pi nf_{sc}(iT - \tau)}. \quad (62)$$
We can formulate $\sum_{i=0}^{N-1} |\frac{d}{d\tau}r(iT - \tau)|^2$ as:

$$\sum_{i=0}^{N-1} |\frac{d}{d\tau}r(iT - \tau)|^2 = \sum_{i=0}^{N-1} \left( \frac{d}{d\tau}r(iT - \tau) \right)\left( \frac{d}{d\tau}r(iT - \tau) \right)^*$$

$$= \frac{1}{N} \sum_{i=0}^{N-1} \sum_{n,m=-\left\lceil \frac{N-1}{2} \right\rceil}^{\left\lceil \frac{N-1}{2} \right\rceil} \left( \frac{d\vartheta}{d\tau}S_n + \vartheta S_n(-j2\pi nf_{sc}) \right) e^{j2\pi nf_{sc}(iT - \tau)} \cdot \left( \frac{d\vartheta}{d\tau}S_m + \vartheta S_m(j2\pi mf_{sc}) \right)^* e^{-j2\pi mf_{sc}(iT - \tau)}$$

$$= \left\lceil \frac{N-1}{2} \right\rceil \sum_{n,m=-\left\lceil \frac{N-1}{2} \right\rceil}^{\left\lceil \frac{N-1}{2} \right\rceil} S_n S_n^* \left( \frac{d\vartheta}{d\tau} \right)^2 + j2\pi n f_{sc} \vartheta \left( S_n \frac{d\vartheta}{d\tau} S_n^* - S_n \frac{d\vartheta}{d\tau} S_n^* \right) + 4\pi^2 n^2 f_{sc}^2 S_n S_n^* \vartheta^2$$

$$= \left( \frac{d\vartheta}{d\tau} \right)^2 \left( \sum_{n=-\left\lceil \frac{N-1}{2} \right\rceil}^{\left\lceil \frac{N-1}{2} \right\rceil} |S_n|^2 + 4\pi^2 f_{sc}^2 \vartheta^2 \sum_{n=-\left\lceil \frac{N-1}{2} \right\rceil}^{\left\lceil \frac{N-1}{2} \right\rceil} n^2 |S_n|^2 \right). \quad (63)$$

Based on Equation (23), CRLB[$\tau$] is:

$$\text{CRLB}[\tau] = \frac{\sigma_0^2}{2 \sum_{i=0}^{N-1} |\frac{d}{d\tau}r(iT - \tau)|^2} = \frac{\sigma_0^2}{2 \left( \frac{d\vartheta}{d\tau} \right)^2 \sum_{n=-\left\lceil \frac{N-1}{2} \right\rceil}^{\left\lceil \frac{N-1}{2} \right\rceil} |S_n|^2 + 8\pi^2 f_{sc}^2 \vartheta^2 \sum_{n=-\left\lceil \frac{N-1}{2} \right\rceil}^{\left\lceil \frac{N-1}{2} \right\rceil} n^2 |S_n|^2}. \quad (64)$$

$\vartheta$ can be obtained from the pathloss (PL):

$$\text{PL}_{dB} = 10 \log(d^2) + 20 \log(f_c) + 20 \log \frac{4\pi}{c},$$

$$\text{PL} = 10 \frac{\text{PL}_{dB}}{10}. \quad (65)$$

$\vartheta$ states:

$$\vartheta = \frac{1}{\sqrt{\text{PL}}} = \frac{1}{4\pi f_c}, \quad (66)$$

and

$$\left( \frac{d\vartheta}{d\tau} \right)^2 = \frac{1}{16\pi^2 f_c^2 \tau^4}. \quad (67)$$
Then CRLB[τ] can be reformulated as:

$$CRLB[τ] = \frac{σ_0^2}{\frac{1}{8π^2f^2cτ^4} \sum_{n=\lfloor -\frac{N-1}{2} \rfloor}^{\lfloor N-1 \rfloor} |S_n|^2 + \frac{f^2sc^2}{2f^2cτ^4} \sum_{n=\lfloor -\frac{N-1}{2} \rfloor}^{\lfloor N-1 \rfloor} n^2|S_n|^2}.$$  \hspace{1cm} (68)

The ranging variance is lower bounded by:

$$\text{var}[ρ] \geq CRLB[ρ] = CRLB[τ]c^2$$

$$= \frac{σ_0^2c^2}{\frac{1}{8π^2f^2cτ^4} \sum_{n=\lfloor -\frac{N-1}{2} \rfloor}^{\lfloor N-1 \rfloor} |S_n|^2 + \frac{f^2sc^2}{2f^2cτ^4} \sum_{n=\lfloor -\frac{N-1}{2} \rfloor}^{\lfloor N-1 \rfloor} n^2|S_n|^2}.$$  \hspace{1cm} (69)

Compared with the result from the last subsection, there is one more term added in the denominator:

$$\frac{1}{8π^2f^2cτ^4} \sum_{n=\lfloor -\frac{N-1}{2} \rfloor}^{\lfloor N-1 \rfloor} |S_n|^2.$$  \hspace{1cm} (70)

The physical meaning of this is after having a rough estimation, we also take the RSS into account to correct the TOA estimation. Similarly as in the previous subsection, we assume an optimal estimator:

$$ρ = d + η, \quad η \sim N(0, σ^2(d)), \quad \text{if } d \leq R,$$

where $σ^2(d) = CRLB[ρ]$. The CRLBs for delay estimation is shown in Figure 10. We can find out that for a short distance ranging, exploring the RSS-TOA dependency to correct the TOA estimation can offer a significant gain. Whereas when the neighbor is far, this dependency is less helpful and the accuracy of this TOA-RSS hybrid metric is almost the same as to use only the TOA.

### 5.3 Positioning CRLB without Pathloss Dependency

If the position - pathloss dependency is not considered, the SNR is considered as known (e.g. through channel estimation). Therefore, the covariance matrix is not delay dependent. For both non-cooperative and cooperative positioning with the OFDM signals, the CRLB is similar as in Subsection 4.1 (non-cooperative) and Subsection 4.2 (cooperative), just replacing the ranging variance by its CRLB (Equation (55)).

### 5.4 Positioning CRLB with Pathloss Dependency

Because the ranging variance depends on the GLoS distance, the assumption of parameter-independent covariance matrix from Section 4 does not hold anymore. Therefore, new CRLBs for (non-) cooperative positioning need to be derived.

**Non-Cooperative Positioning**  For non-cooperative positioning, $MT_i$ measures the distance to $BS^k$. The ranging variance depends on $\vec{r}_i$ and $\vec{r}^k$ is constant and known. The marginal likelihood function of $MT_i$ (Equation 30) can be modified as:

$$\ln \left( p \left( \rho_{BS}^{i} | \vec{r}_i \right) \right)$$

$$= \ln \left( \frac{1}{(2π)^{\frac{N}{2}} \det \left[ C_{nc,i}(\vec{r}_i) \right] } \cdot \exp \left[ -\frac{1}{2} \left( \rho_{BS}^{i} - d_{BS}^{i} \right)^T \cdot C_{nc,i}^{-1}(\vec{r}_i) \cdot \left( \rho_{BS}^{i} - d_{BS}^{i} \right) \right] \right),$$  \hspace{1cm} (71)

$$((20)$$
Figure 10: CRLB for TOA estimation with(out) the RSS-TOA dependency. Fifty subcarriers are used for each time. Three groups of subcarriers (0-50, 475-524 and 950-1000) are compared.

where \( \mathbf{d}_{i}^{BS} \) is the vector of GLOS distances of all the terrestrial links. As a multi-parameter estimator with gaussian noise, the covariance matrix is parameter-dependent. The FIM can be formulated according to Equation (28):

\[
J_{nc}[\mathbf{r}_i] = H_{nc,i}^T C_{nc,i}^{-1} H_{nc,i} + \frac{1}{2} \begin{bmatrix}
\text{Trace} \left[ C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial x_i} C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial x_i} \right] & \text{Trace} \left[ C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial y_i} C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial x_i} \right] \\
\text{Trace} \left[ C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial y_i} C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial y_i} \right] & \text{Trace} \left[ C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial y_i} C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial y_i} \right]
\end{bmatrix}.
\] (72)

Expanding the upper-left part of the second term:

\[
\text{Trace} \left[ C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial x_i} C_{nc,i}^{-1} \frac{\partial C_{nc,i}}{\partial x_i} \right] = \sum_{k \in B_i} \frac{1}{(\sigma_{k_i}^2)^2} \frac{\partial (\sigma_{k_i}^2)}{\partial x_i} + \frac{1}{(\sigma_{k_i}^2)^2} \frac{\partial (\sigma_{k_i}^2)}{\partial y_i}.
\] (73)

Replacing \( \sigma_{k_i}^2 \) by \( \text{CRLB}[\rho_{k_i}] \) (Equation (69)), \( \frac{\partial (\sigma_{k_i}^2)}{\partial x_i} \) becomes:

\[
\frac{\partial (\sigma_{k_i}^2)}{\partial x_i} = \frac{c^2 (\sigma_{k_i}^2)^3 (x_i - x^k)}{2 \pi^2 f^2_c (d_{k_i}^c)^4} \left( \frac{1}{(d_{k_i}^c)^2} \sum_{n \in N_{k_i}} \frac{|S_n|^2}{\sigma_0^2} + 2 \pi^2 f^2_c \sigma_0^2 \sum_{n \in N_{k_i}} n^2 \frac{|S_n|^2}{\sigma_0^2} \right).
\] (74)
where $N_k^i$ is the index set of subcarriers used by the $MT_i \rightarrow BS_k$ link. Then

$$\frac{1}{(\sigma_k^i)^2} \frac{\partial (\sigma_k^i)^2}{\partial x_i} = 4(x_i - x_k)^2 \frac{c^2 \sum_{n \in N_k^i} |S_n|^2 + 2(d_k^i)^2 \pi^2 f_{sc}^2 \sum_{n \in N_k^i} n^2 |S_n|^2}{c^2 \sum_{n \in N_k^i} |S_n|^2 + 4(d_k^i)^2 \pi^2 f_{sc}^2 \sum_{n \in N_k^i} n^2 |S_n|^2}.$$  

(75)

Then the term $\frac{1}{(\sigma_k^i)^2} \frac{\partial (\sigma_k^i)^2}{\partial x_i}$ becomes:

$$\frac{1}{(\sigma_k^i)^2} \frac{\partial (\sigma_k^i)^2}{\partial x_i} = \frac{16(x_i - x_k)^2}{(d_k^i)^2} \left( A_k^i + B_k^i \right)^2 \cdot \cos^2 \theta_k^i. \quad (76)$$

At the end, the whole second term of Equation (72) can be similarly derived and the $J[\bar{r}_i]$ states:

$$J_{nc}[\bar{r}_i] = H_{nc,i}^T C_{nc,i}^{-1} H_{nc,i} + \sum_{k \in B_i} \frac{8}{(d_k^i)^2} \left( A_k^i + B_k^i \right)^2 \cdot \left( \begin{array}{cc} \cos^2 \theta_k^i & \cos \theta_k^i \sin \theta_k^i \\ \cos \theta_k^i \sin \theta_k^i & \sin^2 \theta_k^i \end{array} \right).$$

(77)

The global FIM is again a block diagonal matrix:

$$J_{nc}[\bar{r}_{MT}] = \begin{pmatrix} J_{nc}[\bar{r}_1] & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & J_{nc}[\bar{r}_M] \end{pmatrix}. \quad (78)$$

**Cooperative Positioning** For cooperative positioning, similarly as we derived in Subsection 4.2 (Equation (37), (39)):

$$J_{c}[\bar{r}_{MT}] = J_{nc}[\bar{r}_{MT}] + J_{c}[\bar{r}_{MT}],$$

$$J_{c}[\bar{r}_{MT}] \triangleq \begin{pmatrix} J_{c,(1,1)} & \cdots & J_{c,(1,M)} \\ \vdots & \ddots & \vdots \\ J_{c,(M,1)} & \cdots & J_{c,(M,M)} \end{pmatrix}. \quad (79)$$

The difference is that $(\sigma_{c,i})^2$ is replaced by the CRLB($\rho_{c,i}$) and it depends on both $\bar{r}_i$ and $\bar{r}_j$. After the derivation similar as the non-cooperative case (Equation (71) - (75)), the non-cooperative contribution $J_{nc}[\bar{r}_{MT}]$ is the same as non-cooperative FIM (Equation (78)). For the cooperative contribution, first we define $A_{i,j}$ and $B_{i,j}$ similarly as in Equation (75):

$$A_{i,j} \triangleq c^2 \sum_{n \in N_{i,j}} |S_n|^2,$$

$$B_{i,j} \triangleq 2(d_{i,j})^2 \pi^2 f_{sc}^2 \sum_{n \in N_{i,j}} n^2 |S_n|^2. \quad (80)$$
where \( N_{i,j} \) denotes the indices set of subcarriers used by the \( MT_j \rightarrow MT_i \) link. For non-diagonal entity \( J_{c,(i,j)} \) where \( i \neq j \):

\[
J_{c,(i,j)} = -\left( \delta_{i,j} \left( \frac{8}{(d_{i,j})^2} \left( \frac{A_{i,j} + B_{i,j}}{A_{i,j} + 2B_{i,j}} \right)^2 + \frac{1}{(\sigma_{i,j})^2} \right) + \delta_{j,i} \left( \frac{8}{(d_{j,i})^2} \left( \frac{A_{j,i} + B_{j,i}}{A_{j,i} + 2B_{j,i}} \right)^2 + \frac{1}{(\sigma_{j,i})^2} \right) \right) \cdot \left( \begin{array}{cc}
\cos^2 \theta_{i,j} & \cos \theta_{i,j} \sin \theta_{i,j} \\
\cos \theta_{i,j} \sin \theta_{i,j} & \sin^2 \theta_{i,j}
\end{array} \right). \tag{81}
\]

For block diagonal entity \( J_{c,(i,i)} \):

\[
J_{c,(i,i)} = \sum_{j=1}^{M} \left( \delta_{i,j} \left( \frac{8}{(d_{i,j})^2} \left( \frac{A_{i,j} + B_{i,j}}{A_{i,j} + 2B_{i,j}} \right)^2 + \frac{1}{(\sigma_{i,j})^2} \right) + \delta_{j,i} \left( \frac{8}{(d_{j,i})^2} \left( \frac{A_{j,i} + B_{j,i}}{A_{j,i} + 2B_{j,i}} \right)^2 + \frac{1}{(\sigma_{j,i})^2} \right) \right) \cdot \left( \begin{array}{cc}
\cos^2 \theta_{i,j} & \cos \theta_{i,j} \sin \theta_{i,j} \\
\cos \theta_{i,j} \sin \theta_{i,j} & \sin^2 \theta_{i,j}
\end{array} \right). \tag{82}
\]

So far, we have derived the (non-) cooperative positioning CRLB considering the location-pathloss dependency. Compared with the result without this dependency, in general one more term \( \frac{8}{\pi^2} \left( \frac{A + B}{\lambda + 2\pi} \right)^2 \) is added to the inverse of the ranging variance. It can be considered as the lower bound of the inaccuracy of a position estimator with the TOA measurement adjusted by RSS.

### 5.5 Approximate the Pathloss Dependent Positioning CRLB

From the ranging CRLB with pathloss dependency (Equation 69) we can see when \( d \) increases, the first term of the denominator \( \frac{\sigma_{i,j}^2}{S_n^2} \sum_{n=1}^{N} \left( \frac{\lambda}{|S_n|^2} \right) \)
decreases much more rapidly than the second term \( \frac{\sigma_{i,j}^2}{2\pi^2} \sum_{n=1}^{N} \left( \frac{\lambda}{|S_n|^2} \right) n^2 \left( |S_n|^2 \right) \). Therefore, in case of long distance, the first term can be ignored. Then for non-cooperative positioning, Equation (72) can be simplified as:

\[
J[f_i] \approx H_{nc,i}^T C_{nc,i}^{-1} H_{nc,i} + \sum_{k \in \mathcal{B}_i} \left( \frac{2(x_i-x_k)^2}{(\sigma_i^2)^4} \left( \frac{2(x_i-x_k)(y_i-y_k)}{d_i^4} \right) \right) \sum_{n=1}^{N} \left( \frac{\lambda}{|S_n|^2} \right) n^2 \left( |S_n|^2 \right) \cdot \left( \begin{array}{cc}
\cos^2 \theta_i^k & \cos \theta_i^k \sin \theta_i^k \\
\cos \theta_i^k \sin \theta_i^k & \sin^2 \theta_i^k
\end{array} \right). \tag{83}
\]

Similarly for the cooperative case,

\[
J_{c,(i,i)} \approx -\left( \delta_{i,j} \left( \frac{2}{(d_{i,j})^2} + \frac{1}{(\sigma_{i,j})^2} \right) + \delta_{j,i} \left( \frac{2}{(d_{j,i})^2} + \frac{1}{(\sigma_{j,i})^2} \right) \right) \cdot \left( \begin{array}{cc}
\cos^2 \theta_{i,j} & \cos \theta_{i,j} \sin \theta_{i,j} \\
\cos \theta_{i,j} \sin \theta_{i,j} & \sin^2 \theta_{i,j}
\end{array} \right), \tag{84}
\]

\]
and

\[ J_{c,(i,i)} \approx \sum_{j=1}^{M} \left( \delta_{i,j} \left( \frac{2}{(d_{i,j})^2} + \frac{1}{(\sigma_{i,j})^2} \right) + \delta_{j,i} \left( \frac{2}{(d_{j,i})^2} + \frac{1}{(\sigma_{j,i})^2} \right) \right) \cdot \begin{pmatrix} \cos^2 \theta_{i,j} & \cos \theta_{i,j} \sin \theta_{i,j} \\ \cos \theta_{i,j} \sin \theta_{i,j} & \sin^2 \theta_{i,j} \end{pmatrix}. \] (85)

The same result can be also obtained by using \( \lim_{d \to \infty} \left( \frac{A + B}{A + 2B} \right)^2 \) to approximate \( \left( \frac{A + B}{A + 2B} \right)^2 \) in Equation (77), (81) and (82).

6 Resource Allocation Scheme for Cooperative Positioning

State of the art research in cooperative positioning assumes the ranging variances of all links are either identical or depend on the distance between the nodes. However, from the derivation in Section 2 we can see that for a real OFDM signal, this variance also depends on the transmit power, carrier frequency, bandwidth, the number and indices of used subcarriers, etc. All of these can be considered as the resources. Using different resources may lead to a quite different performance. The resource independent variance assumption from the previous research only holds when we consider a Time Division Multiple Access (TDMA) system, where all the links use the same spectrum resource within a specific time slot sequentially. In dense networks, the overall processing delay (\( T_{\text{pro}} \)) of such a system is proportional to the total number of the links \( L \) and increases quadratically with the number of MTs:

\[ T_{\text{pro}} \propto L \sim O(M^2). \] (86)

In a static scenario, a high accuracy can be guaranteed because the delay will not cause any additional uncertainty. However, if a system is dynamic, MTs use the neighbors’ old estimates as the references which are less reliable when the delay increases. To avoid this effect, we divide the spectrum resource into small parts to serve multiple links simultaneously (similar as the Frequency Division Multiple Access (FDMA) technique in communications). As already mentioned before, for a real wireless system, the resources are limited. In order to improve the overall performance, a resource optimization scheme is required. Because of the high diversity and the interaction due to the cooperation, it is difficult to get a global optimal solution. Alternatively, we can use some suboptimal approaches like the greedy algorithm. Moreover, recently the usage of game theory in wireless communication has been discussed, especially for a distributed system [12], [13]. Inspired by that, several resource allocation games are proposed for our distributed cooperative positioning system. In this chapter, we mainly consider the problem of allocating the subcarriers. Although the precise definition of resource allocation may also include distributing power, time slot, etc.

In this section a global greedy algorithm is introduced, which directly works with the global positioning CRLB. Then a partial decentralized approach in-
spired by the bidding game is raised to reduce the computational complexity. These two schemes will be compared with a centralized random allocation scheme (i.e. at each time step, certain amount of resources are allocated to a random link). At the end, we look into the purely decentralized case. First we introduce the non-cooperative game from the Nash equilibrium. Then we design a non-selfish utility function for our decentralized resource allocation game to reduce the interference.

6.1 Centralized Greedy Allocation Scheme

From the previous chapter, we already obtained (or at least estimated) a global positioning CRLB matrix \((2M \times 2M)\):

\[
\text{CRLB}[\hat{r}_{\text{MT}}] = \begin{pmatrix}
G_{(1,1)} & \cdots & G_{(1,M)} \\
\vdots & \ddots & \vdots \\
G_{(M,1)} & \cdots & G_{(M,M)}
\end{pmatrix}.
\] (87)

The trace of each block diagonal submatrix \(\text{Trace}[G_{(i,i)}]\) is the lower bound of the position variance in distance for each MT. The mean of these traces denotes the average performance of all the MTs, which refers to the system’s efficiency. Whereas the variance of them shows the performance fluctuations of different MTs which measures the system’s fairness. These two values are used in the cost function \(f_{\text{cost}}\) for the centralized resource optimization scheme to achieve an efficiency-fairness tradeoff:

\[
\text{efficiency: } E = \frac{1}{M} \sum_{i=1}^{M} \text{Trace}[G_{(i,i)}] \] (88)

\[
\text{fairness: } F = \frac{1}{M-1} \sum_{i=1}^{M} (\text{Trace}[G_{(i,i)}] - E)^2 \] (89)

\[
f_{\text{cost}} = \nu_c \cdot E^2 + (1 - \nu_c) \cdot F \] (90)

where \(\nu_c \in [0, 1]\) is the tradeoff factor. When \(\nu_c = 1\) the system is purely efficiency oriented whereas \(\nu_c = 0\) means the system only cares about the fairness.

The resource allocation problem can be formulated as:

\[
\text{SC}_{\text{MT, opt}} = \arg \min_{\text{SC}_{\text{MT}}} f_{\text{cost}} \quad \text{with the resource constraints} \quad 91
\]

where \(\text{SC}_{\text{MT, opt}}\) is the optimal resource allocation strategy. The global CRLB matrix is obtained from the inverse of the global FIM, whose dimension increases with the number of MTs \((M)\). It is difficult to get a real global optimal solution when \(M\) is high. Alternatively, the greedy algorithm is used which tries to optimize the sub-problems stepwise. Each time we take one piece of resources (could be a group of subcarriers), try to add them to each link, calculate a potential cost function and at the end assign it to the one with the lowest cost. Even with this scheme, the complexity is quite high. The computational complexity of allocating \(N_{\text{res}}\) pieces of resource is \(O(LM^3N_{\text{res}})\) (assuming the complexity of inverting a \(N \times N\) matrix is \(O(N^3)\)), which will dramatically grow when the network density increases. Besides, as a centralized approach, a central unit with very high computation capacity is required.

25
6.2 Partial Decentralized Allocation Bidding Game

The resource allocation problem can also be analogically considered as a bidding game. A central resource pool contains all the free resources and works as a coordinator. Each candidate MT acts as a player of this game. At each time step, the resource pool chooses some resources and the players bid for it. The resource will be assigned to the player who offers the highest price. In our case, the price from each player is designed to denote how much improvement it will get with this additional resource.

We derived an approximation of local positioning CRLB in Subsection 4.3 and the positioning CRLB with the OFDM signals in Section 5. A MT \((MT_i)\) can add the potential new resources to each of its links and calculate the approximated local positioning CRLBs. The MT takes the smallest one and names it \(\text{CRLB}[\vec{r}_i]_{\text{loc,new}}\). The potential improvement can be obtained by subtracting \(\text{CRLB}[\vec{r}_i]_{\text{loc,new}}\) from the current local CRLB approximation \(\text{CRLB}[\vec{r}_i]_{\text{loc,cur}}\). The value of this improvement are transmitted to the central resource pool as the bidding price of \(MT_i\):

\[
\text{Price}_i = \text{CRLB}[\vec{r}_i]_{\text{loc,cur}} - \text{CRLB}[\vec{r}_i]_{\text{loc,new}}.
\] (92)

The resource pool compares the prices from all the MTs and gives this resource to the one with the highest price. This resource will be used for this specific link.

The local CRLB matrix is the inverse of the local FIM (only \(2 \times 2\)). For a dense network the complexity of it can be neglected. Unlike the centralized greedy approach, for the bidding game the computations take place at both the MTs and the central unit. The MT's complexity linearly depends on the number of neighbors \(O((M_i + K_i)N_{\text{res}}))\). The central unit's complexity depends on the number of MTs \(O(MN_{\text{res}}))\).

6.3 Decentralized Resource Allocation Game

For a purely decentralized system, each MT chooses the resources by itself. It is difficult to find an allocation scheme which works individually, meanwhile achieves the global optimal performance. A more severe problem is the interference. As there is no central coordinator available, a MT is not aware of which resources are used by others. Therefore, the subcarriers may be reused by multiple links, which leads to interference. The SNR will be replaced by the SINR (Signal-to-Interference-and-Noise Ratio). Consequently, the ranging accuracy will decrease. It is a common feature of decentralized systems, sometimes referred to as the Price of Anarchy [14].

To reduce this effect, we assume the system works with a random access protocol. Each MT transmit and to listen the positioning signals only within a certain time window. Instead of designing a specific protocol, we only consider the effect of random access in a statistic sense. i.e. If there are two links sharing the same subcarrier, We assume the chance of having interference is \(p_{\text{intf}}\). It is like exploring the diversity in time domain. When \(p_{\text{intf}} = 0\), the subcarriers are used by links sequentially, like in a TDMA system. Whereas when \(p_{\text{intf}} = 1\), all the links measure simultaneously. With this random access assumption, the interference can be reduced with the cost of increasing the processing delay.
Game theory is a mathematical tool to analyze the rational behaviors of human in a competitive environment. It has been applied to predict politics and economy and to make decision in those areas. There are many types of game in the game theory. The most commonly used one is called the non-cooperative game. The idea of the non-cooperative game theory is as follow:

There are several players in a competitive game known as agents. The agents cannot communicate with each other. An agent (say the \(i^{th}\) one: \(a_i\)) has some candidate strategies \(\lambda_{m(i)} \in \Lambda_i, m = 1, 2, \ldots\), where \(\Lambda_i\) is the strategies set for \(a_i\). An utility function \(u_{m(i)}\) can be formulated based on the strategies chosen by \(a_i\) and the others which evaluates the benefit (can also be the cost, penalty, etc. depending on the type of the game) of this choice. A rational agent will be aware of others’ potential strategies and the corresponding effects to itself. The goal of each agent is to optimize its utility function by applying a specific strategy.

The game theory normally cannot find a global optimum. Instead, it looks for a stable state known as the *equilibrium*. The most famous one is called *Nash equilibrium* (NE, named after John Forbes Nash), which states [15]:

**Definition 1 (Nash Equilibrium)** For a beneficial utility function (the larger the better), the joint strategies \((\lambda_{m(1)}, \lambda_{m(2)}, \ldots)\) is a NE, if no agent can get further improvement (e.g. increasing the utility) by exploring its own strategy diversity, i.e.:

\[
\forall a_i : u_{m(i)} \geq u_{m(i)} \quad \forall \lambda_{m(i)} \in \Lambda_i
\]

In [15] Nash proved that for any finite non-cooperative game, at least one NE point exists. A common example of the non-cooperative game is the prisoner’s dilemma [16]: Assuming two men \((a_1, a_2)\) are arrested without evidence. They cannot communicate to each other, but both of them can decide to confess \((\lambda_1 = \text{confess})\) or to keep silent \((\lambda_2 = \text{silent})\). If both confess, they will be prisoned for three months \((u_{\text{confess}} = 3 \text{ months})\). If both of them keep silent, they will be set free after one month \((u_{\text{silent}} = 1 \text{ month})\). However, if one man (say \(a_1\)) confesses and the other keeps silent, the former will be released immediately as a reward \((u_{\text{confess}} = 0 \text{ months})\) whereas the latter one will be sentenced to one year \((u_{\text{silent}} = 12 \text{ months})\), and vice versa.

The strategies and utilities are shown in Table 1, where the joint utilities of strategies pair \((\lambda_{m(1)}, \lambda_{n(2)})\) is denoted as \((u_{m(1)}, u_{n(2)})\). The utility function represents the penalty. The goal of an agent is to minimize its own utility. From Table 1 we can find no matter what the other decides, the rational reaction for an agent is to confess. Therefore, the only NE for this game is both of them.

<table>
<thead>
<tr>
<th>(a_1)</th>
<th>(a_2)</th>
<th>confess</th>
<th>silent</th>
</tr>
</thead>
<tbody>
<tr>
<td>confess</td>
<td>(3,3)</td>
<td>(0,12)</td>
<td></td>
</tr>
<tr>
<td>silent</td>
<td>(12,0)</td>
<td>(1,1)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Prisoner’s dilemma

6.4 Non-Cooperative Allocation Game
confessing (utilities (3,3)). It is clear that the global optimum is achieved when both of them keep silent (utilities (1,1)). However, without communications, none of the individuals has the motivation to keep silent.

Now come back to our problem: For a decentralized cooperative positioning system, MTs are considered as agents and the strategies set includes adding new subcarriers or not, which subcarriers to add, etc. First we investigate a simple case: Assuming two MTs are going to share two subcarriers (sc$_1$ and sc$_2$). The strategies for each MT should contain not adding a new subcarrier (null), adding one subcarrier (sc$_1$ or sc$_2$) and adding both (sc$_1$, sc$_2$). We further assume if a subcarrier is used exclusively, it will contribute to the utility for one unit, but if a subcarrier is shared with both MTs, due to the interference, it only brings 0.1 unit contribution to each utility. Similarly as the prisoner’s dilemma, a table of strategies and utility can be shown in Table 2. The only NE is obtained when both of the agents use both of the subcarriers. If there is no cooperative agreement, an agent will always behave selfishly and tries to occupy both of the subcarriers, even though 80% of the other joint allocation strategies achieve higher global utilities (except the three along the diagonal). The result can be extended to the scenario with more MTs and subcarriers. With non-cooperative resource allocation game, all MTs are trying to occupy as many subcarriers as possible. As the consequence, the interference will be dominant which may lead to a poor performance.

### 6.5 Resource Allocation Game with Cooperative Behavior

In the previous subsection, a non-cooperative allocation game was introduced, where the price of anarchy problem may be severe. To avoid this, an allocation game with the cooperative behavior is proposed. In this game, a MT is not a greedy agent anymore. It will be satisfied when its targeting estimation accuracy is achieved. Meanwhile, it evaluates the probability to have interference based on the number of used subcarriers, and jointly chooses its resource allocation strategy.

We assume there are $N$ subcarriers and $M$ MTs. for $MT_i$ the current number of subcarriers is defined as $n_{sc,i}$, the targeting accuracy as $\varrho_i$ and the current local CRLB approximate as $\text{CRLB}[\vec{r}_i]_{loc,cur}$. the relative need of improvement $\Delta_i$ is:

$$\Delta_i = \frac{\text{CRLB}[\vec{r}_i]_{loc,cur} - (\varrho_i \cdot \gamma)^2}{\varrho_i^2}, \quad (93)$$

$\gamma$ is the redundancy factor due to the fact that the CRLB may be not achievable by a suboptimal positioning algorithm. Instead of using a deterministic strategy, we use a statistic one. If $\Delta_i > 0$ which means $MT_i$ is not satisfied with
the current accuracy. We design $p_{i,\text{get}}$ as the probability of $MT_i$ getting some new subcarriers. $p_{i,\text{get}}$ depends on the required improvement and the guessed interference condition. Whereas if $\Delta_i < 0$ which means the accuracy is more than enough. In order to reduce the potential interference for others, $MT_i$ is willing to release some subcarriers with the probability $p_{i,\text{release}}$.

Assuming all the other MTs have the same number of subcarriers randomly chosen from all the subcarriers. For $MT_i$, the probability that a single subcarrier is interfered by a specific neighbor ($MT_j$)’s ranging signal is:

$$p_{i,j,\text{one}} = \left(\frac{n_{\text{sc},i} - 1}{N - 1}\right) \cdot p_{\text{intf}}. \quad (94)$$

The probability that a single subcarrier is interfered reads:

$$p_{i,\text{one}} = 1 - (1 - \left(\frac{n_{\text{sc},i} - 1}{N - 1}\right) \cdot p_{\text{intf}})^{(M-1)}$$

$$= 1 - (1 - \frac{n_{\text{sc},i}}{N} \cdot p_{\text{intf}})^{(M-1)}. \quad (95)$$

We define a percentage factor $\zeta \in [0, 1]$. Then the probability that less than $\zeta n_{\text{sc},i}$ subcarriers get interference follows the Bernoulli distribution:

$$p_{\zeta} = \sum_{k=0}^{\lfloor \zeta n_{\text{sc},i} \rfloor} \left(\frac{k}{n_{\text{sc},i}}\right) p_{i,\text{one}}^k (1 - p_{i,\text{one}})^{(n_{\text{sc},i} - k)}. \quad (96)$$

If $\Delta_i > 0$, probability of getting some new resources should be constructed in a way that it monotonically increases with respect to both $\Delta_i$ and $p_{\zeta}$. Moreover, it should not exceed the interval of $[0, 1]$. We propose a function which fulfill the above constraints:

$$p_{i,\text{get}} = \nu_{dc} \cdot e^{-\frac{\Delta_i}{\nu_{dc}}} + (1 - \nu_{dc}) p_{\zeta}, \quad (97)$$

where $\nu_{dc} \in [0, 1]$ is a control factor to control the tradeoff between accuracy improvement and interference avoidance. If $\Delta_i < 0$, the probability of releasing some occupied resources should be designed in a way that it monotonically decreases with respect of both $\Delta_i$ and $p_{\zeta}$. It should also be constrained by the interval $[0, 1]$. Similarly, we design the function as follow:

$$p_{i,\text{release}} = \nu_{dc} \cdot e^{\frac{\Delta_i}{\nu_{dc}}} + (1 - \nu_{dc})(1 - p_{\zeta}). \quad (98)$$

By this strategy, the overall number of used subcarriers is controlled by both current accuracy and chance of interference, which on some levels improves the performance. Figure 11 shows how the probability of getting or releasing subcarriers depends on the current local CRLB approximation and the number of subcarriers a MT already has.

7 Simulation

In this chapter, some scenarios are set. The simulation results with the previously presented schemes are shown. At the end of each scenario, the analysis is presented.
Figure 11: The probability of getting some new subcarriers (upper) and the probability of releasing some current subcarriers (lower). $M = 5$, $N = 1000$, $\varrho_i = 0\,5\, m$, $p_{intf} = 1$, $\zeta = 0.5$ and $\gamma = 0.8$

8 Scenario 1

In Scenario 1, thirteen BSs are located uniformly in a $60m \times 60m$ map, two MTs are located at the predefined positions. All the BSs and the MTs are stationary. We used an OFDM system for positioning which has the total bandwidth of 2 MHz (0MHz-2MHz). The subcarrier spacing is set to 10 KHz and the carrier frequency is set to 5.2 GHz. Each used subcarriers transmits the signal with the power of -30dBm. Only the TOA measurements are considered. We use GN for this scenario. The map setup can be found in Figure 12 and the simulation results with different resource allocation schemes are shown in Figure 13. The resource allocation solutions can be found in Figure 14.
Figure 12: The map setup of scenario 1. The red dots are the true positions for both the BSs and the MTs. The blue traces are the connected estimations for MTs.

Figure 13: The comparison of different resource allocation schemes in Scenario 1.
From Figure 13, we can see that in this specific scenario, the resource allocation schemes work well. The result of bidding game is close to the central-
ized greedy algorithm. The greedy algorithm has a much higher complexity: \(O(LM^3)N_{\text{res}}\) (Section 6.1), whereas the bidding game has a relatively low complexity: \(O(MN_{\text{res}})\) for the central unit and \(O((M_i + K^i)N_{\text{res}})\) for each MT. Therefore, we can conclude that the bidding game is more suitable for a real system. The cooperative game performs worse than the greedy scheme and the bidding game but better than the random centralized approach and the non-cooperative game. For a decentralized approach, we find the performance of cooperative game is quite good. For the non-cooperative game, each MT always tries to increase its number of subcarriers. Therefore, after ten iterations, the interference is dominant which makes the error increasing rapidly. From Figure 14 we can see that for the random centralized scheme and the non-cooperative game, the resource are allocated randomly. For the other three, only the links with short distance are chosen. More specifically the allocation solutions from bidding game and greedy scheme are similar. The biggest different between the cooperative game and the other two is that for the cooperative game, the resources will not be spent on the peer-to-peer links.

9 Scenario 2

We set a more general scenario in this section. We use a similar map as in Scenario 1, except there are 5 MTs in the map, initialized with random locations. The MTs can be stationary or moving. The RWP mobility model with the velocity 1 m/s is used to move the MTs. The MTs only move in the centering 50 × 50 area. For the OFDM system, the bandwidth is increased to 10MHz (0MHz-10MHz). In this scenario, the PF with MLF_LT model is used. The map setup can be find in Figure 15 and the simulation results with different resource allocation schemes are shown in Figure 16 - 19.

![Figure 15: The map setup of scenario 2 with the RWP. The red dots are the true positions for both the BSs and the MTs. The blue traces are the connected estimations for MTs. The red lines (mostly overlapped by the blue ones) connect the true position in time, which show the true tracks.](image-url)
Figure 16: The comparison of different resource allocation schemes with stationary MTs in Scenario 2. The dashed lines show the corresponding CRLB.

Figure 17: The comparison of different resource allocation schemes with moving MTs in Scenario 2, without link evaluation.
From Figure 16 we can see for the stationary scenario, the greedy algorithm is the best one. The bidding game and the cooperative game perform similarly. The random centralized is not as good as the former three. The non-cooperative game performs well at the beginning, and then loses the accuracy because the interference becomes severe. We can also see that for stationary case, the performance is close to the CRLB. From Figure 17 we can find the positioning accuracy decreases if the system is dynamic. However, a significant improve-
ment can still be observed by applying the resource allocation schemes. If the link evaluation scheme is used (Figure 18), an improvement can be obtained. If we additionally consider the mobility uncertainty for the link evaluation scheme (Figure 19), a more accurate estimation is achievable. However, in this case, the resource allocation gain is less significant. The greedy scheme and the bidding game perform similarly as randomly allocating the resources. One reason could be that the movement variance information is not correct. In this scenario, the cooperative game performs slightly better than the greedy algorithm and the bidding game. It is interesting to see that the non-cooperative game works even better than the greedy algorithm at the first twenty time steps. In this case, the advantage of resource reusing is more significant than the interference effect. By looking to the curve, an optimal degree of subcarrier reuse can be discovered.

10 Conclusions

We propose a link evaluation scheme to reduce the error propagation effect. The simulation results show that by this scheme, the performance can be improved. It can also be used to consider the dynamic uncertainty which offers a further improvement.

We considered the resources limitation of a real wireless system and proposed several resource allocation schemes. The derived CRLB acts as a crucial factor for allocating the resources. For a centralized system, a greedy algorithm is introduced which divides the problem into many sub-problems and tries to find the optimum for each sub-problem. The complexity of the greedy algorithm is high \( O(LM^3N_{res}) \). To reduce the complexity, a partial decentralized allocation scheme - bidding game is presented. The local approximated CRLB instead of the global CRLB is used to calculate the potential improvement for a MT. The simulation results show that the solution from bidding game is similar as the one from the greedy algorithm. The overall complexity is reduced to \( O(M_i + K_i + M)N_{res} \). For a purely decentralized system, the resource allocation can be considered as a non-cooperative game. we evaluate the Nash equilibrium point where every MT behaves greedy and selfish. The interference becomes severe and jeopardizes the positioning estimation. However, the simulation results show that a certain level of subcarrier reuse can offer some gain. At the end, a cooperative game is presented. As a probabilistic approach, we design the utility function which fulfills the constraints. The simulation results show it works well in both stationary and dynamic cases.

Due to the interference avoidance, the centralized schemes should in general outperforms the decentralized ones. However, from the simulation results we find out that sometimes the cooperative game performs better than the centralized greedy scheme. The reason is that the gain from subcarrier reuse is higher than the interference effect.

References


A.10 Contributions on Hybrid Localization Techniques For Heterogeneous Wireless Networks

Contributions on Hybrid Localization Techniques For Heterogeneous Wireless Networks (Chapter 5)
Mohamed LAARAIEDH
Like in chapter 4, non-hybrid and hybrid localization techniques are investigated in this fifth chapter but in a geometric manner. In this chapter, the problem of localization presented in section 4.1 of the previous chapter is modeled using geometry. A generic algorithm is defined and presented below and it is called Robust Geometric Positioning Algorithm (RGPA). The RGPA is based mainly on the geometric representation of LDPs in a Cartesian coordinates system. We start in section 5.1 by giving the necessary definitions and assumptions which support the proposed geometric algorithm. The second section (5.2) is mainly dedicated to the representation of LDPs in the form of geometric constraints. The technique of fusion of geometric constraints aiming to compute position is also presented in this section. The last section (5.3) presents the whole algorithm flow. Furthermore, carried simulations assuming the scenario defined in previous chapter are presented and discussed in order to evaluate the proposed algorithm and to compare it to algebraic techniques and to the CRLB.

5.1 Geometric Localization Problems: Definitions and Assumptions

A geometric localization problem is based on the concept of constraints. In geometric problems, the meaning of a LDP (i.e. RSSI, TOA, TDOA, and others) is different from both algebraic vision (in which a LDP is a random variable characterized by some uncertainty) and network vision (in which a LDP is a measured value). From the geometric vision, a LDP is seen as a geometric constraint. More explicitly, LDP is seen as a set of points in space that satisfy this geometric constraint. In order to clearly explain this concept, we start in this section by giving some definitions. The concept of constraints will be presented in section 5.2.
5.1.1 Voxel

Like a “pixel” in 2D, a voxel represents a volume element in 3D space (see Figure 5.1). It is the smallest addressable space element; it is the smallest unit of a volume that can be controlled. Voxels are good at representing regularly-sampled spaces that are non-homogeneously filled. Common uses of voxels include volumetric imaging in medicine [125] and representation of terrain in games and simulations [126]. This concept of voxel is the basis of geometric representation of constraints of the proposed algorithm.

![Figure 5.1: A set of voxels in stack. Only one voxel is highlighted.](image1)

![Figure 5.2: A 5-sided prism.](image2)

5.1.2 Prism

In geometry, a \( n \)-sided prism is a polyhedron made of a \( n \)-sided polygonal base, a translated copy, and \( n \) faces joining corresponding sides. Thus, these joining faces are parallelograms. All cross-sections parallel to the base faces are the same shape (see Figure 5.2) [127]. In particular, a room is generally a 4-sided prism with two congruent polygonal faces. The polygonal contour of the room constitutes the two congruent polygonal faces and the other sides are the vertical walls.

5.1.3 Cartesian Coordinate System

An important point which affects the algorithm functioning is the choice of the coordinate system able to address each voxel alone. The proposed RGPA algorithm presented in this chapter is based on the world geodetic system (WGS) used for GPS. The WGS is a standard for use in cartography, geodesy, and navigation. It comprises a standard coordinate frame for the Earth, a standard spheroidal reference surface (the datum or reference ellipsoid) for raw altitude data, and a gravitational equipotential surface (the geoid) that defines the nominal sea level. The latest revision is WGS 84 (dating from 1984 and last revised in 2004) [128]. This coordinate system is the most suitable because it would facilitate the recursive computation of intersections between convex volumes and because it is already tested and verified within GPS systems.
For practical implementations on embedded systems, it is very important to use an indexing technique based on integers encoded on 32 bits. The choice of integers is only related to the purposes of simplicity and embeddability of algorithms. For a cubic voxel with an edge $q = 5\text{mm}$, each axis of the Cartesian coordinate system can be discretized on an interval $\pm D = \pm 2^{31} q = \pm 1073741824\text{km}$ when using only integers encoded on 32 bits. Since the earth radius measures $6370\text{km}$, this discretization is largely sufficient to localize the targeted MS until an altitude of $4367.418\text{km}$. That is, the voxel can be defined with a shorter edge. Nevertheless, satellites positions cannot be addressed with this altitude of $4367.418\text{km}$. A simple calculation reveals that a voxel with an edge of $20\text{cm}$ allows us to address satellites. This is very interesting for hybrid scenarios which include observables coming from satellites. For this type of hybrid scenarios, an encoding on 64 bits can drastically enhance the positioning precision by reducing the voxel volume.

The major advantage that has prevailed in choosing a global coordinate system is the easiness it can offer when devices exchange constraints. Indeed, the constraints available to a newcomer MS in an area can be directly mobilized by the neighbor devices without implementing a special conversion process. Furthermore, the algorithm RGPA can be also implemented properly in a different Cartesian coordinate system with a different origin and voxel size, or even without defining voxel at all if the code is implemented in “floating point”. For illustration, we present below a numerical example which compares the distances between two points obtained respectively by the GPS coordinates and the proposed encoding.

\[
iP_1 - iP_2 = [2210, -3788, -2078]q
\]
\[
(ip_1-ip_2)^2 = [4884100, 14348944, 4318084]q
\]
\[
\sum (ip_1 - ip_2)^2 = 235151128
\]
\[
\sqrt{235151128} = 4852.95
\]
\[
d = 4853q = 24.26\text{m}
\]

Figure 5.3: Distance between the two points as calculated by Google Earth ($d = 24.25\text{m}$).

**Numerical Example**
The GPS coordinates of the chosen points are:

P1: Latitude 48, 117023, Longitude −1.641160, Altitude 120m
P2: Latitude 48, 117163, Longitude −1.640910, Altitude 120m

Using a voxel with $q = 5\text{mm}$, we obtain the two integer coordinates as follows:

\[
iP_1 = [852880570, -24436292, 945113018]q \text{ (12 Bytes)}
\]
\[
iP_2 = [852878360, -24432504, 945115096]q \text{ (12 Bytes)}
\]

The difference between these integers can be exploited to compute the distance $d$
between these two points. The most complex algebraic operation involved in this calculation is the square root. This can easily be implemented by simple dichotomous algorithms. The calculation of the distance with the proposed technique is presented above. Figure 5.3 shows the two points and the distance that separates them on Google Earth. The comparison between the two values reveals a high accuracy performed by the proposed technique of encoding.

5.1.4 Intervals and Boxes

A real interval, denoted \([x, x]\), is defined as a closed and connected subset of \(\mathbb{R}\). The basic operations on intervals are defined as follows [129]:

\[
\begin{align*}
[x] + [y] &= [x + y, x + y] \\
[x] - [y] &= [x - y, x - y] \\
[x] \times [y] &= \min\{xy, x\bar{y}, x\bar{y}, x\bar{y}\}, \max\{xy, x\bar{y}, x\bar{y}, x\bar{y}\} \\
1/ [y] &= [1/y, 1/y] \quad \text{(provided that } 0 \notin [y]) \\
[x] / [y] &= [x] \times 1/ [y]
\end{align*}
\] (5.1)

All continuous basic functions (e.g. \(\sin\), \(\cos\), \(\sqrt{\cdot}\), etc) can be used with intervals. Let \(f\) be one of these functions, we define \(f([x])\) as follows:

\[
f([x]) = \{f(x) \mid x \in [x]\}
\] (5.2)

A n-dimensional box \([x]\) of \(\mathbb{R}^n\) is defined as a Cartesian product of \(n\) intervals \([x_k]\) [129]. In 3D and using the Cartesian coordinate system, a box \([x]\) is denoted \([x, y, z]\) where \([x]\), \([y]\), and \([z]\) are respectively the intervals following the three system axes.

5.2 The Concept of Geometric Constraints

5.2.1 Definition

A geometric constraint is a set of points which satisfy a radio LDP or a location dependent information (LDI) (e.g. inclusion in a room or a building, etc). Possible radio LDPS are RSSI, TOA, TDOA, AOA, and AOD. In this dissertation, we are interested only in RSSI, TOA, and TDOA. For both RSSI and TOA, the geometric constraint takes the form of a spherical shell which is the volume lying between two concentric spheres in 3D and the form of an annulus in 2D. The geometric constraint for TDOA takes the form of a hyperboloid in 3D and a hyperbola in 2D. The 2D forms of these geometric constraints are shown in Figure 5.4. The thickness of these geometric constraints is defined by the LDP uncertainty. Assuming Gaussian error, this thickness is equal to six times the standard deviation of the error (\(\sigma\)) taken of both sides of the true value of the LDP. The choice of six \(\sigma\) is justified by the 3-sigma rule stating that for a normal distribution, nearly all values lie within 3 standard deviations of the mean. The geometric constraint becomes as thick as the LDP is less precise (i.e. as \(\sigma\) is higher). The set of points which satisfy the \(k^{th}\) constraint is called the “feasible set of the constraint” and denoted \(S_k\). We define also the “feasible set of the problem”
5.2 The Concept of Geometric Constraints

$S$ as the intersection of feasible sets of all the constraints (i.e. $S = \bigcap_k S_k$). Both $S_k$ and $S$ can be convex or non-convex volumes. The feasible set of a constraint is always continuous for all considered LDPs by contrast to the feasible set of the problem which can be discontinuous. When a geometric problem presents contradictory constraints, $S$ can be empty and the problem cannot be resolved. In this case, an additional constraint (or a set of constraints) may be necessary to resolve the problem.

![Figure 5.4: The annulus (RSSI or TOA) and the hyperbola (TDOA).](image)

To simplify the representation and the use of a constraint, we define the feasible box of the constraint $[S_k]$ which is a cuboid encompassing the feasible set $S_k$ of the constraint. It is easily defined by an interval on each axis of the coordinate system. This approximation makes easier the use of the constraints. Indeed, the storage and the sharing of the constraint can be performed by only saving the endpoints of three (respectively two) intervals in 3D (respectively in 2D). Besides, the fusing and the intersection of the constraints become easier when adopting this box-based presentation. For both TOA and RSSI, where the constraint takes the form of a spherical shell, the constraint box is defined as the circumscribed cuboid that encompasses this spherical shell. The case of TDOA is more specific because the constraint is infinite. To deal with this specificity, we use the fact that the actual absolute value of the difference of range cannot exceed the distance between the two reference devices with which the TDOA is measured. This property allows us to define the feasible box of the TDOA constraint.

5.2.2 Classification of Geometric Constraints

Geometric constraints can be classified using different criteria. Most important ones are presented below.

Radio vs. Non-Radio Constraints

Radio constraints are constraints which result from a LDP. Possible LDPs are RSSI, TOA, and TDOA. As we said before, other LDPs can be considered such as AOA and AOD. Moreover, some additional radio constraints can be derived from GPS measurements for example. Non-radio constraints are mainly constraints resulting from LDIs.
LDIs are data which are location-dependent but do not result from radio measurements. These informations are usually very precise and firm. The inclusion/exclusion of a device in/of an area are examples of LDIs.

**Soft vs. Hard Constraints**

A geometric constraint can be either soft or hard regarding the device inclusion in the feasible set defined by this constraint. The constraint is said to be hard when the inclusion of the device in the feasible set is certain (i.e. the probability of inclusion is equal to 1). This is usually the case of LDIs. In this case, it is known for example that the device (user) has entered in the room and hasn’t left. Unlike hard constraints, soft constraints are uncertain (i.e. the probability of inclusion is inferior to 1). This is the case of LDPs where the device inclusion in the feasible set is characterized by an uncertainty resulting from the imprecision of measurements.

**Shared vs. Dedicated Constraints**

This criteria is decided with respect to the number of devices (i.e. MS, BS, AP, etc) concerned by the constraint. A constraint which is defined for only one device is a dedicated constraint. LDIs (e.g. an inclusion in an area or a room) is usually a dedicated constraint. The RSSI and TOA constraints are shared between two devices. The TDOA constraint is shared between three devices. While a dedicated constraint can only be used by the considered device, shared constraints can be used in localization of more than one device (i.e. all the devices implied in the constraint).

### 5.2.3 Fusion of Heterogeneous Geometric Constraints

In order to obtain the feasible set of a localization problem, all available constraints should be fused. The fusion of constraints aims to intersect their feasible sets to obtain the feasible set $S$ of the problem. $S$ will then give the solution of the problem and its associated accuracy. As constraints are now presented with their feasible boxes, the intersection will be performed between these boxes. The resulting box will contain $S$ and it is called the feasible box of the problem (denoted $[S]$).

#### 5.2.3.1 Box Based Intersection

Let $K$ be the number of constraints (i.e. the number of LDPs and LDIs). Three scenarios may occur:

1. **Scenario 1 ($s_1$):** All $K$ constraints intersect. In this case, $[S] = \bigcap_{k=1,\ldots,K} [S_k]$.

2. **Scenario 2 ($s_2$):** Only a number $N$ of constraints intersect with $0 < N < K$. In this case, the feasible set of the problem can be defined as the union of intersections obtained with $N$ constraints $[S] = \bigcup_{N} \bigcap_{k=1,\ldots,N} [S_k]$.

3. **Scenario 3 ($s_3$):** All $K$ constraints do not intersect. This case gives $[S] = \emptyset$. A preliminary conclusion is to say that such a problem has no solution and the problem is not localizable.
5.2 The Concept of Geometric Constraints

Figure 5.5 presents an example of these different scenarios using three constraints. While in (s1), the solution is given by the centroid of the obtained \([S]\), in (s2) and (s3) the solution is ambiguous. A technique of constraint widening (CW) is applied in order to get the intersection of all constraints. This technique transforms scenarios (s2) and (s3) into a (s1) scenario. The CW applies a multiplicative factor on each radio constraint to enlarge its feasible set \(S_k\). In order to keep the same ratio between constraints, this multiplicative factor should be proportional to the standard deviation of each LDP. This multiplicative factor is increased gradually until finding an intersection between all feasible boxes. The CW technique is only applied on LDPs and does not concern the LDIs which are firm constraints and cannot be modified. Once the intersection is obtained, the next step is to approximate the feasible set of the problem. The objective is to keep, from the resulting feasible box of the problem, only the set where all constraints are verified (see Figure 5.5-(a)).

Figure 5.5: Intersection between three constraint boxes. The red boxes are the resulting feasible boxes.

5.2.3.2 Characterization of the feasible set of the problem

Once the feasible box \([S]\) of the problem is obtained, it is essential to get the feasible set \(S\) itself. In fact, \(S\) is usually smaller than \([S]\) and the centroid of \(S\) is different from the centroid of \([S]\). This difference may be important and hence the use of \([S]\) centroid may result in a large positioning error. In order to get \(S\) from \([S]\) the set inversion via interval analysis (SIVIA) technique is applied. The principle of SIVIA is to split the initial problem of characterizing \(S\) into a sequence of more manageable tasks [130, 131].

The SIVIA starts by dividing \([S]\) into 8 (4 in 2D) identical smaller boxes \(S_j^{(1)}, \ldots, S_j^{(8)}\). This is done by dividing all three (two in 2D) intervals into two equal smaller intervals.
Then, the constraints will be evaluated conjointly in each of obtained boxes. If a common intersection is found, the box is kept else it is discarded. The obtained boxes can be totally or partially included in \( S \). Let \([S_t]\) denotes the union of all boxes which are totally included in \( S \). These steps result in a new feasible box of the problem which is denoted \([S]^{(1)}\). Recursively, we obtain \([S]^{(i)}\), \(i = 2, 3, ...\) [130, 131]. The stop criteria will be the percentage of \([S_t]\) with respect to \([S]^{(i)}\). Let this percentage be 95%. When reaching this ratio, the resulting \([S_t]\) is taken as the approximation of \( S \). The technique is described by Algorithm 1. An illustration example is also given in Figure 5.6.

![Figure 5.6: Illustration example of SIVIA technique.](https://example.com/figure56.png)
Algorithm 1 Set Inversion Via Interval Analysis

\[
[S_t] \leftarrow \emptyset \\
i \leftarrow 0 \\
\varepsilon \leftarrow 0.95 \\
\text{if } [S] \subseteq S \text{ then} \\
\quad S \leftarrow [S] \\
\text{else} \\
\quad \text{while } \frac{\text{size}([S_t])}{\text{size}([S](i))} < \varepsilon \text{ do} \\
\quad \quad \text{for } j = 1 \text{ to } 8 \text{ do} \\
\quad \quad \quad \text{compute } [S_j^{(i+1)}] \\
\quad \quad \quad \text{if } [S_j^{(i+1)}] \subset S \text{ then} \\
\quad \quad \quad \quad [S_t] \leftarrow [S_t] \cup [S_j^{(i+1)}] \\
\quad \quad \quad \end{proof} \\
\quad \quad \text{end if} \\
\quad \text{end for} \\
\quad i \leftarrow i + 1 \\
\text{end while} \\
S \leftarrow [S_t] \\
\text{end if}
\]

5.2.4 Use of Constraints in a Network

In order to use and share constraints between different devices present in a network, we define the constraint layer array (CLA) concept. The CLA of a device is an array which contains all geometric constraints associated with the device. The number of constraints saved in the CLA is not fixed but it depends on the memory size of the CLA. Each constraint is represented by the following entries:

- Timestamp: The timestamp is the time at which the constraint is recorded by the device. It allows to track the constraint and to remove it in case of expiry. This timestamp is necessary to manage dynamic localization techniques.

- Type: This entry indicates the nature of the constraint (i.e. RSSI, TOA, TDOA, or LDI).

- Parameters: The parameters of the constraint depend mainly on the type of the constraint. In the case of LDI constraint, the parameters are the 6 indexes which delimit the volume of the constraint. In the case of LDPs we save, in addition to the 6 indexes which delimit the feasible box of the constraint, the positions of anchors, the measurement, and its precision (i.e. the variance of measurement).

- Order: The order of a constraint informs about the number of elementary constraints implied. The fusion of constraints of order 1 gives a constraint of order 2. This fusion is necessary when the memory size is not sufficient to save all available constraints. When different constraints are fused, the timestamp, the type, and parameters of each constraint are kept.
• Size: the size of the constraint is saved to facilitate the management of available memory.

These entries take the form of arrays. Let $C_k$ be the $k^{th}$ constraint of the targeted device. $C_k$ is represented by an array as follows:

$$C_k = [\text{Timestamp}|\text{Type}|\text{Parameters}|\text{Order}|\text{Size}]$$  \hfill (5.3)

The CLA is hence defined by the concatenation of all $C_k$ for $k = 1,\ldots, K$:

$$CLA = [C_1|C_2|\ldots|C_K]$$  \hfill (5.4)

5.3 Simulation and Evaluation of The RGPA technique

5.3.1 RGPA Flow

![Diagram of RGPA algorithm steps]

Figure 5.7: Different steps of RGPA algorithm.

The RGPA technique consists in a sequence of simple geometrical calculations aiming to obtain the position of the targeted device. The main steps are presented in Figure 5.7. These steps are described as follows:
5.3 Simulation and Evaluation of The RGPA technique

1. Description in the form of geometric constraints of all radio LDPs or LDIs. As a result of this step, we obtain for each constraint a feasible set \( S_k \), \( k = 1, \ldots, K \) with \( K \) the number of constraints.

2. Definition of the feasible box \([S_k]\) which circumvents the feasible set of each constraint.

3. Fusion of available constraints by intersecting all \([S_k]\): the result of this fusion is a box denoted \([S]\).

4. If all constraints are checked in at least one point of \([S]\), we say that \([S]\) exists and we perform step 5. Else, we perform the constraint widening technique until \([S]\) exists.

5. The feasible set \( S \) is approximated using SIVIA technique applied on \([S]\).

6. The solution is obtained as the centroid of the approximated \( S \).

5.3.2 Simulations and Discussions

Table 5.1: Performances of RGPA, ML, and CRLB applied on non-hybrid localization schemes.

<table>
<thead>
<tr>
<th>LDP</th>
<th>Technique</th>
<th>Cumulative Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>67%</td>
</tr>
<tr>
<td>RSSI</td>
<td>ML</td>
<td>4.034m</td>
</tr>
<tr>
<td></td>
<td>RGPA</td>
<td>4.109m</td>
</tr>
<tr>
<td></td>
<td>CRLB</td>
<td>2.902m</td>
</tr>
<tr>
<td>TOA</td>
<td>ML</td>
<td>1.251m</td>
</tr>
<tr>
<td></td>
<td>RGPA</td>
<td>1.264m</td>
</tr>
<tr>
<td></td>
<td>CRLB</td>
<td>0.836m</td>
</tr>
<tr>
<td>TDOA</td>
<td>ML</td>
<td>1.805m</td>
</tr>
<tr>
<td></td>
<td>RGPA</td>
<td>1.584m</td>
</tr>
<tr>
<td></td>
<td>CRLB</td>
<td>1.386m</td>
</tr>
<tr>
<td>RSSI + TOA</td>
<td>ML</td>
<td>1.096m</td>
</tr>
<tr>
<td></td>
<td>RGPA</td>
<td>1.015m</td>
</tr>
<tr>
<td></td>
<td>CRLB</td>
<td>0.594m</td>
</tr>
<tr>
<td>RSSI + TDOA</td>
<td>ML</td>
<td>1.458m</td>
</tr>
<tr>
<td></td>
<td>RGPA</td>
<td>1.282m</td>
</tr>
<tr>
<td></td>
<td>CRLB</td>
<td>0.842m</td>
</tr>
<tr>
<td>TOA + TDOA</td>
<td>ML</td>
<td>0.936m</td>
</tr>
<tr>
<td></td>
<td>RGPA</td>
<td>0.770m</td>
</tr>
<tr>
<td></td>
<td>CRLB</td>
<td>0.476m</td>
</tr>
<tr>
<td>RSSI + TOA + TDOA</td>
<td>ML</td>
<td>0.913m</td>
</tr>
<tr>
<td></td>
<td>RGPA</td>
<td>0.780m</td>
</tr>
<tr>
<td></td>
<td>CRLB</td>
<td>0.392m</td>
</tr>
</tbody>
</table>
In order to compare the proposed geometric technique to the ML technique which is the most accurate algebraic technique, we consider the same scenario as in chapter 4 (see Figure 4.1 and section 4.1). For each localization scheme, we compare the RGPA technique to the maximum likelihood technique initialized randomly. These two techniques are also compared to the CRLB. To perform these comparisons, we plot in Figure 5.8 and Figure 5.9 the CDFs of absolute localization errors and CRLB for respectively different non hybrid and hybrid schemes. Table 5.1 summarizes these CDFs for non-hybrid and hybrid schemes.

![CDFs of positioning error using RGPA, ML, and CRLB applied on non-hybrid localization schemes.](image)

These figures and tables show that the proposed RGPA technique outperforms the ML technique randomly initialized (and thus, all algebraic techniques) except in the case of TOA scheme where it gives the same performances as the ML technique and in the case of the scheme (RSSI+TOA) where ML and RGPA gives very close performances. The highest gains are performed in schemes involving TDOA. The comparison between the CDFs of the ML and RGPA techniques and the CDFs of the CRLB shows that RGPA is closer to the CRLB than the ML technique.
Figure 5.10 plots the CDFs of all localization schemes using RGPA technique. This figure shows, like in Chapter 4, that TOA and TDOA outperform the RSSI. Moreover, the figure highlights that the fusion of hybrid LDPS enhances the positioning accuracy. According to this figure, the schemes (RSSI+TOA+TDOA) and (TOA+TDOA) offer the best positioning accuracy among all schemes. Furthermore, the adding of TDOA or TOA to RSSI widely enhances the performances of positioning accuracy when comparing to the positioning accuracy performed by the non-hybrid RSSI scheme. These enhancements validate one of the objectives of this thesis which consists in demonstrating that available RSSIs must be aided by time based LDPS in order to reach the requested positioning accuracy.

![CDF plots](image)

(a) RSSI + TOA
(b) RSSI + TDOA
(c) TOA + TDOA
(d) RSSI + TOA + TDOA

Figure 5.9: CDFs of positioning error using RGPA, ML, and CRLB applied on the fusion of RSSI, TOA, and TDOA.
5.4 Conclusion

In this chapter, we have geometrically investigated the localization problem. A generic algorithm RGPA is proposed for resolving localization problems. The basis of RGPA is the geometric representation of LDPs and LDIs in the form of sets of points (feasible sets). This representation is based on interval analysis theory. In order to simplify the computation, a feasible box is defined to encompass each constraint. The intersection between all feasible boxes of a problem give the feasible box of constraints where the solution lies. This box contains the feasible set of the problem. In order to approximate this feasible set, the SIVIA algorithm is applied. Once the feasible set of the problem is obtained, its centroid gives the estimate of the targeted location. The problem of non-intersection between constraints is resolved using a constraint widening technique which consists in conjointly enlarging the feasible sets of all constraints proportionally to their precisions until finding an intersection between them.

The proposed algorithm is promising compared to algebraic techniques. The carried simulations have shown that the RGPA outperforms all algebraic algorithms especially in the case of TDOA where it overcomes singularities occurring when using iterative optimization. Besides, this algorithm may be enhanced to involve tracking and cooperative based localization. Furthermore, the algorithm is easily embeddable in devices and would not consume much resources because it is based only on simple mathematical operations.
A Hybrid Data Fusion Based Cooperative Localization Approach for Cellular Networks

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

A Hybrid Data Fusion Based Cooperative Localization Approach for Cellular Networks
Ziming He, Yi Ma, and Rahim Tafazolli
©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
A Hybrid Data Fusion Based Cooperative Localization Approach for Cellular Networks

Ziming He, Yi Ma, and Rahim Tafazolli
Centre for Communication Systems Research
University of Surrey, Guildford, UK
e-mail: {z.he, y.ma, r.tafazolli} @surrey.ac.uk

Abstract—One of the major challenges of Cellular network based localization techniques is lack of hearability between mobile terminals (MTs) and base stations (BSs), thus the number of available anchors is limited. In order to solve the hearability problem, previous work assume some of the MTs have their location information via Global Positioning System (GPS). These located MT can be utilized to find the location of an un-located MT without GPS receiver. However, its performance is still limited by the number of available located MTs for cooperation. This paper consider a practical scenario that hearability is only possible between a MT and its home BS. Only one located MT together with the home BS are utilized to find the location of the un-located MT. A hybrid cooperative localization approach is proposed to combine time-of-arrival and received signal strength based fingerprinting techniques. It is shown in simulations that the proposed hybrid approach outperform the stand-alone time-of-arrival techniques or received signal strength based fingerprinting techniques in the considered scenario. It is also found that the proposed approach offer better accuracy with larger distance between the located MT and the home BS.

I. INTRODUCTION

Cellular network based localization techniques have recently received increasing interests in both localization and communication community, e.g., [1]-[7]. This is not only because of the request made by Federal Communication Commission (FCC) about the accurate localization of the mobile terminals (MTs), but many other applications about the location information such as location sensitive billing, fleet management, mobile yellow pages, etc. [8]. Previous approaches includes time-of-arrival (TOA) [1], time-difference-of-arrival (TDOA) [2], angle-of-arrival (AOA) [9], received signal strength (RSS) techniques [3], and fingerprinting based approaches [10]. Various signal characteristics, including RSS and multipath delay can be utilized for fingerprinting based localization [10].

The location estimation is based on training sequences sent by MTs or BSs. Usually, the BSs serve as reference nodes (i.e. anchors) for localization. TOA and TDOA approaches generally require at least three BSs for localization. AOA approaches, on the other hand, require only a minimum of two BSs. If the number of available BSs is less than the minimum requirement, ambiguities of location estimation exist and a large estimation error may be introduced. Hybrid techniques, which utilize combinations of the available parameters, have also been proposed, e.g., [9]-[11]. These approaches are especially useful in hearability-restricted conditions, where power control is employed [12].

Cooperative localization is also a suitable approach in hearability-restricted conditions. Previous work about cooperative localization for cellular networks are proposed in [12]-[14]. For example, the work in [14] employs Global Positioning System (GPS) enable MTs (i.e. located MTs) to serve as anchors. Then, at least three located MTs can find the location of an un-located MT. In this case, even if no BS is involved in localization, localization of the un-located MT is possible. However, cooperative localization is still limited by the number of available located MTs for cooperation, especially when the density of MTs is low.

This paper consider a hearability-restricted scenario, which is depicted in Fig. 1. In this scenario, hearability is only possible between an un-located MT and its home BS, which can serves as an anchor. This scenario reflect a practical case, where the un-located MT is located nearing the home BS. In this case, the hearability may not be possible between a MT and its inter-cell BSs due to interference. Since the number of available MTs may also be limited in some practical case, we also assume only one located MT is available to help the un-located MT to find its location.

In the considered scenario, TOA techniques is employed to estimate the distance between the un-located MT and the home BS (or the located MT). The reason we consider TOA based distance estimation is that it offer more accurate distance estimation than RSS based techniques [15]. Moreover, an
RSS fingerprinting database is connected to the home BS, which can estimate the location of the un-located MT if the corresponding RSS information is obtained.

Since there are totally only two anchors available, ambiguity of location estimation exists for TOA approaches. The ambiguities also exist using the RSS based fingerprinting approach. These ambiguities are caused by the same RSS measurement for different location of the un-located MT. Since ambiguities exist for both the TOA approach and RSS based fingerprinting approach in this scenario, the localization accuracy degrades significantly. In order to remove the ambiguities and improve the localization accuracy, a hybrid data fusion based approach is proposed to fuse the estimates obtained from both of the two approaches. The simulation results show that the proposed approach can offer much more accurate location estimation than TOA approaches or RSS based fingerprinting approach in the considered scenario. Moreover, the proposed approach offer better accuracy with larger distance between the located MT and the home BS.

II. SYSTEM MODEL

As depicted in Fig. 1, an un-located MT has only two anchors, including the home BS and the located MT. Training sequences are sent between the un-located MT and the home BS (or the located MT) for TOA estimation. Training sequences for RSS-based fingerprinting approach are sent between the un-located MT and its home BS. Different training signal is sent with different time or frequency to avoid collision. All the estimated TOA and RSS are collected at the home BS for centralized processing. Denote \( \mathbf{a} = (a_x, a_y) \) as the true 2-D location of the located MT, \( \mathbf{u} = (u_x, u_y) \) as the true location of the un-located MT. Without loss of generality, the location of the home BS is set as \( \mathbf{o} = (0, 0) \).

A. Modeling of TOA based distance estimation errors

For simplicity, clock synchronization between the un-located MT and the home BS (or located MT) is assumed for the TOA estimation. In practice, however, the TOA estimation can be implemented by using the Two-way Time Transfer or the Double Token Exchange techniques, where the strict clock synchronization is not necessary [16]. The TOA based distance estimates can be modeled as

\[
\hat{d}_{bs} = d_{bs} + e_{bs} + b_{bs} \quad (1)
\]
\[
\hat{d}_{mt} = d_{mt} + e_{mt} + b_{mt} \quad (2)
\]

where \( d_{bs} \) denotes the true distance between the un-located MT and the home BS, \( d_{mt} \) the true distance between the un-located MT and the located MT, \( \hat{d}_{bs} \) and \( \hat{d}_{mt} \) the corresponding estimates, respectively, \( e_{bs} \) and \( e_{mt} \) the Gaussian noise of \( d_{bs} \) and \( d_{mt} \), \( b_{bs} \) and \( b_{mt} \) the non-line-of-sight (NLOS) error of \( d_{bs} \) and \( d_{mt} \), \( b_{bs} \) and \( b_{mt} \) follow exponential distribution [13] and has p.d.f

\[
p(b) = \begin{cases} \lambda e^{-\lambda b}, & b \geq 0 \\ 0, & b < 0 \end{cases} \quad (3)
\]

where \( b \) denotes NLOS error, \( \mathbb{E}(b) = \frac{1}{\lambda} \).

B. Modeling of the location estimates of the located MT

Since the location of the located MT is obtained using GPS signals, this location information is imperfect. Denote \( \mathbf{\hat{a}} = (\hat{a}_x, \hat{a}_y) \) as the estimated location using GPS. The estimated location is modeled as

\[
\hat{a}_x = a_x + e_x \quad (4)
\]
\[
\hat{a}_y = a_y + e_y \quad (5)
\]

where \( e_x \) and \( e_y \) denotes the error of location estimation from GPS signals and modeled as independent Gaussian random variables with variance \( \omega_{\hat{a}_x}^2 \) and \( \omega_{\hat{a}_y}^2 \).

C. Modeling of the location estimates using RSS fingerprinting approach

Since the RSS fingerprints are collected at only one BS, there are ambiguities of location estimation. Then the location estimates vector can be written as \( \mathbf{s} = [s_0, s_1, \ldots, s_M] \), one of which is the true location estimate and the others are the ambiguities, \( M \) the number of ambiguities. \( s_j = (s_{jx}, s_{jy}), j \in [0, M] \).

\[
s_{0x} = u_x + n_x \quad (6)
\]
\[
s_{0y} = u_y + n_y \quad (7)
\]

\( n_x \) and \( n_y \) the error of location estimation from fingerprinting approaches and modeled as independent Gaussian random variables with variance \( \omega_{s_x}^2 \). 

III. PROPOSED ALGORITHM

In this section, localization algorithm is proposed to hybrid the estimates from TOA approaches and RSS-based fingerprinting approach.

Denotes the location estimates provided by TOA approach as \( \mathbf{G} = [\mathbf{g}_0, \mathbf{g}_1] \), one of which is the true estimate and the other one the ambiguity. \( \mathbf{g}_i = (g_{ix}, g_{iy}), i \in [0, 1] \). \( \mathbf{G} \) can be calculated by finding the intersections of two circles, one of which has center \( \mathbf{a} \) and radius \( \hat{d}_{bs} \), the other one has center \( \mathbf{u} \) and radius \( \hat{d}_{mt} \). \( \mathbf{G} \) can be calculated using the following equations with respect to \( x \) and \( y \)

\[
\begin{cases}
x^2 + y^2 = \hat{d}_{bs}^2 \quad (8)
\end{cases}
\]

Use the first equation in (14) to subtract the second one to get \( x \), then plug \( x \) into the first equation in (14) yields

\[
Ay^2 + Bx + C = 0 \quad (9)
\]

where

\[
A = 1 + \left( \frac{\hat{d}_{bs}^2}{\omega_{s_x}^2} \right), \quad B = -2D\hat{d}_{bs}^2/\omega_{s_x}^2, \quad C = D^2 - \hat{d}_{bs}^2
\]

In order to have two different solutions for (9), the following equation should be satisfied.

\[
B^2 - 4AC > 0 \quad (10)
\]

Then the two intersections are obtained as

\[
\begin{cases}
\mathbf{g}_0 = \left[ D + \frac{\hat{d}_{bs}^2 B - \sqrt{B^2 - 4AC}}{2A}, \frac{B - \sqrt{B^2 - 4AC}}{2A} \right] \\
\mathbf{g}_1 = \left[ D + \frac{\hat{d}_{bs}^2 B + \sqrt{B^2 - 4AC}}{2A}, \frac{B + \sqrt{B^2 - 4AC}}{2A} \right]
\end{cases}
\]

(11)
Estimates obtained from RSS fingerprinting approach, i.e. S
Estimates obtained from TOA approach, i.e. G
|
---

Fig. 2. Proposed Hybrid Algorithm

Otherwise, if (10) is not satisfied, the two intersections are defined as

$$g_0 = g_1 = \left( \frac{\hat{a}_x}{2}, \frac{\hat{a}_y}{2} \right)$$  (12)

The ambiguities usually introduce large errors. And the true estimates are usually closer to the true location compared with the ambiguities. Based on the above fact, the proposed hybrid algorithm is depicted in Fig. 2. Firstly, the estimate pair $g_i$ and $s_j$ with the shortest euclidean distance is found, the other pairs are treated as ambiguities and removed. Then, the final estimate is the averaged of the estimate pair. The algorithm is written as follows,

$$\left( n, m \right) = \arg\min_{i,j} \|g_i - s_j\|,$$  (13)

Then the final location estimate is $\hat{u} = (\hat{u}_x, \hat{u}_y)$, where

$$\begin{align*}
\hat{u}_x &= \frac{g_{nx} + s_{mx}}{2} \\
\hat{u}_y &= \frac{g_{ny} + s_{my}}{2}
\end{align*}$$  (14)

The disadvantage of this algorithm is that an ambiguity pair may have very small distance, in this case, the ambiguity removal fail. However, the probability of this case to happen is very small.

IV. SIMULATION RESULTS AND DISCUSSIONS

Simulations is performed to evaluate the proposed algorithm in this section. The model for simulations is depicted in Fig. 3. Without loss of generality, the line linking home BS and the located MT is set as X axis. Thus, $a_y = 0$. The location of the un-located MT is uniformly distributed within the circle with radius $R$. The ambiguities of RSS based fingerprinting approach are also uniformly distributed within the circle. For simplicity, the variances of the two TOA based distance estimation are assumed to have the same value $\sigma^2$. The NLOS error of the two distance estimates are assumed to have the same exponential p.d.f with parameter $\lambda$. Monte Carlo simulation is carried out to evaluate the location error, which is defined as $\mathbb{E}(\|\hat{u} - u\|)$ ($\mathbb{E}$ denotes expectation). The accuracy of the proposed approach is compared with the stand-alone TOA approach and RSS based fingerprinting approach. For the two stand-alone approaches, the final estimate is randomly chosen from $G$ and $S$, respectively.

Since the accuracy of these approaches depend on various parameters, including $\sigma$, $\lambda$, $\omega_{rss}$, $\omega_{gps}$, $a_x$, $r$, $M$. The effect of different parameters on accuracy are evaluated by simulations and the results are shown in Fig. 4 to 9. The parameters for simulations is shown in Tab. I. Fig. 4 to 7 shows that with different value of $\sigma$, $\lambda$, $\omega_{rss}$, $\omega_{gps}$, the proposed hybrid approach significantly outperform stand-alone TOA approach and RSS based fingerprinting approach. The reason is the proposed approach remove the ambiguities introduced by stand-alone approaches. Fig. 8 shows that when $M$ is small enough, the proposed algorithm can not offer better accuracy than the stand-alone RSS based fingerprinting approach. However, the proposed algorithm still outperform the TOA approach. The reason is that there is limited number of ambiguities from RSS-based fingerprinting approach need to be removed, however, the proposed approach still can remove the ambiguities from TOA approach. Fig. 9 shows that when the located MT is far enough from the home BS, the proposed approach outperform the two stand-alone approaches. A larger distance between the located MT and the home BS leads to better accuracy of the proposed approach.
V. CONCLUSIONS

In this paper, we consider a practical scenario that hear-ability is only possible between a MT and its home BS. In addition, located MTs can be utilized to find the location of an un-located MT without GPS receiver. In the considered scenario, only one located MT together with the home BS are utilized to find the location of the un-located MT. TOA techniques is employed to estimate the distance between the un-located MT and the home BS (or the located MT). In addition, an RSS fingerprinting database is connected to the home BS, which can estimate the location of the un-located MT if the corresponding RSS information is obtained. A hybrid cooperative localization approach was proposed to combine time-of-arrival and received signal strength based fingerprinting techniques. It was shown in simulations that the proposed hybrid approach outperform the stand-alone time-of-arrival techniques or received signal strength based fingerprinting techniques in the considered scenario. It was also found that the proposed approach offer better accuracy with larger distance between the located MT and the home BS.

VI. ACKNOWLEDGMENT

This work has been performed in the framework of the ICT project ICT- 248894 WHERE 2, which is partly funded by the European Union FP7.

REFERENCES

Fig. 7. Effect of $\omega_{gps}$ on accuracy.

Fig. 8. Effect of $M$ on accuracy.

Fig. 9. Effect of $a_x$ on accuracy.


A.12  Accuracy Limits and Mobile Terminal Selection Scheme for Cooperative Localization in Cellular Networks

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Accuracy Limits and Mobile Terminal Selection Scheme for Cooperative Localization in Cellular Networks
Ziming He, Yi Ma, and Rahim Tafazolli
Accuracy Limits and Mobile Terminal Selection
Scheme for Cooperative Localization in Cellular Networks

Ziming He, Yi Ma, and Rahim Tafazolli
Centre for Communication Systems Research
University of Surrey, Guildford, UK
e-mail: {z.he, y.ma, r.tafazolli}@surrey.ac.uk

Abstract—This paper considers cooperative localization in cellular networks. In this scenario, several located mobile terminals (MTs) are employed as reference nodes to find the location of an un-located MT. The located MTs send training sequences in the uplink, then the un-located MT performs distance estimation using received signal strength techniques. The localization accuracy of the un-located MT is characterized in terms of squared position error bound (SPEB) [1]. By taking into account the imperfect a priori location knowledge of the located MTs, the SPEB is derived in a closed-form. The closed-form indicates that the effect of the imperfect location knowledge on SPEB is equivalent to the increase of the variance of distance estimation. Moreover, based on the obtained closed-form, a MT selection scheme is proposed to decrease the number of located MTs sending training sequences, thus reduce the training overhead for localization. The simulation results show that the proposed scheme can reduce the training overhead with the paid of accuracy. And with the same training overhead, the accuracy of the proposed scheme is better than that of random selection.

I. INTRODUCTION

Cellular network based localization techniques have been extensively investigated over the past decade, e.g., [2]-[7]. Previous approaches include time-of-arrival (TOA) [2], time-difference-of-arrival (TDOA) [3] and received signal strength (RSS) techniques [4]. The major difference between TOA and TDOA approaches is: TOA approaches require tight clock synchronization between base-station (BS) and mobile terminal (MT), while TDOA approaches only require tight clock synchronization among BSs. Many of the wireless standards only mandate timing clock synchronization among BSs, the MT clock might have a drift of a few microseconds [8]. One of the advantages of RSS approach is that it does not require clock synchronization.

The training signals for location estimation can be sent either by BS through forward link (i.e. downlink) or MT through reverse link (i.e. uplink). Take TDOA-based approach as an example, most state-of-the-arts focus on downlink TDOA, such as Enhanced-Observed Time Difference (E-OTD) for Global System for Mobile Communications (GSM), Observed TDOA (OTDOA) for Universal Mobile Telecommunications System (UMTS) [3]. Since Orthogonal frequency-division multiplexing (OFDM) is a candidate for future mobile networks, downlink TDOA is investigated in OFDM cellular network in [5]. Another alternative is uplink TDOA (U-TDOA) approaches [6], which employs location measurement unit (LMU) at BSs to listen to the training signals sent by the MTs in uplink.

Cooperative localization has been extensively investigated in wireless sensor networks (WSN) [9]-[10]. Recently, the concept of cooperative localization is introduced to cellular networks [11]-[13]. The work in [11] and [13] assume that clock synchronization is perfect among BSs. Then the distances between MTs are estimated using TOA techniques. A more practical cooperative localization approach for cellular network is proposed in [12]. Their work employs located MTs to serve as reference nodes. The located MTs send training sequences to an un-located MT for location estimation. However, their work did not investigate localization accuracy of this approach. Since each located MT sends training sequences for distance estimation, the number of training sequences increases with the increases of the located MTs. The number of training sequences for localization is overhead for communications. Therefore, the overhead should be reduced, otherwise the efficiency of communication is decreased.

This paper considers a single cell scenario as depicted in Fig. 1. The home BS is responsible for collecting the distance estimates and perform location estimation. There are several located MTs, whose locations are known a priori at the home BS, and there is an un-located MT without any location knowledge. The a priori location knowledge comes from successive localization or tracking [1], and they are usually imperfect. The located MTs send training sequences for the un-located MT to perform distance estimation using RSS techniques. Then, the home BS collects the distance estimates and calculate the location of the un-located MT.

The main contributions of this paper are: 1) The localization accuracy of the un-located MT is characterized in terms of squared position error bound (SPEB) [1]. By taking into account the imperfect a priori location knowledge of the located MTs, the SPEB is derived in a closed-form. The closed-form indicate that, the effect of the imperfect knowledge on SPEB is equivalent to the increase of the variance of RSS-based distance estimation. 2) A MT selection scheme is proposed using the obtained closed-form. Utilizing the a priori location

Abstract—This paper considers cooperative localization in cellular networks. In this scenario, several located mobile terminals (MTs) are employed as reference nodes to find the location of an un-located MT. The located MTs send training sequences in the uplink, then the un-located MT performs distance estimation using received signal strength techniques. The localization accuracy of the un-located MT is characterized in terms of squared position error bound (SPEB) [1]. By taking into account the imperfect a priori location knowledge of the located MTs, the SPEB is derived in a closed-form. The closed-form indicate that, the effect of the imperfect location knowledge on SPEB is equivalent to the increase of the variance of distance estimation. Moreover, based on the obtained closed-form, a MT selection scheme is proposed to decrease the number of located MTs sending training sequences, thus reduce the training overhead for localization. The simulation results show that the proposed scheme can reduce the training overhead with the paid of accuracy. And with the same training overhead, the accuracy of the proposed scheme is better than that of random selection.

I. INTRODUCTION

Cellular network based localization techniques have been extensively investigated over the past decade, e.g., [2]-[7]. Previous approaches include time-of-arrival (TOA) [2], time-difference-of-arrival (TDOA) [3] and received signal strength (RSS) techniques [4]. The major difference between TOA and TDOA approaches is: TOA approaches require tight clock synchronization between base-station (BS) and mobile terminal (MT), while TDOA approaches only require tight clock synchronization among BSs. Many of the wireless standards only mandate timing clock synchronization among BSs, the MT clock might have a drift of a few microseconds [8]. One of the advantages of RSS approach is that it does not require clock synchronization.

The training signals for location estimation can be sent either by BS through forward link (i.e. downlink) or MT through reverse link (i.e. uplink). Take TDOA-based approach as an example, most state-of-the-arts focus on downlink TDOA, such as Enhanced-Observed Time Difference (E-OTD) for Global System for Mobile Communications (GSM), Observed TDOA (OTDOA) for Universal Mobile Telecommunications System (UMTS) [3]. Since Orthogonal frequency-division multiplexing (OFDM) is a candidate for future mobile networks, downlink TDOA is investigated in OFDM cellular network in [5]. Another alternative is uplink TDOA (U-TDOA) approaches [6], which employs location measurement unit (LMU) at BSs to listen to the training signals sent by the MTs in uplink.

Cooperative localization has been extensively investigated in wireless sensor networks (WSN) [9]-[10]. Recently, the concept of cooperative localization is introduced to cellular networks [11]-[13]. The work in [11] and [13] assume that clock synchronization is perfect among BSs. Then the distances between MTs are estimated using TOA techniques. A more practical cooperative localization approach for cellular network is proposed in [12]. Their work employs located MTs to serve as reference nodes. The located MTs send training sequences to an un-located MT for location estimation. However, their work did not investigate localization accuracy of this approach. Since each located MT sends training sequences for distance estimation, the number of training sequences increases with the increases of the located MTs. The number of training sequences for localization is overhead for communications. Therefore, the overhead should be reduced, otherwise the efficiency of communication is decreased.

This paper considers a single cell scenario as depicted in Fig. 1. The home BS is responsible for collecting the distance estimates and perform location estimation. There are several located MTs, whose locations are known a priori at the home BS, and there is an un-located MT without any location knowledge. The a priori location knowledge comes from successive localization or tracking [1], and they are usually imperfect. The located MTs send training sequences for the un-located MT to perform distance estimation using RSS techniques. Then, the home BS collects the distance estimates and calculate the location of the un-located MT.

The main contributions of this paper are: 1) The localization accuracy of the un-located MT is characterized in terms of squared position error bound (SPEB) [1]. By taking into account the imperfect a priori location knowledge of the located MTs, the SPEB is derived in a closed-form. The closed-form indicate that, the effect of the imperfect knowledge on SPEB is equivalent to the increase of the variance of RSS-based distance estimation. 2) A MT selection scheme is proposed using the obtained closed-form. Utilizing the a priori location
knowledge, the proposed scheme allows the home BS to select some of the located MTs to send training sequences. Thus, the training overhead is reduced. The simulation results show that the proposed scheme can reduce the training overhead with the paid of accuracy. And with the same training overhead, the accuracy of the proposed scheme is better than that of random selection.

II. SYSTEM DESCRIPTION AND SPEB

A. System Description

There are $N$ located MTs sending training sequences for location estimation. Each of the MTs sends only one training sequence for a single location estimation, and the training sequences sent by different located MTs are orthogonal either in time or frequency to avoid collision.

Denote $\theta_n \triangleq (x_n, y_n)$ as the 2-D true location of the $n$-th located MTs ($n \in [1, N]$), $\theta \triangleq (x, y)$ as the true location of the un-located MT, $p(\theta_1, \ldots, \theta_N)$ as the joint p.d.f of the $a$ priori locations knowledge of located MTs, where $x$ and $x_n$ denote X coordinates, and $y$ and $y_n$ denote Y coordinates. $p(\theta_1, \ldots, \theta_N)$ is known at the home BS. The considered problem can be formulated as $N + 1$ nodes cooperative localization with $a$ priori location knowledge. The parameters of interests is given by $\theta = [\theta_1, \ldots, \theta_N, \theta]^T$.

Denotes $P_r$, $P_t$, $L$ as $N \times 1$ vectors collect the RSS at the un-located MT, transmitted power of the located MTs, and path loss in dB scale, respectively. $N \times 1$ vectors e are independent Gaussian random variable representing log-normal fading. Denote $\sigma$ as the standard deviation of the element in e. In a typical case, $\sigma = 6 - 8$ (dB) [14]. Then, the following equation holds [15]

$$P_r = P_t - L + e$$

where

$$L = 10 \gamma \log_{10} d$$

$$d_n = \sqrt{(x_n - x)^2 + (y_n - y)^2}$$

$d_n$ denotes the $n$-th element in $d$.

B. CRLB Formulation

In order to derive SPEB, we will first formulate the Cramér-Rao lower bound (CRLB). The CRLB with $a$ priori knowledge reads [16, Page 84]

$$\mathbb{E}_{P_r, \theta} \left[ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right] \geq J^{-1},$$

where

$$J = J_\theta + J_P$$

$\hat{\theta}$ denotes estimates of $\theta$, $\mathbb{E}_{P_r, \theta}$ the expectation with respect to $P_r$ and $\theta$, $J$ the Fisher information matrix (FIM), $J_\theta$ the FIM from the observations [1] and has the following expression

$$J_\theta = \mathbb{E}_\theta \left[ \frac{\partial}{\partial \theta} \ln p(\theta) \left( \frac{\partial}{\partial \theta} \ln p(\theta) \right)^T \right],$$

$J_P$ the FIM from the $a$ priori knowledge [1] and has the following expression

$$J_P = \mathbb{E}_\theta \left[ \frac{\partial}{\partial \theta} \ln p(\theta) \left( \frac{\partial}{\partial \theta} \ln p(\theta) \right)^T \right],$$

$p(\theta)$ the joint p.d.f of $a$ priori knowledge of $\theta$. Since the un-located MT do not have any $a$ priori location knowledge, $p(\theta) = p(\theta_1, \ldots, \theta_N)$. According to [17], $J_\theta$ can be written as

$$J_\theta = \frac{1}{\varepsilon^2} H^T H$$

where $H = [G \ U]$ (where $G$ and $U$ are zeros, expect for the following elements)

$$G(n, 2n - 1) = \frac{\cos \phi_n}{d_n}$$

$$G(n, 2n) = \frac{\sin \phi_n}{d_n}$$

$$U(n, 1) = -\frac{\cos \phi_n}{d_n}$$

$$U(n, 2) = -\frac{\sin \phi_n}{d_n}$$

where $\phi_n$ denotes the angle from the $n$-th located MT to the un-located MT, i.e., $\phi_n = \tan^{-1} \frac{y_n}{x_n - x}$. Then, (11) can be further written as

$$J_\theta = \frac{1}{\varepsilon^2} \left[ \begin{array}{c} G^T G \\ U^T G \\ U^T U \end{array} \right]$$

Assuming $p(\theta_1, \ldots, \theta_N)$ is $2N$-order Gaussian p.d.f, $J_P$ can be written as [16, Page 85]

$$J_P = \left[ \begin{array}{cc} \Omega^{-1} & 0 \\ 0 & 0 \end{array} \right]$$
where $\Omega$ denotes the covariance matrix of the a priori location knowledge. For mutually independent a priori knowledge, $\Omega = \text{diag}\{\omega^2_1 I_2, \ldots, \omega^2_n I_2\}$, where $\omega^2_n$ denotes the variance of a priori knowledge of the $n$-th located MT. $I_2 \times 2$ identity matrix. And $\omega = [\omega_1^2, \ldots, \omega_N^2]$. The FIM can be written as

$$J = \begin{bmatrix} \frac{1}{\epsilon^2} G^T G + \Omega^{-1} & \frac{1}{\epsilon} G^T U \\ \frac{1}{\epsilon} U^T G & \frac{1}{\epsilon^2} U^T U \end{bmatrix} \Delta \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

(14)

The CRLB of location estimation of the un-located MT reads

$$F = [J^{-1}]_{2 \times 2} = \begin{pmatrix} C - B^T A^{-1} B & A^{-1} M \\ M & B^{-1} 
\end{pmatrix}^{-1}$$

(15)

where $[J^{-1}]_{2 \times 2}$ denotes the last $2 \times 2$ diagonal submatrix of $J^{-1}$.

C. Closed-form of SPEB

According to [1], the SPEB reads

$$\mathcal{P} = \text{tr} \{ F \}$$

(16)

If the a priori knowledge is perfect, $\omega = 0$, (16) reduce to

$$\mathcal{P} = \text{tr} \{ C^{-1} \}$$

(17)

which is the SPEB for single MT localization with $N$ anchors. In the following, we derive the closed-form of (16) and (17).

Matrix $A$ can be written as

$$A = \text{diag} \{ A_1, \ldots, A_N \}$$

(18)

where the sub-matrix $A_n$ ($n \in [1, N]$)

$$A_n = \begin{bmatrix} \frac{\cos^2 \phi_n + \frac{4}{\epsilon^2} \frac{\sin \phi_n \cos \phi_n}{\beta_n}}{\cos^2 \phi_n + \frac{4}{\epsilon^2} \frac{\sin^2 \phi_n}{\beta_n}} & \frac{\sin \phi_n \cos \phi_n}{\cos^2 \phi_n + \frac{4}{\epsilon^2} \frac{\sin^2 \phi_n}{\beta_n}} \\ \frac{\sin^2 \phi_n}{\cos^2 \phi_n + \frac{4}{\epsilon^2} \frac{\sin^2 \phi_n}{\beta_n}} & \frac{\sin \phi_n \cos \phi_n}{\cos^2 \phi_n + \frac{4}{\epsilon^2} \frac{\sin^2 \phi_n}{\beta_n}} \end{bmatrix}$$

(19)

Then the inverse of $A$ reads

$$A^{-1} = \text{diag} \{ A_1^{-1}, \ldots, A_N^{-1} \}$$

(20)

where

$$A_n^{-1} = \begin{bmatrix} \frac{\omega_n^4 \sin^2 \phi_n + \frac{4}{\epsilon^2} \frac{\sin \phi_n \cos \phi_n}{\beta_n}}{\beta_n} & \frac{\omega_n^2 \sin \phi_n \cos \phi_n}{\beta_n} \\ \frac{\omega_n^2 \sin \phi_n \cos \phi_n}{\beta_n} & \frac{\omega_n^4 \sin^2 \phi_n + \frac{4}{\epsilon^2} \frac{\sin \phi_n \cos \phi_n}{\beta_n}}{\beta_n} \end{bmatrix}$$

(21)

and $\beta_n = \omega_n^2 + \frac{4}{\epsilon^2} d_n^2$. Since

$$M = \frac{1}{\epsilon^2} \left[ \sum_{n=1}^{N} \frac{\omega_n^2 \cos^2 \phi_n}{d_n^2 \beta_n} - \frac{\omega_n^2 \sin \phi_n \cos \phi_n}{d_n^2 \beta_n} \right]$$

(22)

and

$$C = \frac{1}{\epsilon^2} \left[ \sum_{n=1}^{N} \frac{\omega_n^2 \cos^2 \phi_n}{d_n^2 \beta_n} - \frac{\omega_n^2 \sin \phi_n \cos \phi_n}{d_n^2 \beta_n} \right]$$

(23)

Finally, the closed-form of (16) reads

$$\mathcal{P} = \frac{1}{\mathcal{P}_N} \left( \frac{\sum_{n=1}^{N} \cos^2 \phi_n}{\beta_n} \right)$$

(25)

The closed-form of (17) is simply replacing $\beta_n$ in (25) with $\epsilon^2 d_n^2$, which is actually the CRLB of RSS-based distance estimation [18]. Therefore, it is concluded that the effect of the imperfect location knowledge on SPEB is equivalent to the increase of the variance of RSS-based distance estimation.

III. MT Selection Scheme

Utilizing the a priori knowledge $\omega$, the home BS select $S$ located MTs out of $N$ located MTs ($S \leq N$) to sent training sequences. The MT selection scheme is firstly obtained by considering a special case, then the proposed scheme is evaluated for a general case in Section IV. For this special case, $\phi_n = 2\pi(n-1)/N$, $d_1 = d_2 = \cdots = d_n = d$, the $k$-th located MT is the one the BS does not select, and the other $N - 1$ are the selected MTs, and

$$\begin{cases} \omega^2_n = \omega^2_k, & \text{when } n \neq k \\ \omega^2_n = \omega^2_k, & \text{when } n \neq k \end{cases}$$

(26)

For this special case, the following equations holds

$$\begin{align*} \sum_{n=1}^{N} \cos^2 \phi_n &= \frac{N-1}{N} \quad (N \geq 3) \\ \sum_{n=1}^{N} \sin^2 \phi_n &= \frac{N}{2} \quad (N \geq 3) \\ \sum_{n=1}^{N} \cos \phi_n \sin \phi_n &= 0 \\
\end{align*}$$

(27)

Applying (27) to (25), the SPEB without MT selection (i.e. $\mathcal{P}_N$) and SPEB with the $N - 1$ selected MTs (i.e. $\mathcal{P}_{N-1}$) reads

$$\mathcal{P}_N = \frac{N-1 + \frac{1}{\beta_k}}{N \beta_k + N / 25 \beta_k}$$

(28)

$$\mathcal{P}_{N-1} = \frac{4 \beta (N-1)}{N^2 - N}$$

(29)

where $\beta = \omega^2 + \epsilon^2 d^2$, $\beta_k = \omega^2_k + \epsilon^2 d^2$. It is found that

$$\mathcal{P}_N < \mathcal{P}_{N-1}$$

(30)

and

$$\lim_{\omega^2 \rightarrow \infty} \mathcal{P}_N = \mathcal{P}_{N-1}$$

(31)

(30) indicate that with MT selection, the training overhead decreases by 1 with the paid of accuracy. (31) indicate that there is no accuracy degradation when $\omega^2$ is infinity. Let the accuracy degradation satisfying the following equation

$$\mathcal{P}_N \geq \eta \mathcal{P}_{N-1}$$

(32)

where $\eta \in [0, 1)$. Then, (32) can be rewritten as

$$\frac{\beta}{\beta_k} \geq \rho(N) \triangleq \frac{2\eta(N-1) - (N-2)}{(N-1)(N-2)(1-\eta)}$$

(33)

If $\beta_k$ satisfying (33), the $k$-th MT can be excluded with required accuracy degradation. $\eta$ can be set large enough for
TABLE I
MT SELECTION SCHEME

<table>
<thead>
<tr>
<th>MT selection algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $\omega = \text{sort}(\omega)$</td>
</tr>
<tr>
<td>2: for $S = N : 4$</td>
</tr>
<tr>
<td>3: $\omega_2 = \omega(S)$, $\omega_2 = \frac{\sum_{s=1}^{S-1} \omega_s}{S-1}$</td>
</tr>
<tr>
<td>4: if $\omega_2^2 &lt; \rho(S)$</td>
</tr>
<tr>
<td>5: break;</td>
</tr>
<tr>
<td>6: end</td>
</tr>
<tr>
<td>7: end</td>
</tr>
<tr>
<td>8: The MTs with a priori knowledge $\omega_1, \ldots, \omega_S$ are the selected MTs</td>
</tr>
</tbody>
</table>

The numerical results in Fig. 2 shows $\rho(N)$ as a function of $\eta$ and $N$. It is observed from the upper plot that $\rho(N)$ decreases with $N$ increases. This means that with fixed $\omega$, a larger $N$ leads to a smaller minimum requirement of $\omega_2^2$ in order to exclude the $k$-th located MT. It is observed from the lower plot that $\rho(N)$ increases with $\eta$ increases. This means that with fixed $\omega$, a larger $\eta$ leads to a larger minimum requirement of $\omega_2^2$ in order to exclude the $k$-th located MT.

The simulation results in Fig. 3 shows the effect of $N$ and $\eta$ on SPEB and training overhead. Fig. 4 shows the effect of $\eta$ on SPEB and training overhead. The parameters for simulations are shown in Fig. 3 and Fig. 4. For each realization, $N$ located MTs and one un-located MT are generated uniformly within the circle with radius $R$, as shown in Fig. 1. Assuming the elements in $\omega$ subject to independent Rayleigh distribution. The variance of the Rayleigh random variables is $\frac{(4-\pi)10^5}{2}$ (m$^2$). For each realization, the SPEB are calculated for the scheme without MT selection, with proposed MT selection, and with random MT selection, respectively. For the scheme without MT selection, all the $N$ located MTs are used to calculate the SPEB. For the proposed MT selection scheme, $S$ located MTs are selected according to the algorithm in Tab. I. Then, the SPEB is calculated using (25). For random MT selection scheme, $S$ located MTs are selected randomly from the $N$ MTs to calculate the SPEB. Thus, the training overhead is the same for the proposed scheme and random selection scheme. For performance comparison, the SPEB and training overhead is averaged over 10000 realizations.

The SPEB curves without selection in Fig. 3 shows that under the condition of imperfect location knowledge, the accuracy increases with the increases of the number of involved located MTs. It is observed from the upper and lower plot of Fig. 3 that, compared with the scheme without MT selection, the proposed MT selection scheme can reduce training overhead with the paid of accuracy in terms of SPEB. And the proposed MT selection offer better accuracy than random MT selection with the same training overhead.

In addition, Fig. 3 shows that for larger $N$, the proposed scheme can reduce more training overhead. This is because for larger $N$, the minimum requirement of $\omega_2^2$ is smaller, as discussed previously. Thus, more located MTs satisfy the requirement in (33). Fig. 4 shows that for larger $\eta$, the proposed scheme reduce less training overhead. This is because for larger $\eta$, the minimum requirement of $\omega_2^2$ is larger. Thus, less located MTs satisfy the requirement in (33).

IV. Simulation Results and Discussion

As shown in Tab. I, the proposed MT selection scheme depend on the threshold $\rho(N)$, which is related to $N$ and $\eta$. an acceptable accuracy degradation. At this stage, the problem we want to address is how many and which located MTs need to be selected for a general case. Since the variance of the distance estimation is not known a priori, we simply set $\varepsilon^2d^2 = 0$, then (33) can be further written as

$$
\frac{\omega_2^2}{\omega^2} \geq \rho(N)
$$

(34)

$\omega^2\rho(N)$ is the minimum requirement of $\omega_2^2$ in order to exclude the $k$-th MT. According to the requirement in (34), the proposed MT selection scheme is presented in Tab. I. sort(·) denotes the MATLAB function, which sorts the elements of vector in ascending order. After MT selection using the algorithm in Tab. I, the training overhead is decreased by $N - S$. Since the proposed scheme is obtained from a special case, the accuracy degradation no longer satisfy (32). The accuracy degradation is evaluated using simulations in Section IV.

TABLE II
PARAMETERS FOR SIMULATIONS

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$</td>
<td>5 (m)</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>6 (dB)</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>4</td>
</tr>
<tr>
<td>$R$</td>
<td>100 (m)</td>
</tr>
</tbody>
</table>

Fig. 2. $\rho(N)$ as a function of $\eta$ and $N$
This paper considered cooperative localization in cellular networks. In this scenario, several located MTs are employed to find the location of an un-located MT. The located MTs sent training sequences in the uplink, then the un-located MT perform distance estimation using received signal strength techniques. The localization accuracy of the un-located MT is characterized in terms of SPEB. By taking into account imperfect a priori location knowledge of the located MTs, the SPEB was derived in a closed-form. The closed-form indicated that the effect of the imperfect knowledge on SPEB is equivalent to the increase of the variance of RSS-based distance estimation. Moreover, based on the obtained closed-form, a MT selection scheme was proposed to decrease the number of located MTs sending training sequences, thus reduce the training overhead for localization. The proposed scheme can adaptively select the located MTs according to the a priori knowledge and the number of located MTs. The simulation results shown that the proposed scheme can reduce the training overhead with the paid of accuracy. And with the same training overhead, the accuracy of the proposed scheme is better than that of random selection.

VI. ACKNOWLEDGEMENT

This work has been performed in the framework of the ICT project ICT- 248894 WHERE 2, which is partly funded by the European Union FP7.

REFERENCES

A.13 Cooperative Localization in a Distributed Base Station Scenario

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Cooperative Localization in a Distributed Base Station Scenario
Ziming He, Yi Ma, and Rahim Tafazolli
Cooperative Localization in a Distributed Base Station Scenario

Ziming He, Yi Ma, and Rahim Tafazolli
Centre for Communication Systems Research
University of Surrey, Guildford, UK
e-mail: {z.he, y.ma, r.tafazolli}@surrey.ac.uk

Abstract—Previous work about cooperative localization in cellular networks usually consider a centralized processor (CP) is available for location estimation. This paper consider cooperative localization in a distributed base station (BS) scenario, where there is no CP, and the distributed BSs are responsible for location estimation. In this scenario, Global Positioning System (GPS) enable mobile terminals (MTs), i.e., located MTs, are employed as reference nodes. Then, several located MTs can help to find the locations of an un-located MT, by estimating the distance between the un-located MT using received signal strength techniques. Two localization approaches are proposed, the first approach requires only one BS to collect all the assistance information for localization and estimate the location. The second approach distribute the location estimation task to several BSs. The communication overhead between distributed BSs are investigated for these two approaches. Moreover, by taking into account the effect of imperfect location knowledge of the located MTs, the accuracy limits of both approaches are derived. The simulation results shows that compared with the first approach, the second approach can reduce the communication overhead between distributed BSs with the paid of accuracy.

I. INTRODUCTION

Cellular network based localization techniques have been extensively investigated over the past decade, e.g., [1]-[6]. Previous approaches includes time-of-arrival (TOA) [1], time-difference-of-arrival (TDOA) [2] and received signal strength (RSS) techniques [3]. The major difference between TOA and TDOA approaches is: TOA approaches require tight clock synchronization between base-station (BS) and mobile terminal (MT), while TDOA approaches only require tight clock synchronization among BSs. Many of the wireless standards only mandate timing clock synchronization among BSs, the MT clock might have a drift of a few microseconds [7]. One of the advantages of RSS approach is that it does not require clock synchronization.

Cooperative localization has been extensively investigated in wireless sensor networks (WSN) [8]-[9]. Recently, the concept of cooperative localization is introduced to cellular networks [10]-[12]. The work in [10] and [12] assume that different MTs can directly communicate with each other in a peer-to-peer (P2P) manner, and clock synchronization is perfect between MTs. Then the distances between MTs is estimated using TOA techniques. Although direct communication between wireless nodes is usually available in WSN, it is not true for cellular networks. Actually, MTs never communicate directly with each other in cellular networks [11]. A more practical cooperative localization approach for cellular network is proposed in [11]. Their work employs Global Positioning System (GPS) enable MTs (i.e., located MTs) to serve as reference nodes, then the located MTs can help to find the location of an un-located MT. All the assistance information required for location estimation can be forwarded to a centralized processor (e.g. serving radio network controller). In this case, P2P communications are not assumed between MTs. In addition, the distances between located MTs and un-located MTs are estimated using RSS techniques, thus clock synchronization is not required between MTs. Note that the assistance information includes the location information of the located MTs and RSS-based distance estimates.

From the communication architecture viewpoint, BSs with centralized processor (CP) has several drawbacks. As a network expands, it is expensive to require each new BS to be connected directly to the CP [13]. Therefore, it is interested to employ distributed BSs, which communicates directly with their neighbors [13]. In distributed BS scenarios, where a CP is not available, the location estimation can be performed at the distributed BSs. It is of interest to know which BS (or BSs) are going to perform location estimation and how much assistance information is transmitted between BSs. Note that the assistance information for location estimation is overhead for communications. This overhead should be reduced, otherwise it will decrease the efficiency of communication systems.

This paper consider a distributed BS scenario, as depicted in Fig. 1. The serving BS of an un-located MT is the BS in the center, and the serving BSs of several located MTs are the other surrounding BSs. The un-located MT communicates with its home BS on its own uplink frequency, the located MTs listen to this frequency and perform RSS-based distance estimation. Two localization approaches are proposed for the considered scenario. For the first approach, all the assistance information is forwarded to the central BS for location estimation. For the second approach, initial location estimation are performed at the surrounding BSs, receptively. Then the initial estimates are forwarded to the central BS to obtain the final estimate of the un-located MT. Compared with the first approach, the second approach reduce the communication overhead among distributed BSs. However, the second approach may lead to the loss of localization accuracy, due to the fact that the location estimation are distributed among BSs.
The major contributions of this paper are: 1) By taking into account the effect of imperfect location knowledge of the located MTs, we derive the accuracy limits of RSS-based cooperative localization in terms of squared position error bound (SPEB) [14]. The derived SPEB can be used to benchmark the accuracy of the approach in [11]. 2) We consider cooperative localization in a distributed BS scenario, which have not been investigated before. Two approaches are proposed for this scenario. Based on 1), the accuracy and communication overhead of these two approaches are compared. Simulation results show that compared with the first approach, the second approach can reduce the communication overhead between distributed BSs with the paid of accuracy.

II. SYSTEM DESCRIPTION AND COMMUNICATION OVERHEAD

A. System Description

The system model is depicted in Fig. 1, a seven cells model is considered. The MTs communicate directly with their home BSs, and BSs communicate directly with their neighboring BSs. BSi is the home BS of the cell with identification (ID) i, i ∈ [0, 6]. The un-located MT belongs to BS0. The considered located MTs are distributed within any G surrounding cells (G ∈ [1, 6]), each of which has M (M ≥ 3) located MTs. The total number of located MTs is N = MG. The un-located MT communicates with its home BS on its uplink frequency, which is known by the located MTs. The located MTs listen to this frequency and perform RSS-based distance estimation. Location estimation is performed at BSs after the distance estimates are obtained at the located MTs. Assuming the un-located MT require the location information of itself, the location estimate has to be sent to the un-located MT by BS0 via downlink channel.

B. Communication Overhead

In order to estimate the location of the un-located MT, assistance information is required. The assistance information for localization includes RSS-based distance estimates and a priori locations knowledge of the located MTs obtained from GPS. Since the a priori locations knowledge is imperfect, it can be treated as random variables with a given probability density function (p.d.f). For simplicity, it is assumed that the a priori locations knowledge of the located MTs are independent Gaussian random variables with the same variance ω2. Then the covariance matrix of the a priori knowledge is a 2N × 2N diagonal matrix Ω with identical diagonal entries ω2. ω is the accuracy of GPS and have a typical value of 5–10 (m) [15]. It is reasonable to assume all the BSs knows ω, the located MTs only need to forward the mean of the a priori knowledge to BSs.

For approach 1, all the assistance information is firstly sent to the surrounding BSs by the located MTs. Then, the surrounding BSs forward it to BS0, which estimate the location and then sent the estimate to the un-located MT via downlink. The communication overhead to forward distance estimates is 32N (bits) (2-D location requires two float). Thus, the total overhead between BSs is 96N bits for a single location estimation.

For approach 2, when each surrounding BS obtain the assistance information, initial location estimation of the un-located can be carried out, since M ≥ 3. Denote ξg ≜ (ug, vg) as the initial location estimate obtained by the g-th (g ∈ [1, G]) surrounding BS, α2g and β2g as the variance of ug and vg, respectively. Assuming α2g and β2g can also be obtained at the g-th BS, ξg, α2g and β2g are forwarded to BS0. Then approach 2 creates overhead of 32 × 4G = 128G (bits). At the BS0, the final location estimate ̂θ can be written as a weighting combination of ξg

\[ ̂θ = (u^T W_u + v^T W_v) \]

where

\[ W_u = \frac{[\alpha_1^2, \ldots, \alpha_G^2]^T}{\sum_{g=1}^{G} \alpha_g^2} \]
\[ W_v = \frac{[\beta_1^2, \ldots, \beta_G^2]^T}{\sum_{g=1}^{G} \beta_g^2} \]
\[ u = [u_1, \ldots, u_G]^T, \quad v = [v_1, \ldots, v_G]^T. \]

A summary of communication overhead of these two approaches is shown in Tab I. For a fixed N, the maximum overhead of approach 2 is 128N/3 (bits) (when M = 3), the minimum overhead is 128 (bits) (when G = 1). Thus, for a single location estimation, approach 2 can reduce at least 160N/3 (bits) and at most 96N − 128 (bits) compared with approach 1. Obviously, more overhead can be reduced for a larger N.

If the un-located MT and located MTs have nearly the same mobility, the located MTs need to continue forwarding their updated locations (i.e. mean of the a priori knowledge) to
TABLE I
COMMUNICATION OVERHEAD BETWEEN BSs

<table>
<thead>
<tr>
<th>Approach</th>
<th>Overhead (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>approach 1</td>
<td>96N</td>
</tr>
<tr>
<td>approach 2</td>
<td>128G</td>
</tr>
</tbody>
</table>

their serving BSs for location update of the un-located MT. Then the surrounding BSs employ approach 1 or 2 to forward data to BS0. The overhead is large for successive localization or tracking, thus the overhead reduction using approach 2 is meaningful.

III. LOCALIZATION ACCURACY

The localization accuracy of two approaches is analyzed in this section. The accuracy is evaluated in terms of the SPEB defined in [14]. We first derive the SPEB of the approach 1, then the SPEB of the approach 2 can be easily obtained. Note that the SPEB of the approach 1 can be utilized to benchmark the accuracy of the approach in [11], which utilize CP to calculate the location of the un-located MT.

A. SPEB of the approach 1

Denote \( \theta_n \triangleq (x_n, y_n) \) as the 2-D true location of the \( n \)-th located MTs \( (n \in [1, N]) \), \( \theta \triangleq (x, y) \) as the true location of the un-located MT, \( p(\theta_1, \ldots, \theta_N) \) as the joint p.d.f of the \( a \) priori locations knowledge of located MTs, where \( x \) and \( x_n \) denote X coordinates, and \( y \) and \( y_n \) denote Y coordinates. The considered problem can be formulated as \( N + 1 \) nodes cooperative localization with \( a \) priori location knowledge. The parameters of interests is given by \( \theta = [\theta_1, \ldots, \theta_N, \theta]^T \).

Denotes \( P_r, P_t, L \) as \( N \times 1 \) vectors collect the RSS at located MTs, transmitted power of the un-located MT, and path loss in dB scale, respectively. \( N \times 1 \) vectors \( e \) denote X and Y elements are independent Gaussian random variable representing log-normal fading. Denote \( \sigma \) as the standard deviation of the element in \( e \). In a typical case, \( \sigma = 6 - 8 \) (dB) [16]. Note that all the entries in \( P_t \) are identical since the un-located MT broadcast training sequence. Then, the following equation holds [17]

\[
P_r = P_t - L + e
\]  

(4)

where

\[
L = 10^{ \gamma } \log_{10} \alpha
\]  

(5)

\[
d_n = \sqrt{(x_n - x)^2 + (y_n - y)^2}
\]  

(6)

d_n denotes the \( n \)-th element in \( d \).

The Cramér-Rao lower bound (CRLB) with \( a \) priori knowledge reads [18, Page 84]

\[
\mathbb{E}_{P_r, \theta} \left[ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right] \geq J^{-1},
\]  

(7)

where

\[
J = J_{\theta} + J_P
\]  

(8)

\( \hat{\theta} \) denotes estimates of \( \theta \), \( \mathbb{E}_{P_r, \theta} \) the expectation with respect to \( P_r \) and \( \theta \), \( J \) the Fisher information matrix (FIM), \( J_{\theta} \) the FIM from the observations [14] and has the following expression

\[
J_{\theta} = \mathbb{E}_d \left[ \frac{\partial}{\partial \theta} \ln p(P_r, \theta) \left( \frac{\partial}{\partial \theta} \ln p(P_r, \theta) \right)^T \right],
\]  

(9)

\( J_P \) the FIM from the \( a \) priori knowledge [14] and has the following expression

\[
J_P = \mathbb{E}_\theta \left[ \frac{\partial}{\partial \theta} \ln p(\theta) \left( \frac{\partial}{\partial \theta} \ln p(\theta) \right)^T \right],
\]  

(10)

\( p(\theta) \) the \( a \) priori joint p.d.f of \( \theta \), \( p(\theta) = p(\theta_1, \ldots, \theta_N) \), since the un-located MT do not have any \( a \) priori location knowledge. According to [19], \( J_{\theta} \) can be written as

\[
J_{\theta} = \frac{1}{\varepsilon^2} H^T H
\]  

(11)

where \( H = \nabla^T \otimes \mathbf{d} \) (\( \otimes \) denotes the Kronecker product, \( \nabla \triangleq \left[ \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \cdots, \frac{\partial}{\partial y}, \frac{\partial}{\partial x} \right]^T \), \( \varepsilon = (\sigma \ln 10)/(10\gamma) \), \( \gamma \) the path loss factor, \( \gamma = 2 \) for free space, \( \gamma = 4 \) is often used to characterize the path loss in urban areas [16]. \( H \) can be further expressed as

\[
H = \left[ \begin{array}{c} \mathbf{G} \\ \mathbf{U} \end{array} \right]
\]  

(12)

where the elements of \( \mathbf{G} \) and \( \mathbf{U} \) are zeros, expect for the following elements

\[
\begin{align*}
\mathbf{G}(n, 2n - 1) &= \cos \phi_n \\
\mathbf{G}(n, 2n) &= \sin \phi_n \\
\mathbf{U}(n, 1) &= -\cos \phi_n \\
\mathbf{U}(n, 2) &= -\sin \phi_n
\end{align*}
\]  

(13)

(14)

where \( \phi_n \) denotes the angle from the \( n \)-th located MT to the un-located MT, i.e., \( \phi_n = \tan^{-1} \frac{y - y_n}{x - x_n} \). Then (11) can be further written as

\[
J_{\theta} = \frac{1}{\varepsilon^2} \left[ \begin{array}{cc} \mathbf{G}^T \mathbf{G} & \mathbf{G}^T \mathbf{U} \\ \mathbf{U}^T \mathbf{G} & \mathbf{U}^T \mathbf{U} \end{array} \right]
\]  

(15)

For \( 2N \)-order Gaussian p.d.f \( p(\theta_1, \ldots, \theta_N) \), \( J_P \) can be written as [18, Page 85]

\[
J_P = \left[ \begin{array}{cc} \Omega^{-1} & 0 \\ 0 & 0 \end{array} \right]
\]  

(16)

Then

\[
J = \left[ \begin{array}{cc} \frac{1}{\varepsilon^2} \mathbf{G}^T \mathbf{G} & \Omega^{-1} \\ \frac{1}{\varepsilon^2} \mathbf{U}^T \mathbf{G} & \frac{1}{\varepsilon^2} \mathbf{U}^T \mathbf{U} \end{array} \right]
\]  

\( \triangleq \left[ \begin{array}{cc} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{array} \right] \) (17)

The CRLB of the estimate using approach 1 reads

\[
\mathbf{F} = [J^{-1}]_{2 \times 2} = (\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B})^{-1}
\]  

(18)

where \([J^{-1}]_{2 \times 2}\) denotes the last \( 2 \times 2 \) diagonal submatrix of \( J^{-1} \). According to [14], the SPEB reads

\[
\mathcal{P} = \text{tr} \{ \mathbf{F} \}
\]  

(19)
If the location estimation from GPS is perfect, then \( \omega = 0 \), (19) reduce to
\[
P = \text{tr} \{ C^{-1} \} \tag{20}
\]
which is the SPEB for single MT localization with \( N \) anchors.

**B. SPEB of approach 2**

Denote the CRLB matrix for estimation of the un-located MT as \( \mathbf{F}_g \), which can be calculated using (18), then \( \frac{\sigma^2}{\omega} = \mathbf{F}_g(1,1) \), \( \beta_G^2 = \mathbf{F}_g(2,2) \). Assuming \( \xi_q \) from different surrounding BSs are mutually independent, then the SPEB of the estimate using this approach reads
\[
P = \mathbf{W}_u^T \mathbf{A}_\alpha \mathbf{W}_u + \mathbf{W}_v^T \mathbf{A}_\beta \mathbf{W}_v \tag{21}
\]
where \( \mathbf{A}_\alpha = \text{diag} \{ \alpha_1^2, \ldots, \alpha_G^2 \} \), \( \mathbf{A}_\beta = \text{diag} \{ \beta_1^2, \ldots, \beta_G^2 \} \).

**IV. SIMULATION RESULTS AND DISCUSSION**

In this section, the simulation results investigate the accuracy of approach 1 and 2 in terms of SPEB. The parameters for simulations are shown in Tab. II. The un-located MTs and located MTs are randomly generated with 10000 realizations. For each realization, the SPEB is calculated using (19), then the empirical cumulative distribution function (c.d.f) of SPEB is plotted. For each realization, the location of the un-located MT is \( \mathbf{h} = (\text{real}(\rho_0 e^{j\phi_0}), \text{imag}(\rho_0 e^{j\phi_0})) \), the location of the \( n \)-th located MT is \( \mathbf{h}_n = (\text{real}(\rho_0 e^{j\phi_0}), \text{imag}(\rho_0 e^{j\phi_0})) \). \( \rho_0 \) and \( \rho_n \) are uniformly distributed within \([0, r]\) and \([r, R]\), respectively. \( \phi_0 \) is uniformly distributed within \([0, 2\pi]\). If the cell ID of the \( n \)-th located MT is \( i (i \in [1, 6]) \), \( \phi_0 \) is uniformly distributed within \([\pi(i - 1)/3 - \pi/6, \pi(i - 1)/3 + \pi/6]\).

Fig. 2 investigate the effect of the cell IDs of the located MTs. For the upper plot of Fig. 2, \( N = 24, G = 2 \), then \( M = 12 \). It is observed that with cell IDs 1 and 4, the approaches have the best accuracy. The worse appears with cell IDs 1 and 2. The reason is cell 1 and 4 has the largest separated distance and cell IDs 1 and 2 has the smallest. The similar result is observed from the lower plot of Fig. 2, where \( N = 24, G = 3, M = 8 \). The best accuracy appears with cell IDs 1, 3 and 5 and the worse appear with cell IDs 1, 2 and 3.

Fig. 3 investigate the effect of \( M \) on SPEB with fixed \( G \). For the upper plot, \( G = 2 \) with cell IDs 1 and 4, and for the upper plot, \( G = 3 \) with cell IDs 1, 3 and 5. It is observed that for a fixed \( G \), the increase of \( M \) leads to the increase of the accuracy, since the total number of located MTs \( N \) increases. It is also observed from Tab. I that for a fixed \( G \), the overhead for approach 2 is fixed, but the overhead for approach 1 increase with \( M \) increases. This indicate that for a fixed \( G \), approach 2 can reduce more overhead with a larger \( M \).

Fig. 4 investigate the effect of \( G \) on SPEB with fixed \( N \). For fair comparison, the cell IDs are selected in order to achieve the best accuracy, according to the observation in Fig. 2. For \( G = 2 \), the cell IDs are 1 and 4; for \( G = 3 \), the cell IDs are 1, 3 and 5; for \( G = 4 \), the cell IDs are 1, 2, 4 and 5; for \( G = 5 \), the cell IDs are 1, 2, 3, 4 and 5; for \( G = 6 \), the cell IDs are 1, 2, 3, 4, 5 and 6. The effect of \( G \) is evaluated with \( N = 18, 24, 30, 36 \), respectively. It is observed from the four subplots that with \( G = 2 \), approach 1 offer slightly worse accuracy than the accuracy with \( G = 3, 4, 5, 6 \). However, approach 2 offer the best accuracy with \( G = 2 \). Thus, the accuracy degradation of approach 2 compared to approach 1 is the smallest with \( G = 2 \). It is also observed from Tab. I that with a fixed \( N \), approach 2 can reduce more overhead for a smaller \( G \). For the case \( G = 1 \), the two approaches offer the same accuracy, but approach 2 can reduce the overhead by \( 96N - 128 \) (bits).

It is concluded from Fig. 3 and Fig. 4 that approach 1 usually offer better accuracy than approach 2, but approach 2 outperform approach 1 in terms of communication overhead between BSs. Thus, approach 2 can reduce the overhead the with the paid of accuracy.

**V. CONCLUSIONS**

This paper considered cooperative localization in a distributed BS scenario, where there is no CP, and the distributed BSs are responsible for location estimation. In this scenario, the located MTs are employed as reference nodes. Then,
several located MTs can help to find the locations of an unlocated MT; by estimating the distance between the un-located MT using RSS techniques. We proposed two localization approaches; approach 1 requires only one BS to collect all the assistence information for localization and estimate the location. Approach 2 distribute the location estimation task to several BSs. Moreover, by taking into account the effect of imperfect location knowledge of the located MTs, the accuracy limits of both approaches are derived. The simulation results shown that compared with approach 1, approach 2 can reduce the communication overhead between distributed BSs with the paid of accuracy.

VI. ACKNOWLEDGEMENT

This work has been performed in the framework of the ICT project ICT-248894 WHERE 2, which is partly funded by the European Union FP7.

REFERENCES


A.14 Sensor Localization using Nonparametric Generalized Belief Propagation in Network with Loops

©2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Sensor Localization using Nonparametric Generalized Belief Propagation in Network with Loops
Vladimir Savic and Santiago Zazo
Sensor Localization using Nonparametric Generalized Belief Propagation in Network with Loops

Vladimir Savic
Signal Processing Application Group
Polytechnic University of Madrid
Ciudad Universitaria S/N, 28040 Madrid, vladimir@gaps.ssr.upm.es

Santiago Zazo
Signal Processing Application Group
Polytechnic University of Madrid
Ciudad Universitaria S/N, 28040 santiago@gaps.ssr.upm.es

Abstract - Belief propagation (BP) is one of the best-known graphical model for inference in statistical physics, artificial intelligence, computer vision, etc. Furthermore, a recent research in distributed sensor network localization showed us that BP is an efficient way to obtain sensor location as well as appropriate uncertainty. However, BP convergence is not guaranteed in a network with loops. In this paper, we propose localization using generalized belief propagation based on junction tree method (GBP-JT) and nonparametric (particle-based) approximation of this algorithm (NGBP-JT). We illustrate it in a network with loop where BP shows poor performance. In fact, we compared estimated locations with Nonparametric Belief Propagation (NBP) algorithm. According to our simulation results, GBP-JT resolved the problems with loops, but the price for this is unacceptable large computational cost. Therefore, our approximated version of this algorithm, NGBP-JT, reduced significantly this cost, with little effect on accuracy.

Keywords: Localization, generalized belief propagation, junction tree, particle filters, loops.

1 Introduction

The localization consists in obtaining the relative or absolute position of a sensor node together with the uncertainty of its estimate. Equipping every sensor with a GPS receiver or equivalent technology may be expensive, energy prohibitive and limited to outdoor applications. Therefore, we consider the problem in which some small number of sensors, called anchor nodes, obtain their coordinates via GPS or by installing them at points with known coordinates, and the rest, unknown nodes, must determine their own coordinates. If unknown nodes were capable of high-power transmission, they would be able to make measurements with all anchor nodes (single-hop technique). However, we prefer to use energy-conserving devices without power amplifier, with lack the energy necessary for long-range communication. In this, multi-hop, case, each sensor has available noisy measurements only to several neighboring sensors.

A recent direction of research in distributed sensor network localization is the use of particle filters [1, 2]. In [3], Ihler et al. formulated the sensor network localization problem as an inference problem on a graphical model and applied particle based variant of belief propagation (BP) methods [4], the so-called nonparametric belief propagation (NBP) algorithm, to obtain an approximate solution to the sensor locations. Comparing with deterministic algorithms [5, 6, 7], the main advantages of this statistical approach are its easy implementation in a distributed fashion and sufficiency of a small number of iterations to converge. Furthermore, NBP is capable of providing information about location estimation uncertainties and accommodating non-Gaussian distance measurement errors. However, NBP convergence is not guaranteed in a network with loops [4, 8], or even if NBP converges, it could provide us less accurate estimates.

Therefore, in this paper, we present a new variant of the NBP method which solves problem with loops. We propose localization using generalized belief propagation based on junction tree (GBP-JT) and nonparametric (particle based) approximation of this algorithm (NGBP-JT). Junction tree model is a generalization of belief propagation (BP) that is correct for arbitrary graphs. Jordan proved it using elimination procedure [9]. Comparing with Ihler’s Nonparametric Belief Propagation (NBP) algorithm, GBP-JT converges well in network with loops, but the price for this is unacceptable large computational cost. Therefore, we implemented approximated version of this algorithm, NGBP-JT, by drawing high-dimensional particles from appropriate cliques in the network. Moreover, in order to draw samples in high-probabilistic area, we used improved sampling procedure which utilizes information from first phase of the algorithm as an a priori. This version reduces significantly computational cost (over a 100 times), with little effect on accuracy.

The remainder of this paper is organized as follows. In Section 2, we review standard BP and condition for its convergence. In Sections 3 and 4, we propose GBP-JT and NGBP-JT algorithms, respectively. Simulation results are presented in Section 5. Finally, Section 6 provides some conclusions and future work perspective.
2 Convergence of Belief Propagation

In the standard BP algorithm, the belief at a node $i$ is proportional to the product of the local evidence at that node $(\psi_i(x_i))$, and all the messages coming into node $i$:

$$M_i(x_i) = k\psi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i)$$

(1)

where $k$ is a normalization constant and $N(i)$ denotes the neighbors of node $i$. The messages are determined by the message update rules:

$$m_{ji}(x_i) = \sum_{x_j} \psi_j(x_j)\psi_{ij}(x_i, x_j) \prod_{k \in N(j) \backslash i} m_{kj}(x_j)$$

(2)

where $\psi_{ij}(x_i, x_j)$ is pairwise potential between nodes $i$ and $j$. On the right-hand side, there is a product over all messages going into node $j$ except for the one coming from node $i$.

In practical computation, one starts with nodes at the edge of the graph, and only computes a message when one has available all the messages required. It is easy to see [4] that each message needs to be computed only once for single connected graphs. That means that whole computation takes a time proportional to the number of links in the graph, which is dramatically less that the exponentially large time that would be required to compute marginal probabilities naively. In other words, BP is a way of organizing the "global" computation of marginal beliefs in terms of smaller local computations.

The BP algorithm, defined by equations (1) and (2), does not make a reference to the topology of the graph that it is running on. Thus, there is nothing to stop us from implementing it on a graph that has loops. One starts with some initial set of messages, and simply iterates the message-update rules (2) until they eventually converge, and then can read off the approximate beliefs from the belief equations (1). But if we ignore the existence of loops and permit the nodes to continue communicating with each other, messages may circulate indefinitely around these loops, and the process may not converge to a stable equilibrium. One can indeed find examples of graphical models with loops, where, for certain parameter values, the BP algorithm fails to converge or predicts beliefs that are inaccurate. On the other hand, the BP algorithm could be successful in graphs with loops, e.g. error-correcting codes defined on Tanner graphs that have loops [10].

This can be proved using Bethe’s approximation to the "free energy" [4, 8]. The fixed points of the BP algorithm correspond to the stationary points of the Bethe "free energy". To make this more clear, let’s define for one graphical model, a joint probability function $p(x)$, we can define a "distance" between $p(x)$ and $b(x)$, called Kullback-Leibler (KL) distance, by:

$$D(b(x) \parallel p(x)) = \sum_{x} b(x)\ln \frac{b(x)}{p(x)}$$

(3)

The KL distance is useful because it is always non-negative and is zero if and only if the two probability functions $p(x)$ and $b(x)$ are equal.

Statistical physicists generally assume that Boltzmann’s law is true:

$$p(x) = \frac{1}{Z} e^{-E(x)/T}$$

(4)

where $Z$ is a normalization constant, and the "temperature" $T$ is just a parameter that defines a scale of units for the "energy" $E$. For simplicity, we can choose $T = 1$. Using eqs. (3) and (4), we find the KL distance:

$$D(b(x) \parallel p(x)) = \sum_{x} b(x)E(x) + \sum_{x} b(x)\ln b(x) + \ln Z$$

(5)

So we see that this KL distance will be zero when approximate probability function $b(x)$ will equal to the exact probability function $p(x)$. The Bethe approximation is the case when joint belief $b(x)$ is function of single-node beliefs $b(x_i)$ and two-node beliefs $b(x_i, x_j)$. Yedidia et al. proved [4] that for a single-connected graph, values of these beliefs that minimize the Bethe free energy, will correspond to the exact marginal probabilities. For graph with loops, these beliefs will only be approximations, although a lot of them are quite good.

3 Localization using Generalized Belief Propagation

Our goal in this section is to develop new localization algorithm using generalized belief propagation based on junction tree method (GBP-JT). Junction tree algorithm is a standard method for exact inference in graphical model [9]. The graph is first triangulated (added "virtual" edges so that every loop of length more than 3 has a chord). Given a triangulated graph, with cliques $C_i$ and potentials $\psi_C(x_C)$, and given corresponding junction tree which defines links between the cliques, we send the following message from clique $C_i$ to clique $C_j$ by the message update rule:

$$m_{ij}(x_{Sij}) = \sum_{C \in N(i) \cup j} \psi_C(x_C) \prod_{k \in N(i) \cup j} m_{kj}(x_{Sk})$$

(6)

where $S_{ij} = C_i \cap C_j$, and $N(i)$ are the neighbors of clique $C_i$ in the junction tree. The belief at clique $C_i$ is proportional to the product of the local evidence at that clique and all the messages coming into clique $i$.
Beliefs for single nodes can be obtained via further marginalization:

\[ M_j(x_i) = \sum_{C_i \in i} M_j(x_i) \quad \text{for } i \in C_j \]  

(8)

The equations (6), (7), and (8) represent generalized belief propagation algorithm which is valid for arbitrary graphs. The BP algorithm defined with (1) and (2) is a special case of GBP-JT, obtaining by noting that the original tree is already triangulated, and has only pairs of nodes as cliques. In this case, sets \( S_j \) are single nodes, and marginalization using eq. (8) is unnecessary.

Let’s show how it works in our example in Figure 1. The network has 10 nodes, 5 anchors (nodes 6-10) and 5 unknowns (nodes 1-5). There is a loop 1-2-4-5-3, so we have to triangulate it by adding two more edges (2-3 and 3-4). Then we can define 8 cliques in the graph:

- \( C_1 = \{x_1, x_2, x_3\} \)
- \( C_2 = \{x_2, x_1, x_4\} \)
- \( C_3 = \{x_1, x_4, x_5\} \)
- \( C_4 = \{x_4, x_9\} \)
- \( C_5 = \{x_5, x_3\} \)
- \( C_6 = \{x_1, x_6\} \)
- \( C_7 = \{x_2, x_7\} \)
- \( C_8 = \{x_3, x_8\} \)

The appropriate potentials of 3-node cliques are given by:

\[ \psi_{C_1}(x_1, x_2, x_3) = \psi_{12}(x_1, x_2) \psi_{13}(x_1, x_3) \]
\[ \psi_{C_2}(x_2, x_3, x_4) = \psi_{24}(x_2, x_4) \]
\[ \psi_{C_3}(x_3, x_4, x_5) = \psi_{35}(x_3, x_5) \psi_{45}(x_4, x_5) \]  

(9)

Note that “virtual” edges do not appear in these equations since they are used only to define cliques. Other cliques, defined over pairs of nodes, are nothing else than potential functions between two nodes already known from standard BP:

\[ \psi_{C_1}(x_4, x_9) = \psi_{49}(x_4, x_9) \]
\[ \psi_{C_5}(x_1, x_6) = \psi_{16}(x_1, x_6) \]
\[ \psi_{C_7}(x_2, x_7) = \psi_{27}(x_2, x_7) \]  

(10)

The junction tree corresponding to the network in Figure 1 is shown in Figure 2. As we can see, “anchor cliques” \( C_4 - C_6 \) do not receive messages, so this graph does not contain loops. Actually, these “anchor cliques” also include one unknown node so we can send them messages, but this node also could be located marginalizing the belief of some other clique.

In the next step, we can compute all messages using equation (6). The complete set of messages is given by:

\[ m_{21}(x_1) = \psi_{16}(x_1, x_6) \]
\[ m_{23}(x_3) = \psi_{510}(x_5, x_{10}) \]
\[ m_{27}(x_2) = \psi_{27}(x_2, x_7) \]
\[ m_{42}(x_4) = \psi_{49}(x_4, x_9) \]
\[ m_{63}(x_6) = \psi_{38}(x_3, x_8) \]
\[ m_{72}(x_2, x_3) = \psi_{27}(x_2, x_7) \psi_{38}(x_3, x_8) \]  

(11)

where asterisk denotes the known location of the anchor node and the messages from "anchor cliques" are directly replaced by appropriate potential function. The beliefs of cliques are computed using equation (7):

\[ M_1(x_1, x_2, x_3) = \psi_{12}(x_1, x_2) \psi_{13}(x_1, x_3) \psi_{27}(x_2, x_7) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) \]
\[ M_2(x_2, x_3, x_4) = \psi_{12}(x_2, x_3) \psi_{13}(x_1, x_3) \psi_{27}(x_2, x_7) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) \]
\[ M_3(x_3, x_4, x_5) = \psi_{12}(x_3, x_4) \psi_{13}(x_1, x_3) \psi_{27}(x_2, x_7) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) m_{42}(x_4, x_9) \]
\[ M_4(x_4, x_5, x_6) = \psi_{35}(x_3, x_5) \psi_{45}(x_4, x_5) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) m_{42}(x_4, x_9) m_{63}(x_6, x_8) \]
\[ M_5(x_5, x_3, x_6) = \psi_{510}(x_5, x_{10}) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) \]
\[ M_6(x_1, x_6) = \psi_{16}(x_1, x_6) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) m_{27}(x_2, x_7) \]
\[ M_7(x_2, x_7) = \psi_{27}(x_2, x_7) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) m_{27}(x_2, x_7) m_{42}(x_4, x_9) \]
\[ M_8(x_3, x_8) = \psi_{38}(x_3, x_8) \psi_{45}(x_4, x_5) \psi_{27}(x_2, x_7) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) m_{27}(x_2, x_7) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) m_{27}(x_2, x_7) \]
\[ M_9(x_4, x_9) = \psi_{49}(x_4, x_9) \psi_{27}(x_2, x_7) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) m_{27}(x_2, x_7) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) m_{27}(x_2, x_7) \]
\[ M_{10}(x_5, x_{10}) = \psi_{510}(x_5, x_{10}) \psi_{38}(x_3, x_8) m_{21}(x_1, x_3) m_{23}(x_3, x_4) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) m_{27}(x_2, x_7) m_{42}(x_4, x_9) m_{63}(x_6, x_8) m_{72}(x_2, x_3) m_{27}(x_2, x_7) \]

Now it’s easy to compute beliefs of single nodes by marginalizing beliefs of cliques using eq. (8). Obviously, it’s sufficient to know beliefs of \( C_1 \) and \( C_3 \) since these cliques include all unknown nodes. Marginalization of \( C_1 \) provides degree of freedom and could be used to check the estimated positions of some nodes (in our case, for nodes 2, 3 and 4).
Finally, in order to use this method for localization, we have to define potential functions. In our case, we assumed that we didn’t obtain a priori information about node position, so single-node potentials are equal to 1 (in opposite case, beliefs computed using eq. (7) have to be multiplied by their own potentials). The pairwise potential between nodes $t$ and $u$ is given by [3]:

$$
\psi_{tu}(x_t, x_u) = \begin{cases} 
P_t(x_t, x_u) p_r(d_{tu} - \|x_t - x_u\|), & \text{if } o_{tu} = 1, \\
1 - P_t(x_t, x_u), & \text{otherwise.} 
\end{cases}
$$

(13)

where $P_t$ is the probability of detecting nearby sensors; in our case we used improved model which assumes that the probability of detecting nearby sensors falls off exponentially with squared distance:

$$
P_t(x_t, x_u) = \exp\left(-\frac{1}{2}\|x_t - x_u\|^2 / R^2\right)
$$

(14)

where $R$ is the transmission radius. The binary variable $o_{tu}$ indicates whether this observation is available ($o_{tu} = 1$) or not ($o_{tu} = 0$). And the last remaining parameter is measured distance. The unknown node $t$ obtains a noisy measurement $d_{tu}$ of its distance from detected node $u$:

$$
d_{tu} = \|x_t - x_u\| + \nu_{tu}, \quad \nu_{tu} \sim p_r(x_t, x_u)
$$

(15)

In our case, we used Gaussian distribution for $p_r$, but, as we can see, it's very easy to change it to any desired distribution (e.g. obtained by training experiment in the deployment area).

The proposed GBP-JT algorithm is not unique. There are a lot of variations of this method; the best known is cluster variation method [8]. However, it can be shown that they are quite similar. For example, in [8] is described the relationship between different region-based approximations. The main goal is achieved in all of them: estimated beliefs are correct in network with loops. However, the price for this is unacceptable large computational cost, so we are going to implement approximated version of GBP-JT algorithm.

4 Nonparametric Generalized Belief Propagation

In order to obtain acceptable spatial resolution for unknown nodes, number of discrete points in the deployment area (e.g. $N_x \times N_y$ for 2D grid) must be too large for GBP to be computationally feasible [3]. Besides, the presence of nonlinear relationships and potentially highly non-Gaussian uncertainties in sensor localization makes GBP undesirable. However, using particle-based representations via non-parametric generalized belief propagation (NGBP), enables the application of GBP to inference in sensor networks. In this section we propose NGBP-JT, particle-based approximation of GBP-JT method for the same example of network from previous section (Figures 1 and 2).

4.1 Drawing initial particles

Let’s draw $N_C$ weighted particles from cliques $C_1$ and $C_3$:

$$\{W_t', X_t'\} = \{W_t', [X_{t1}', X_{t2}', X_{t3}']\}
$$

$$\{W_t', X_t'\} = \{W_t', [X_{t1}', X_{t2}', X_{t3}']\}
$$

(16)

where $W_t'$ represents weight of 6-dimensional particle $X_t'$ from clique $C_n$ which consists from three 2-dimensional particles from node $t$ ($X_{tn}'$). For now, we don’t need particle from clique $C_2$. There is a lot of ways to draw these particles. In general, we can draw all particles uniformly within the deployment area, but it requires

Figure 2. The junction tree corresponding to the network in Figure 1
significant large number of particles (e.g. 100 particles
drawn for each node, corresponds to $100 \times 100 \times 100 = 10^6$
particles of its clique). Therefore, we will immediately
include all information available within the clique: potential
functions given by (9) and (13) which represent our
information about distance between nodes within the
clique. First, we draw particles of node $t$ uniformly within
the deployment area. To draw particle of any neighboring
node $u$, we shift particle of node $t$ in the random direction
for an amount which represents the observed distance
between these two nodes:

$$X_{u,a}^t = X_{a,t}^t + (d_{u} + v)[\sin(\theta^t) \cos(\theta^t)],$$

$$\theta^t \sim \text{Unif}[0, 2\pi], \ j = 1, \ldots, N_c.$$ 

(17)

We will use simplified notation of the above equation:

$$X_{u,a}^t = \text{shift}(X_{a,t}^t, d_{u})$$

(18)

Assuming that we have already drawn samples, e.g.
from nodes 1 and 5, we can compute particles of other
nodes:

$$X_{4,3}^t = \text{shift}(X_{4,5}^t, d_{3})$$

$$X_{4,4}^t = \text{shift}(X_{4,5}^t, d_{4})$$

$$X_{4,5}^t = \text{shift}(X_{4,5}^t, d_{5})$$

(19)

Since these particles are drawn from $\psi_{c_1}$ and $\psi_{c_3}$
respectively, and that we already included all information
which place these particles in high-probabilistic regions
with respect to $X_{4,5}^t$ and $X_{4,5}^t$ (see eqs. (13) and (19)),
all clique’s weights can be approximated with the same value:

$$W_{i}^t = \frac{1}{N_c}, \ j = 1, \ldots, N_c$$

(20)

Note that all particles of nodes within the clique have
one common weight, e.g. $[W_{1}^t, X_{1,1}^t, X_{1,2}^t, X_{1,3}^t]$ . Our initial
set of particles for clique $C_t$ is illustrated in Figure 3.

![Figure 3. Initial set of particles for clique $C_t$](image)

4.2 Computing messages

Having drawn all particles, we can now compute all
messages. Messages $m_{12}$ and $m_{32}$ are function of $m_{12}$ and
$m_{32}$ respectively (see eqs. (11)), so they will be computed
later. Also, messages from the “anchor cliques” will be
directly replaced with appropriate potential function. So we
start with messages $m_{12}$ and $m_{32}$ which depends on $\psi_{C_1}$
and $\psi_{C_3}$ from which we already have drawn particles. Let’s
represent these two messages in slightly different form:

$$m_{12}(x_1, x_2, x_3) = \sum_{i} m_{12}(x_1, x_2, x_3)$$

(21)

$$m_{32}(x_1, x_2, x_4) = \sum_{i} m_{32}(x_1, x_2, x_4)$$

(22)

Defined factors, $m_{12}$ and $m_{32}$, are some kind of
unmarginalized messages, so we’ll call them joint
messages. Now it’s very easy to compute weighted particles
from these joint messages ($W_{m_{12}, X_{1_2}^t}$ ) :

$$W_{12}^t = \psi_{27}(X_{1,2}^t, \psi_{26}(X_{1,3}^t, \psi_{16}(X_{1,4}^t, \psi_{15}(X_{1,5}^t))$$

(23)

$$W_{32}^t = \psi_{49}(X_{3,4}^t, \psi_{38}(X_{3,5}^t, \psi_{510}(X_{3,6}^t, \psi_{51}(X_{3,7}^t))$$

(24)

Before computing final messages, we noticed usual
problem, sample depletion [2], the problem when one, or
few, of the weights are much larger than the rest. This
means that any sample-based estimate will be unduly
dominated by the influence of just few particles. In our
case, it’s expected because we are working in 6-
dimensional space where it’s very hard to draw good
sample (clique with position and shape similar to the right
one – see Figure 3). Therefore, we resample with
replacement [1, 3], which will produce $N_c$ equal-weight
particles ($W_{m_{12}} = 1 / N_c$ ). In our case, we have to resample
from cliques, thus the easiest way is to resample from
single nodes using standard resampling procedure [1], and
then to synchronize indexes in order to keep original shapes
of the particles. This procedure is illustrated for $M_{12}$, by
the following pseudocode:

$$[W_{12}, X_{1_1}, \text{index}] = \text{resample}(W_{12}, X_{1_1})$$

for $j = 1: N_c$

$$X_{1_2}^t = X_{1_2}^t \text{index}(j), X_{1_3}^t = X_{1_3}^t \text{index}(j)$$

(25)

1 Note the difference between $X_{1_2}^t$ and $X_{1_2}^t$
where \( \{W_{21}, X_{11}\} \) is the vector of \( N_C \) particles from node 1 (part of joint message), and \( \text{index} \) is the vector of old (pre-resampled) indexes of new particles.

Now we are ready to compute particles from messages \( \{w_{11}^m, x_{11}^m\} \). The marginalization of joint messages is very easy since we already have weighted particles from them. So we just need to discard one data, and keep the same weights. Thus, they are given by:

\[
X_{11}^j = X_{11}^j(2:3) = [X_{11}^{j_1}, X_{11}^{j_3}] \quad \text{and} \quad W_{11}^j = W_{11}^j(2:3) = W_{11}^j(1) = 1 / N_C
\]

\[
x_{11}^j = X_{11}^j(1:2) = [X_{11}^{j_1}, X_{11}^{j_2}] \quad \text{and} \quad W_{11}^j = W_{11}^j(1:2) = 1 / N_C \quad (26)
\]

Finally, we can compute particles of other two messages, \( m_{23} \) and \( m_{21} \). According to eqs. (11), they are function of \( \psi_{C,m_{21}} \) and \( \psi_{C,m_{23}} \), respectively, so we will draw particles from these products and then re-weight by remainder of eq. (11). Actually, two single-node particles of messages \( m_{21} \) and \( m_{23} \) are already computed (see eqs. (26)), so we have just to draw missing particle using information from \( \psi_{C_1} \), the observed distance between nodes 2 and 3. The result of this procedure is particles of joint messages \( M_{21} \) and \( M_{23} \). Marginalizing them, we obtain final messages \( m_{21} \) and \( m_{23} \). The complete procedure is given as follows:

\[
X_{11}^j = \text{[shift}(x_{12}^j(2),d_{2a})], \quad x_{12}^j(1), \quad x_{12}^j(2)]
\]
\[
W_{11}^j = \psi_{a}(X_{11}^j(3),x_{11}^j)\psi_{a3}(X_{11}^j(2),x_{11}^j)\psi_{27}(X_{11}^j(1),x_{11}^j)w_{12}^j
\]
\[\text{resample and synchronize}\]
\[
x_{11}^j = X_{11}^j(1:2) = [X_{11}^{j_1}(1), X_{11}^{j_2}(2)], \quad W_{11}^j = W_{11}^j(1:2) = 1 / N_C
\]

\[
X_{12}^j = \text{[shift}(x_{12}^j(1),d_{2a})]
\]
\[
W_{12}^j = \psi_{a}(X_{12}^j(3),x_{12}^j)\psi_{a3}(X_{12}^j(2),x_{12}^j)\psi_{27}(X_{12}^j(1),x_{12}^j)w_{12}^j
\]
\[\text{resample and synchronize}\]
\[
x_{12}^j = X_{12}^j(2:3) = [X_{12}^{j_2}, X_{12}^{j_3}(3)], \quad W_{12}^j = W_{12}^j(2:3) = 1 / N_C \quad (27)
\]

\[
X_{13}^j = \text{[shift}(x_{13}^j(2),d_{3a})], \quad x_{13}^j(1), \quad x_{13}^j(2)]
\]
\[
W_{13}^j = \psi_{a}(X_{13}^j(3),x_{13}^j)\psi_{a3}(X_{13}^j(2),x_{13}^j)\psi_{27}(X_{13}^j(1),x_{13}^j)w_{13}^j
\]
\[\text{resample and synchronize}\]
\[
x_{13}^j = X_{13}^j(1:2) = [X_{13}^{j_1}, X_{13}^{j_2}(2)], \quad W_{13}^j = W_{13}^j(1:2) = 1 / N_C \quad (28)
\]

4.3 Computing final beliefs

To estimate beliefs of unknown nodes, we are going to compute beliefs of cliques using already computed particles of messages. According to eqs. (12), beliefs \( M_1 \), \( M_2 \) and \( M_3 \) are function of \( \psi_{C,m_{21}} \), \( \psi_{C,m_{23}} \) and \( \psi_{C,m_{23}} \), respectively, so we will draw particles from these products and then re-weight by remainder of eqs. (12).

Let’s start with \( M_1 \) and its corresponding product \( \psi_{C,m_{21}} \). As we can see in eqs. (9), \( \psi_{C_1} \) includes information about distance between nodes 1 and 2, as well as between nodes 1 and 3. Besides, message \( m_{21} \) includes information about positions of nodes 2 and 3. So we just need to locate node 1, using available positions and distances. It could be done geometrically by intersecting circles, but we prefer statistical approach which is quite easier since we already have weighted particles from them. Therefore, we expect 6-dimensional message. So, if we want to avoid to draw randomly missing particles, we should have to draw single-node particles from \( x_1 \), which is easy to draw samples (\( x_1^{(1)}, x_1^{(2)} \)), and then re-weight by remainder of eq. (12). The following procedure shows it:

\[
X_{11}^j = \text{[shift}(x_{12}^j(2),d_{2a})], \quad x_{12}^j(1), \quad x_{12}^j(2)]
\]
\[
X_{12}^j = \text{[shift}(x_{12}^j(1),d_{2a})]
\]
\[
X_{13}^j = \text{[shift}(x_{13}^j(2),d_{3a})]
\]
\[
W_j = \psi_{a}(X_{11}^j(3),x_{12}^j)\psi_{a3}(X_{11}^j(2),x_{12}^j)\psi_{27}(X_{11}^j(1),x_{12}^j)w_{12}^j
\]
\[
W_j = 1 / N_C \quad (30)
\]

where functions \( \psi_{a}(X_{11}^j(3),x_{12}^j) \), \( \psi_{a3}(X_{11}^j(2),x_{12}^j) \) and \( \psi_{27}(X_{11}^j(1),x_{12}^j) \) are predefined. If we do not obtain “good particle” after \( k_{\text{max}} \) iterations, that’s mean that these two circles cannot intersect, so our particle is the position shifted for \( d_{2a} \) in random direction. This is not a problem because this wrong particle will obviously have a very small weight (filtered by potential functions from anchors). The other problem is bimodality, if the circles intersect in two points; but the wrong particle will be also filtered in same way. The same procedure is done for \( M_2 \) since we have distance between nodes 5 and 4, and as well as between nodes 5 and 3, and message \( m_{23} \) which includes information about positions of nodes 3 and 4.

As we already mentioned, the belief \( M_2 \) is not necessary since other two cliques include positions of all unknown nodes. Anyway, we will show the procedure because it’s slightly different. We have to draw particles from product of two 4-dimensional messages and as result we expect 6-dimensional message. So, if we want to avoid to draw randomly missing particles (e.g. for message \( m_{23}(x_3,x_4) \), we would have to draw single-node particles from \( x_4 \), we will directly draw particles from the product \( \psi_{C_2}(x_2,x_3,x_4)m_{23}(x_2,x_3)m_{23}(x_2,x_4) \) and then re-weight by remainder of eq. (12). The following procedure shows it:

\[
X_{11}^j = \text{[shift}(x_{12}^j(2),d_{2a})], \quad x_{12}^j(1), \quad x_{12}^j(2)]
\]
\[
X_{12}^j = \text{[shift}(x_{12}^j(1),d_{2a})]
\]
\[
X_{13}^j = \text{[shift}(x_{13}^j(2),d_{3a})]
\]
\[
W_j = \psi_{a}(X_{11}^j(3),x_{12}^j)\psi_{a3}(X_{11}^j(2),x_{12}^j)\psi_{27}(X_{11}^j(1),x_{12}^j)w_{12}^j
\]
\[
W_j = 1 / N_C \quad (30)
\]

where functions \( \psi_{a}(X_{11}^j(3),x_{12}^j) \), \( \psi_{a3}(X_{11}^j(2),x_{12}^j) \) and \( \psi_{27}(X_{11}^j(1),x_{12}^j) \) are predefined. If we do not obtain “good particle” after \( k_{\text{max}} \) iterations, that’s mean that these two circles cannot intersect, so our particle is the position shifted for \( d_{2a} \) in random direction. This is not a problem because this wrong particle will obviously have a very small weight (filtered by potential functions from anchors). The other problem is bimodality, if the circles intersect in two points; but the wrong particle will be also filtered in same way. The same procedure is done for \( M_2 \) since we have distance between nodes 5 and 4, and as well as between nodes 5 and 3, and message \( m_{23} \) which includes information about positions of nodes 3 and 4.

2 For the simplicity, updated and old particles are denoted by same symbols.
The final estimated positions of unknown nodes are given by mean values of particles from clique $C_m$:

$$x_{m}^{est} = \frac{\sum_{j=1}^{N_C} W_{m}^{j} x_{m,j}^{/}}{\sum_{j=1}^{N_C} W_{m}^{j}}$$  \hspace{1cm} (31)$$

4.4 Improved sampling procedure

There is one important modification to this algorithm that can reduce significantly the initial number of particles. As we already mentioned, if we draw $N$ particles from one node, generally it corresponds to $N_C = N^3$ particles of a 3-node clique. However, we included information about distance, so our new number for the same clique is $N_C = NN_\theta^2$ where $N_\theta$ represents the number of possible angles. But this number is still very large, so we would like to include additional information.

We assumed that there is no a priori information about node position. However, after very first phase of algorithm, we computed joint messages $M_{12}$ and $M_{32}$ which include current information about positions of cliques $C_1$ and $C_3$. At this point, particles are concentrated in a smaller region (except for very few of them), so we can draw new set of particles around single-node particles of joint messages. For $C_1$, it’s done by following procedure:

$$d^1 = \text{Unif}(0, r), \quad X_{13}^{/} = \text{shift}(X_{13}^{*}, d^1)$$
$$X_{12}^{*} = \text{shift}(X_{13}^{*}, d_{12}), \quad X_{13} = \text{shift}(X_{13}^{*}, d_{13})$$
$$W_{1}^{/} = 1 / N_C$$

- compute messages again

where $r$ is the radius of deployment area of new particles. Computing messages again is very important since we have to run algorithm from the beginning. Of course, for clique $C_3$, we use analog procedure. This improved procedure allows us to decrease initial number of samples to $N_C = NN_\theta^2 / n$ where $n$ is the reducing factor that could be found experimentally. Theoretically, it’s proportional to the ratio of the new to the old deployment area.

5 Simulation Results

We simulated the network from Figure 1 using NBP, GBP-JT and NGBP-JT algorithms. We placed 10 nodes in $2m \times 2m$ area, 5 anchors and 5 unknowns. We set the values of transmission radius ($R = 25\%$ of diagonal length of the deployment area) and standard deviation of measured distance ($\sigma = 0.1m = 14\%R$). Number of iteration for NBP is set to the length of the longest path in the graph ($N_{iter} = 5$), and for GBP-JT/NGBP-JT there is obviously, in our example, just one iteration. Number of particle for NBP is set to $N = 400$, so corresponding number of grid points for GBP-JT is $N_g = 20 \times 20$. For NGBP-JT we used improved sampling procedure with radius $r = 0.1$ and experimentally we found out reducing factor which do not change the accuracy ($n \approx 4$). Assuming that minimum number of possible angles could be approximated with $N_\theta = 10$, we set the number of clique’s particles to $N_C = NN_\theta^2 / n = 10000$.

We ran the simulation for NBP and NGBP-JT, and obtained results shown in Figure 4. Obviously, the location estimates for the NGBP-JT are more accurate since this algorithm is correct for network with loops. NBP algorithm does not converge well for a few nodes, but for some other values of parameters, or with different positions of some nodes, it provides estimates with almost same accuracy as NGBP-JT. However, comparing uncertainties for NBP and NGBP-JT (contours in Figures 4a and 4b), we can see that NBP provides us better guarantees of its estimate.
Finally, we checked the averaged accuracy with respect to the deviation of measured distance for all three methods (Figure 5). The accuracy of GBP-JT is always higher than accuracy of NBP and NGBP-JT. GBP-JT provides us better accuracy than NBP for some usual values of distance deviation (e.g. for measurements using time of arrival, the error is 5-20 %R [11]), and unexpectedly worse accuracy for higher values of mentioned deviation. Anyway, this accuracy could be increased, using larger number of particles (e.g. increasing $N_\theta$), until the “bottom line” defined by the accuracy of GBP-JT.

Comparing with NBP/NGBP-JT, the computational cost of GBP-JT is, of course, very large (311 MFlops) and absolutely unacceptable. Nonparametric approximation of this algorithm decreased it around 25 times, and improved sampling additional 4 times. So the final computational cost of NGBP-JT in simulated example is 7.89 MFlops, just double as many comparing with NBP (3.38 MFlops).

6 Conclusions
As presented in this article, junction tree model is a generalization of belief propagation that is correct for arbitrary graphs. We proposed localization using generalized belief propagation based on junction tree method and nonparametric approximation of this algorithm (NGBP-JT). Our main goal was to solve the problem with loops with some acceptable computational cost and we achieved it using NGBP-JT approach. We can conclude that this algorithm could provide higher accuracy with acceptable computational cost. The main open direction for future work is generalizing this algorithm for an ad-hoc network. This would probably require additional computation necessary for “construction” of junction tree cliques within the network. Moreover, communication cost has to be considered since it’s obvious that we can not exchange thousands of particles without any compression. This will be a part of our future research.

Acknowledgment
This work has been performed in the framework of the ICT project ICT-217033 WHERE, which is partly funded by the European Union and partly by the Spanish Education and Science Ministry under Grant TEC2007-67520-C02-01/2/TCM. Furthermore, we thank partial support by program CONSOLIDER-INGENIO 2010 CSD2008-00010 COMONSENS.

References
A.15 An Experimental Study of RSS-based Indoor Localization using Nonparametric Belief Propagation based on Spanning Trees

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

An Experimental Study of RSS-based Indoor Localization using Nonparametric Belief Propagation based on Spanning Trees

Vladimir Savic and Santiago Zazo
An Experimental Study of RSS-based Indoor Localization using Nonparametric Belief Propagation based on Spanning Trees

Vladimir Savic, Adrián Población, Santiago Zazo and Mariano García
Signal Processing Applications Group, Polytechnic University of Madrid, Spain
{vladimir, santiago, mariano}@gaps.ssr.upm.es, adrian.poblacion.hernandez@alumnos.upm.es

Abstract—Nonparametric belief propagation (NBP) is the well-known method for cooperative localization in wireless sensor networks. It is capable to provide information about location estimation with appropriate uncertainty and to accommodate non-Gaussian distance measurement errors. However, the accuracy of NBP is questionable in loopy networks. Therefore, in this paper, we propose a novel approach, NBP based on spanning trees (NBP-ST) created by breadth first search (BFS) method. In addition, we propose a reliable indoor model based on obtained received-signal-strength (RSS) measurements in our lab. According to our experimental results, NBP-ST performs better than NBP in terms of accuracy and communication cost in the networks with high connectivity (i.e., highly loopy networks).

Keywords—Indoor localization; nonparametric belief propagation; spanning tree; breadth first search; sensor networks.

I. INTRODUCTION

The belief propagation (BP) algorithm, proposed by Pearl [1], is a way of organizing the global computation of marginal beliefs in terms of smaller local computations within the graph. It is one of the best-known graphical model for distributed inference in statistical physics, artificial intelligence, computer vision, error-correcting codes, positioning, etc. The whole computation takes a time proportional to the number of links in the graph, which is significantly less than the exponentially large time that would be required to compute marginal probabilities naively.

Due to the presence of nonlinear relationships and highly non-Gaussian uncertainties, the standard BP is undesirable. In addition, in order to obtain acceptable spatial resolution for the sensors, the discrete space (grid) in the deployment area must be made too large for BP to be computationally feasible. However, particle-based approximation via non-parametric belief propagation (NBP), proposed by Ihler et al. [2], makes BP acceptable for inference in sensor networks. The main features of this approach are easy implementation in a distributed fashion and sufficiency of a small number of iterations to converge. Furthermore, NBP is capable to provide information about location uncertainties and to accommodate non-Gaussian measurement errors. This is the main advantage of NBP comparing with well-known deterministic methods, e.g., [3], [4]. In our application (indoor positioning), the distance error model is not even close to Gaussian model, thus this is our motivation for choosing NBP.

However, BP convergence is not guaranteed in a network with loops [1], [5] or even with convergence, it could provide us less accurate estimates. Regarding localization using NBP, the convergence is usually sufficient, but the accuracy is questionable. In the current state of the art, there are few solutions for networks with loops, but mostly they have not been used for the localization. The well-known solutions based on generalized belief propagation (GBP) [5]–[7], which are based on clusters or cliques, are still very complex for the large-scale ad-hoc/sensor networks. Therefore, in this paper, we propose NBP based on spanning trees (NBP-ST) created by breadth first search (BFS) method [8] which is optimal for the unweighted graphs. NBP-ST algorithm represents two (or more) independent runnings of the NBP algorithm based on formed spanning trees. In order to obtain realistic distance measurements for indoor scenario, we performed experiments in our lab using IRIS wireless sensor nodes equipped with AT86RF230 transceiver [9], [10]. According to our experimental results, NBP-ST performs better than NBP in terms of accuracy and communication cost in the networks with high connectivity (i.e., highly loopy networks). Furthermore, the communication cost is nearly constant with respect to the transmission radius. However, the drawbacks of proposed method are slightly higher (10-40%) computational cost and poor performance in low connected network. Anyway, the latter is not a problem since for the low-connected networks we can keep using NBP.

The remainder of this paper is organized as follows. In Section II, we provide a short review on cooperative localization using NBP. In Section III, we propose NBP method based on spanning trees. Experimental results for indoor scenario are presented in Section IV. Finally, Section V provides some conclusions and future work perspective.

II. COOPERATIVE LOCALIZATION USING NONPARAMETRIC BELief PROPAGATION

We consider the case in which some small number of anchor nodes, obtain their coordinates via GPS or by installing them at points with known coordinates, and the rest, unknown nodes, must determine their own coordinates. We suppose that all sensors with unknown positions obtain noisy
distance measurements of nearby subset of the other sensors in the network. This measurements can be obtained using a broadcast transmission from each sensor as all other sensors listen. Typical measurements techniques [11] are time of arrival (TOA), time difference of arrival (TDOA), receive signal strength (RSS) and angle of arrival (AOA). In this paper, we use RSS measurements.

Let us denote the received power by $P_r(d)$. It is empirically accepted to model $P_r(d)$ as log-normally distributed random variable with a distance dependent mean value. Thus, converted in dBm, it is given by:

$$P_r(d)[dBm] = P_0(d_0)[dBm] - 10n_p \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma$$  \hspace{1cm} (1)

where $P_0(d_0)$ is known reference power value in dBm milliwatts at a reference distance from the transmitter, $n_p$ is the path loss exponent that measures the rate at which the RSS decreases with distance, typically between two and four depending on the environment, $X_\sigma$ is a zero mean Gaussian distributed random variable with standard deviation $\sigma$ which accounts for the random effects of shadowing. It is trivial to conclude from (1) that, given $P_r(d)[dBm]$, the estimated distance between a transmitter and receiver is:

$$d = d_0 \cdot 10^{\frac{P_r(d)[dBm] - P_0(d_0)[dBm]}{10n_p}} \cdot 10^{X_\sigma/10}$$  \hspace{1cm} (2)

As we can see, the distance error is multiplicative (i.e., log-normally distributed) which means that RSS-based distance estimates have variance proportional to their true distance. Therefore, RSS is most valuable in high-density sensor networks which we target in this paper.

Regarding detailed statistical framework for localization using BP/NBP, we refer the reader to our previous results [12]. We define here only BP and appropriate potential functions. Since the single-node potential (prior information about position) is equal to one (no prior information), we only need to define pairwise potential function. The pairwise potential (probabilistic information about distance) between nodes $i$ and $u$, is given by:

$$\psi_{iu}(x_i, x_u) = \begin{cases} P_d(x_i, x_u)p_{d}(d_{iu} - \|x_i - x_u\|), & \text{if } o_{iu} = 1 \\ 1 - P_d(x_i, x_u), & \text{otherwise} \end{cases}$$  \hspace{1cm} (3)

where $d_{iu}$ is estimated distance, $o_{iu}$ is a binary variable which indicates if there is a detection or not, and $P_d(x_i, x_u)$ is probability of detection. We can now estimate the sensor positions by applying the BP algorithm. Each node $i$ computes its belief $M_i^t(x_i)$, the posterior marginal distribution of 2D position $x_i$ at iteration $i$, by taking a product of its local potential $\psi_t$ with the messages from its set of neighbors $G_i$:

$$M_i^t(x_i) \propto \psi_t(x_i) \prod_{u \in G_i} m_{ui}^t(x_i)$$  \hspace{1cm} (4)

The messages $m_{ui}$, from node $u$ to node $t$, are computed by:

$$m_{ui}^t(x_t) \propto \sum_{x_u} \psi_{ui}(x_t, x_u) \frac{M_u^{t-1}(x_u)}{m_{tu}^{t-1}(x_u)}$$  \hspace{1cm} (5)

In the first iteration of this algorithm, it is necessary to initialize $m_{ui}^1 = 1$ and $M_i^1 = 1$ (in case of no prior information) for all $u, t$, and then repeat computation using (4) and (5) until sufficiently converge.

Since we use NBP, the belief and message update equations, (4) and (5), are performed using stochastic approximations, in two stages: first, drawing $N$ weighted particles $\{(x_1^{j,1}, x_1^{j,2})\}$ from the belief $M_1^t(x_t)$, then using these particles to approximate each outgoing message $m_{it}$. The detailed description of the algorithm can be found in [2].

### III. NBP BASED ON SPANNING TREES

If we ignore the existence of loops and permit the nodes to continue communicating with each other, messages may circulate indefinitely around these loops, and the process may not converge to a stable equilibrium, or even if it converges, it could provide us less accurate estimates [1], [5], [6]. That is the main reason why we are going to implement NBP based on spanning trees (NBP-ST).

We start by describing the basics of graphical models. An undirected graph $G = (V, E)$ consists of a set of nodes $V$ that are joined by a set of edges $E$. A loop is a sequence of distinct edges forming a path from a node back to itself. A tree is a connected graph without any loops. A spanning tree is an acyclic subgraph that connects all the nodes of the original graph. A root node is a node without parent and leaf node is a node without children. In order to define a graphical model, we place at each node a random variable $x_s$ taking values in some space. In case of localization, this random variable represents the 2D or 3D position and each edge indicate that measurement is available. If we exclude anchors, the graph is undirected, but only for the first phase (spanning tree formation) we assume that it is directed (starting from chosen root node).

The optimal method for spanning tree formation for unweighted graphs is breadth first search (BFS) method [8]. It begins at the root node and explores all the neighboring nodes. Then each of those neighbors explores their unexplored neighbor nodes, and so on, until all nodes are explored. In this way, there will not be a loop in the graph because all nodes will be explored just once. The worst case complexity is $O(v + e)$, where $v$ is the number of nodes and $e$ is the number of edges in the graph, since every node and every edge will be explored in the worst case.

In case of NBP localization, we exclude all the anchors from the BFS algorithm since they do not form the loops in the graph (they just send, and never receive the messages). A graph generally has a large number of spanning trees, but since our graph is unweighted we choose few (minimum
2) of them in a partly random way. In order to choose spanning tree, it is sufficient to choose root nodes for all spanning trees, then the algorithm will automatically set the spanning tree. Taking into account that we want to maximize the difference between two spanning trees, the root nodes can be chosen in two ways:

- The first root node we choose randomly from the set of all unknown nodes. The second root node has to be as far as possible from the first root node. Thus, it should be one of the leaf nodes which is the maximum-hop away from the root.
- We choose two (or more) anchor nodes which are far away from each other. The closest unknown nodes to chosen anchor nodes will be chosen as roots.

If we want to form more spanning trees, the analog procedure can be used. Since we will apply this algorithm for the indoor scenario, where the anchor nodes are usually fixed, we will use the second option. Note that, using BFS, it is not possible to form two spanning trees with completely different edges, and that usually some of the edges will be out of both spanning trees. Thus, if we want to ensure that all edges are used, we have to add more spanning trees, but it is usually not necessary since it would only provide us a redundant information. It is especially the case in highly connected networks (i.e., with high transmission radius) which we target in this paper.

The NBP method is naturally distributed through the network which means that there is no central processor which will handle all computations. Therefore, the proposed BFS method has to be done in a distributed fashion. This can be simply done if each unknown node initially broadcast its ID to all neighbors, which will continue to broadcast to others, and so on, until each unknown node has a list of all unknown nodes in the graph. One anchor node (e.g., with lowest ID) has to be assigned to choose the root node from that list and give him the permission to start BFS algorithm. Then, the chosen root node has all initial data to start the BFS algorithm, and, when it is necessary, has only to broadcast all data to all its neighbors. In the end, the last visited node has the output result (edges in spanning tree) and it just has to start multihop broadcast until each unknown node receives this result.

Finally, NBP-ST algorithm represents two (or more) independent runnings of the NBP algorithm based on formed spanning trees. Each running will provide us weighted particles of the node beliefs computed by (4). The collection of particles from all spanning trees represents our final output, from which we can easily extract any parameter that we need (e.g., mean value for location estimate). The pseudocode in Alg. 1 illustrates the NBP-ST method.

### IV. Experimental Results

In this section we start with the description of the setup used for the experiments performed in our lab. We then create reliable indoor model using obtained measurements and import all data into Matlab in order to check the performance of the proposed method in high-density sensor network.

**Algorithm 1** NBP-ST method for localization

1: for all nodes do
2: Take sensing actions
3: Set all parameters to the initial values
4: Broadcast own and all received IDs and listen for other sensor broadcasts (until receive all IDs)
5: end for
6: Set a list of nodes for BFS (excluding anchors)
7: Choose root node
8: for all spanning trees do
9: Run BFS
10: Run NBP on defined spanning tree
11: Choose root node as far as possible from the previous roots
12: end for
13: Fuse all beliefs into one and compute location estimates

Figure 1: (a) Crossbow’s IRIS wireless sensor node, (b) Illustration of the experiment in our lab

#### A. Experimental Setup

For our experiments, we use Crossbow’s IRIS wireless sensor nodes (Fig. 1a) equipped with AT86RF230 transceiver. The AT86RF230 is high performance RF-CMOS 2.4 GHz radio transceiver specially targeted for low cost ZigBee/IEEE802.15.4 applications. The transmitter provides programmable output power: -17 dBm up to 3 dBm. The receiver, with -101 dBm sensitivity, generates digital signal with 3 dB granularity. The data is stored in a 128-byte dual port SRAM, from which 8 bytes are reserved. More details in [9], [10].

In order to estimate the distance between sensors, we placed two sensors, 2m from the floor, in our 5m x 10m lab (Fig. 1b) and set the transmission power to 3 dBm. Then, we obtained RSS measurements at 8 equidistant intersensor distances (k·1.2m, k = 1, ..., 8). For each of them, we obtained 1000 measurements. Because of the 3 dB
B. Indoor Modeling using Distance Measurements

Using obtained RSS measurements, our goal is to obtain all necessary parameters for indoor model: path-loss exponent, reliable distance estimation, probability of detection and potential functions.

Path-loss exponent. First, we define a reference point \( (d_0 = 2.4m, -61 \text{ dBm}) \). The path-loss exponent \( (n_p) \) could be easily obtained using another reference point, but this is not an optimal way. The better option is to, using all measured data, minimize the root mean square (RMS) error with respect to \( n_p \):

\[
e_{\text{rms}}^d(n_p) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (d_{\text{measured}}^i(n_p) - d_{\text{true}}^i)^2}
\]

(6)

where \( n \) is the number of inter-sensor distances (in our case, \( n = 8 \)) and \( d_{\text{measured}}^i(n_p) \) is given by:

\[
d_{\text{measured}}^i(n_p) = d_0 \cdot 10^{-\frac{\text{RSS}_i - \text{RSS}_0}{10n+p}}
\]

(7)

Note that equations (2) and (7) are equivalent since the measured power includes the noise which accounts for the random effects of shadowing. According to Fig. 2a (dashed line), the optimal value of the path-loss exponent is \( n_p = 2.7 \).

Reliable distance estimation. Using obtained measurements and estimated \( n_p \), we can estimate the distance. As we expected our indoor model is not similar to the ideal one (Fig. 2b), so the distance cannot be always trustfully estimated using (7). For instance, the averaged received power of -66 dBm corresponds to three different distances (4.6m, 7m and 9.6m), so the sensor has no other option, but to guess. This is because the power is not monotonically decreasing function of the distance. Therefore, we have to cut out the area below the threshold power (-64 dBm) because this area corresponds to the non-monotonic part of the function. Above the threshold, each received power corresponds to the unique distance, which makes this model reliable for our scenario. In addition, since we excluded data below the threshold, we must re-estimate \( n_p \) using only the remaining data. According to Fig. 2a, \( n_p = 1.2 \). We illustrated in Fig. 2c, the distance estimation which corresponds to the true value of 1.2m. As we can see, the error model \((d_{\text{measured}} - d_{\text{true}})\) is not similar to the log-normal distribution, so we will not use any parametric form of the error distribution. Moreover, we have three different error models (for 1.2m, 2.4m, and 3.6m). Thus, in order to import this error model into Matlab, we will simply draw the samples from the nearest error model and add it to the true distance (i.e., this is nearest neighbor interpolation, so for the true value of e.g., 2.9m, we use error model for 2.4m).

Probability of detection. For each inter-sensor distance, we found that RSS is above the power defined by sensitivity. This is expected because we set the transmission power to the maximum which could even provide us around 75m radius, according to ZigBee standard. Anyway, we have to follow defined reliable model, so we assume that, if the power is less than threshold (-64 dBm), there is no communication between nodes. This could be easily forced by software. As we can see in Fig. 2b, the corresponding distance is 4m, so this will be the maximum value of transmission radius. Note that, in our case, we didn’t detect communication failures (link quality indicator is always maximum), so we set \( P_d = 1 \) in the transmission range, and otherwise \( P_d = 0 \). This is expected due to the very small distance between nodes.

Potential functions. Since we don’t have any a priori information about positions of unknown nodes, we have to define only pairwise potential. According to (3), given anchor node (or particle of unknown node), the position of other node is shifted in the random direction by mea-
Figure 3: Pairwise potential function $\psi_{ut}(x^*_{\text{t}}, x_u)$ ($x^*_{\text{t}}$ - anchor, $x_u$ - unknown) using (a) log-normal model, (b) indoor model from our lab.

Figure 4: (a) Original network, (b),(c) two corresponding spanning trees. Connections between anchors (marked by red squares) and unknowns (marked by black circles) are not shown.

Figure 5: Comparison of accuracy

C. Simulations

We placed 50 unknowns and 10 anchors in 5m x 10m area (Fig. 4). Unknown nodes are deployed randomly within this area and anchor nodes are fixed (8 along the edges and 2 in center area). This constraint, realistic for indoor scenario, helps the unknown nodes near the edges which suffer from low connectivity. The number of iteration is set to $N_{\text{iter}} = 3$. According to our analysis this number is sufficient for good convergence. All simulations are done for $N = 50$ particles with respect to the transmission radius ($R = 2m - 4m$). Finally, each point in the simulations represents the average over 30 Monte Carlo trials.

Using the defined scenario, we compared NBP and NBP-ST algorithms. For NBP-ST, we used 2 and 3 spanning trees. The error is defined as Euclidean distance between true and estimated location. As we can see in Fig. 5, NBP-ST performs better than NBP starting from some value of $R$, which controls the connectivity. We can conclude the same for the coverage (Fig. 6), which represents the percentage of located nodes with error less than predefined tolerance. Obviously, for higher values of $R$, there is a large number of loops in the network (hundreds, in our case) which decreases the performance of the NBP method. For lower values of $R$, we could expect that NBP-ST performs with higher (or same) accuracy, but we cannot forget that, by using only 2 or 3 spanning trees, we didn’t include all information (i.e., removed edges) that we have. Thus, in this case the NBP overperforms NBP-ST.

To measure the communication cost, we count elementary messages, where one elementary message is defined as simple scalar data. We assumed that this data is represented in single precision floating-point format that occupies 4 bytes in the memory. As we have already mentioned, 8 bytes are already reserved, so the size of elementary message is 12 bytes. According to Fig. 7, NBP-ST performs better than NBP for $R > 3.3m$ only if we use 2 spanning trees. In order to explain this we have to remember two main things we have taken into account: removing the edges in order to form the spanning trees and running NBP two times in these spanning trees. First operation decreases the communication, but the second one increases it. Therefore, in low connected networks the second operation predominates, but in high connected networks the first one predominates. Regarding computational cost, NBP (0.15-0.25 MFlops per node) slightly overperforms NBP-ST (0.21-0.28 MFlops per node). Furthermore, it is important that the communication cost is nearly constant with respect to the transmission radius. This feature provides us more precise information about battery life. Finally, if we use 3 or more spanning trees, both computational and communication cost will be obviously significantly higher.

The final conclusion is that NBP-ST (with 2 spanning trees) algorithm performs better than NBP in terms of accuracy and communication cost, for $R > R_{\text{min}}$. In our case $R_{\text{min}} = 3.4m$, but this parameter depends on the density in the network (i.e., average connectivity). On the other hand, if the unique goal is accuracy, user should increase the number of spanning trees.
V. CONCLUSIONS AND FUTURE WORK

As presented in this paper, NBP localization algorithm has poor performance in highly loopy networks. Moreover, the connectivity in these networks is very high which makes communication burden for low-power applications. Therefore, we proposed NBP-ST method based on spanning trees created by the BFS method which is optimal for the unweighted graphs. The BFS method is done in a distributed way which makes the algorithm applicable in ad-hoc/sensor networks. We can conclude that NBP-ST method performs better than NBP in terms of accuracy, and communication cost in highly connected networks. However, the drawbacks of proposed method are slightly higher computational cost and poor performance in low connected network. There remain few open directions for the future work. One possible future line is the implementation of localization algorithm based on nonparametric generalized belief propagation (NGBP). Some versions already exist, but they are still very complex for the large-scale sensor networks. Furthermore, real-time tracking using these methods could be an interesting direction. This will be a part of our future research.

ACKNOWLEDGMENT

This work is supported by the FPU fellowship from Spanish Ministry of Science and Innovation. Furthermore, we thank partial support by project ICT-217033 WHERE, program CONSOLIDER-INGENIO 2010 CSD2008-00010 COMONSENS and National Project M3HF.

REFERENCES


©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Vladimir Savic and Santiago Zazo

Vladimir Savic and Santiago Zazo
Signal Processing Applications Group, Polytechnic University of Madrid
Avda. Complutense 30, 28040 Madrid, Spain
{vladimir, santiago}@gaps.ssr.upm.es

Abstract – Nonparametric belief propagation (NBP) is well-known probabilistic method for cooperative localization in sensor networks. However, due to the double counting problem, NBP convergence is not guaranteed in the networks with loops or even if NBP converges, it could provide us less accurate estimates. The well-known solution for this problem is nonparametric generalized belief propagation based on junction tree (NGBP-JT). However, there are two problems: how to efficiently form the junction tree in an arbitrary network, and how to decrease the number of particles while keeping the good performance. Therefore, in this paper, we propose the formation of pseudo-junction tree (PJT), which represents the approximated junction tree based on thin graph. In addition, in order to decrease the number of particles, we use a set of very strong constraints. The resulting localization method, NGBP based on PJT (NGBP-PJT), overperforms NBP in terms of accuracy and communication cost in any arbitrary network.

Keywords: cooperative localization, belief propagation, nonparametric generalized belief propagation, clique tree, junction tree, wireless sensor networks.

1 Introduction
We consider the case in which some small number of anchor nodes, obtain their coordinates via GPS or by installing them at points with known coordinates, and the rest, unknown nodes, must determine their own coordinates. We suppose that all sensors with unknown positions obtain noisy distance measurements of nearby subset of the other sensors in the network. Typical measurement techniques [1] are time of arrival (TOA), time difference of arrival (TDOA), received signal strength (RSS) and angle of arrival (AOA).

Nonparametric belief propagation (NBP) [2] is well-known probabilistic method for cooperative localization in sensor networks. It is capable to provide information about location estimation with appropriate uncertainty and to accommodate non-Gaussian distance measurement errors. This is the main advantage of NBP comparing with well-known deterministic methods [1]. However, due to the double counting problem, NBP convergence is not guaranteed in the networks with loops [3] or even if NBP converges, it could provide us less accurate estimates. Since we have already provided detailed description of this problem in [5], we'll skip it in this paper.

The well-known solution for the loopy graphs is generalized belief propagation based on junction tree (GBP-JT) method [4], which is a standard method for exact inference in graphical models. In our previous work [5], we applied nonparametric approximation of this method (NGBP-JT) for the localization in small-scale network and showed that it can overperform NBP in terms of accuracy. However, there remained two main problems: how to efficiently form the junction tree in an arbitrary network, and how to decrease the number of particles while keeping the good performance. Therefore, in this paper, we propose the formation of pseudo-junction tree (PJT), which represents the approximated junction tree based on thin graph. In addition, in order to decrease the number of particles, we use a set of very strong constraints: the measured distances and the bounded boxes. The resulting localization method, NGBP based on PJT (NGBP-PJT), overperforms NBP in terms of accuracy and communication cost in any arbitrary network.

The remainder of this paper is organized as follows. In Section 2, we provide the background on junction tree formation. In Section 3, we propose pseudo-junction tree formation. Localization using NGBP-PJT method for an arbitrary graph is proposed in Section 4. Simulation results are presented in Section 5. Finally, Section 6 provides some conclusions and future work perspective.

2 Junction Tree Formation
We start by describing the basics of graphical models. A graphical model is a probabilistic model for which a
graph denotes the conditional independence structure between random variables. There are two main types: directed graphical models (or Bayesian networks) and undirected graphical models (or Markov networks). For the cooperative localization problem, we use Markov networks.

An undirected graph \( G = (V, E) \) consists of a set of nodes \( V \) that are joined by a set of edges \( E \). A loop is a sequence of distinct edges forming a path from a node back to itself. A clique is a subset of nodes such that for every two nodes in clique, there exists an link connecting the two. A tree is a connected graph without any loops, and a spanning tree is an acyclic subgraph that connects all the nodes of the original graph. Regarding directed graphs, we define a root node, which is a node without parent, and leaf node, which is a node without children. In order to define a graphical model, we place at each node a random variable taking values in some space. Each edge in the graph represents the information about conditional dependency between two connected nodes. In this way, we reduced the complexity of computing the joint probability density function (pdf) of the graph. In case of cooperative localization, this random variable represents the 2D or 3D position, and each edge, which indicates that measurement is available, represents probabilistic information about distance. If we exclude the anchors nodes, the graph is obviously undirected.

Junction tree (JT) algorithm is a method for the exact inference in arbitrary graph. That can be proved elimination procedure \[4\]. It is based on triangulated graph, i.e., a graph with additional “virtual” edges so that every loop of length more than 3 has a chord. In triangulated graph, each 3-node loop (which is not part of any larger clique) represents 3-node clique, and each edge (which is not part of any 3-node clique) represents 2-node clique. If possible, larger cliques (> 3) should be avoided using optimal triangulation procedure. Using these cliques as hypernodes, we can define a cluster graph \[6\] by connecting each pair of cliques with minimum one common node (i.e., non-empty intersection). Using cluster graph, we can create a lot of clique trees, but just very few of them represent the junction tree. The junction tree is a maximum spanning tree of the cluster graph, with weights given by the cardinality of the intersections between cliques. It is already proved \[6\] that this is a way to satisfy the main property of the junction tree, the running intersection property (RIP). The RIP is satisfied if and only if each node, which is in two cliques \( C_i \) and \( C_j \), is also in all cliques on the unique path between \( C_i \) and \( C_j \). If the RIP is not satisfied for any node, there is no theoretical guarantee that the belief of that node in one clique is the same as its belief in another clique.

We illustrate the whole procedure in Fig. 1. We first triangulate the graph by adding the edge between nodes 2 and 5 (Fig. 1a). Then we form the cluster graph (Fig. 1b) with cliques \( C_i(t, u, v) \) and separator sets \( S_{ij}(q, r) \) \((S_{ij} = C_i \cap C_j)\), where \( t, u, v \) are the nodes in the clique, and \( q, r \) are the separator nodes. Finally, any spanning tree represents the clique tree like in Fig. 1c and Fig. 1d. The tree in Fig. 1d is maximum spanning tree (\(|S_{12}| > |S_{13}|\)), so it represents the junction tree of the initial graph. Note that the tree in Fig. 1c does not satisfy RIP since the node 6, which is in \( C_1 \) and \( C_2 \), is not in \( C_3 \).

The described procedure represents the exact formation of junction tree, also called chordal graph method. The main problem of this approach is the triangulation phase. Finding, a minimum triangulation, i.e., one


Algorithm 1 Searching for thin graph and cliques using modified Breadth First Search (BFS) method

1: Input: node list \( Q \) and root node \( \text{root} \)
2: Create more node lists: \( \text{Nodes}, \text{NewVisit} \leftarrow Q \)
3: Set current root: \( r \leftarrow \text{root} \)
4: Create list of neighbors for all nodes \( n \in Q: G_n \)
5: while \( \text{Nodes} \) is not empty do
6:    for all nodes \( t \in G_r \) do
7:        if \( t \in \text{Nodes} \) then
8:            Remove \( t \) from \( \text{Nodes} \)
9:            Insert \( t \) in \( \text{WaitingRoots} \)
10:           Insert \( d_{r,t} \) in \( T \)
11:        else if \( d_{r,t} \notin T \) and \( r \in \text{NewVisit} \) then
12:            Insert \( d_{r,t} \) in \( T \)
13:            Remove \( r \) from \( \text{NewVisit} \)
14:        Create 3-node cliques:
15:        for all \( q \in \text{PreviousRoots} \) do
16:            if \( \{d_{q,t}, d_{t,q}\} \in T \) then
17:                \( C_{3\text{nodes}} \leftarrow \{r, t, q\} \)
18:            end if
19:        end for
20:    end if
21: end for
22: Insert \( r \) in \( \text{PreviousRoots} \)
23: Set current root: \( r \leftarrow \) first unused node from \( \text{WaitingRoots} \)
24: end while
25: Create 2-node cliques \( C_{2\text{nodes}} \): each edges in \( T \) which is not subset of \( C_{3\text{nodes}} \)
26: Output: thin graph \( \{Q, T\} \) and cliques \( C = C_{2\text{nodes}} \cup C_{3\text{nodes}} \)

where the largest clique has minimum size, is \( NP \)-hard problem due to the number of permutations that must be checked. Of course, there exist approximate methods which are less expensive, but still too costly. For more details, see Chapter 10 in [6].

3 Pseudo-Junction Tree Formation

Due to the high complexity of the optimal junction tree formation, it’s necessary to find some approximation that will be suitable for localization scenario. Thus, we are going to make the following assumptions:

(a) The number of cliques should be reasonable (i.e., in the order of number of nodes),

(b) In order to decrease the dimensionality of the problem, each clique will include no more than 3 nodes.

(c) Since the triangulation is expensive procedure, we are going to avoid it, even if it causes the break of RIP for some small percentage of the nodes.

After these approximations, the final result represents, strictly speaking, the clique tree. However, since it is very close to junction tree (measured by percentage of nodes that satisfies RIP), we call it pseudo-junction tree (PJT).

In order to satisfy the conditions (a) and (b), we need to decrease the number of edges in the graph by formation of thin graph. That can be easily done using modified version of breadth first search (BFS) method. Standard BFS method [7] begins at randomly chosen root node and explores all the neighboring nodes. Then each of those neighbors explores their unexplored neighbor nodes, and so on, until all nodes are explored. In this way, there will not be a loop in the graph because all nodes will be explored just once. Thus, the final result of BFS is spanning tree. The worst case complexity is \( O(v + e) \), where \( v \) is the number of nodes and \( e \) is the number of edges in the graph, since every node and every edge will be explored in the worst case.

Nevertheless, the spanning tree is very coarse approximation of the original graph since it excludes a lot of edges from the graph. For example, in any spanning tree, one communication failure breaks the graph into the two parts. As a consequence, we need more spanning trees in order to have reasonable accurate inference in graphical models. Therefore, we modify standard BFS method by permitting each root node to make the additional visit to the node that was already visited by some of the previous roots. All edges found by first and second visit, together with all nodes from original graph, represent the thin graph. In addition, the second visit will automatically form a loop, so we use it to form 3-node clique. The 2-node cliques can be found easily by taking all edges that appear in thin graph, but which are not already subset of any 3-node clique. The worst complexity is \( O(v + e + v_n) \), since for each of the additional visit we need to check all previous roots (i.e., \( n_r \) represents the average number of previous roots). The detailed pseudocode is shown in Alg. 1, and an example of original graph and corresponding thin graph is shown in Fig. 2a and Fig. 2b, respectively.
The main benefit of thin graph is that it mainly includes 3-node loops. The number of these loops, which is obviously always less than number of nodes, is nearly constant with respect to connectivity, so the number of cliques will be constant as well. On the other hand, the main drawback is that there exist the loops which include more that 3 nodes, but just very few of them. These loops should be triangulated, but we prefer to avoid it in order to keep reasonable complexity. Thus, for n-node loops (n > 3), we form maximum n 2-node cliques, using each edge (which is not already subset of any 3-node clique) of the loop as a clique.

Having defined cliques, we can form the cluster graph by connecting all pairs of cliques with non-empty intersection. As we already mentioned, the junction-tree, as well as pseudo-junction tree, is the maximum spanning tree of the cluster graph. It can be found using e.g., Prim’s algorithm [8], as shown in Alg. 2. The Prim’s algorithm is a method that finds a maximum (minimum) spanning tree for a connected weighted undirected graph. That means that the total weight of all the edges in the final tree is maximized (minimized). In our case, the algorithm starts with a list (i.e., CurrentList in Alg. 2) which initially includes only randomly chosen root clique. At each step, among all the edges between the cliques in the list and those not in the list yet, it chooses the one with maximum weight and increases the list by adding the explored clique. Finally, it stops when all the cliques are spanned. The example of pseudo-junction tree is shown in Fig. 2c. The worst case complexity is $O(e \cdot \log(v))$ [8], but in our case the weights are binary, so it will be significantly faster.

The BP/GBP methods are naturally distributed through the graph which means that there is no central unit which will handle all computations. Thus, the proposed pseudo-junction tree formation has to be done in a distributed way. It’s already well-known that there is straightforward distributed way to form any spanning tree. Since this part is out of topic of this paper, we refer the reader to [9]. However, due to the modification of standard BFS algorithm, we explain here how to form the cliques in distributed way. This can be easily done, when root node visit one of the already visited nodes, by adding node’s IDs in the (initially empty) clique list. This list must be broadcasted to all the nodes in the list, and to the next root node. Having defined all cliques, it remains to define the communication between neighboring cliques. Since, the separator sets, between each pair of neighboring cliques, are always non-empty, the separator nodes are responsible to perform the communication. For example, in Fig. 2c, the node 3 will request all the data from node 9, and upon receiving, it will send the data to node 10, and vice versa.

The approximations we made usually break the RIP for some small number of nodes. For instance, in the pseudo-junction tree in Fig. 2c, the node 10 (due to the non-triangulated 4-node loop: 3-9-5-10), and node 7 (due to the appearance of 4-node clique: 2-6-5-7) do not satisfy the RIP. Therefore, we don’t have a guarantee that the belief of that node in one clique is the same as its belief in another clique [6]. Anyway, for the localization, this is not a problem since we incorporate additional constraints (see Section 4.1) for the initial set of particles. Regarding other applications, this algorithm could be used as well, but we recommend the potential users to test it before starting any implementation.

4 Nonparametric Generalized Belief Propagation

Generalized Belief Propagation based on junction tree (GBP-JT) is a standard method for exact inference in graphical model. This can be proved using elimination procedure, [4]. Given cliques $C_i$ and its potentials $\psi_{C_i}(x_{C_i})$, and given the corresponding junction tree which defines links between the cliques, we send the following message from clique $C_i$ to clique $C_j$ by
the message update rule:

$$m_{ij}(x_{S_{ij}}) = \sum_{C_i \setminus S_{ij}} \psi_{C_i}(x_{C_i}) \prod_{k \in G_i \setminus j} m_{ki}(x_{S_{ki}})$$  \hspace{1cm} (1)

where $S_{ij} = S_{ji} = C_i \cap C_j$, and where $G_i$ are the neighbors of clique $C_i$ (including anchor nodes, which are not part of PJT). The belief at clique $C_i$ is proportional to the product of the local evidence at that clique and all the messages coming into clique $i$:

$$M_i(x_{C_i}) \propto \psi_{C_i}(x_{C_i}) \prod_{j \in G_i} m_{ji}(x_{S_{ji}})$$  \hspace{1cm} (2)

Finally, the single-node beliefs can be obtained via further marginalization. Equations (1), (2) represent GBP-JT algorithm which is valid for arbitrary graphs. The standard BP algorithm [2] is a special case of GBP-JT, obtaining by noting that the original tree is already triangulated, and has only pairs of nodes as cliques. In that case, sets $S_{ij}$ are single nodes, and marginalization is unnecessary.

In order to adapt GBP-JT to iterative scenario for cooperative localization, the previous equations, at iteration $m + 1$, can be written as:

$$m_{ij}^{m+1}(x_{S_{ij}}) = \frac{1}{m_{ji}^m(x_{S_{ji}})} \sum_{C_i \setminus S_{ij}} M_i^m(x_{C_i})$$  \hspace{1cm} (3)

$$M_i^{m+1}(x_{C_i}) \propto \psi_{C_i}(x_{C_i}) \prod_{j \in G_i} m_{ji}^{m+1}(x_{S_{ji}})$$  \hspace{1cm} (4)

At the beginning, it’s necessary to initialize $m_{ij}^1 = 1$, and $M_i^1 = \psi_{C_i}$. The clique potentials are given as a product of all single-node and pairwise potentials. The single-node potential (the prior) of node $t$ is given by $\psi_t(x_t) = 1$ within the bounding box, and otherwise $\psi_t(x_t) = 0$. The bounding box of node $t$, created using approximated distances to anchors as constraints [10], represents the region of the deployment area where the node $t$ is localized. The pairwise potential $\psi_{tu}$, which represents the probabilistic information about the distance between nodes $t$ and $u$, is given by:

$$\psi_{tu}(x_t, x_u) = \begin{cases} p_v(d_{tu} - \|x_t - x_u\|), & \text{if } d_{tu} < R, \\ 0, & \text{otherwise.} \end{cases}$$  \hspace{1cm} (5)

where $d_{tu}$ represents the distance between nodes $t$ and $u$, $p_v$ the noise distribution of the measured distance, and $R$ the transmission radius. The more general model, which incorporates the probability of detection, can be found in [2, 10].

Due to the high complexity, the presence of nonlinear relationships, and potentially highly non-Gaussian uncertainties, we use nonparametric (particle-based) approximation of GBP-JT method (NGBP-JT). Moreover, due to the problems explained in previous sections, we are going to use PJT instead of JT. Therefore, in following subsections, we propose NGBP based on PJT (NGBP-PJT) for any arbitrary network. Before continuing, we recommend the reader to read [5], where we provided the analysis of GBP-JT and NGBP-JT for the small-scale network.

4.1 Drawing particles from the cliques

Let us draw $N_C$ weighted particles, $\{W_{C_k}^{k,m}, X_{C_k}^{k,m}\} (k = 1, ..., N_C, m = 1)$, from clique $i$. Since it’s computationally very expensive to draw particles from $M_i^1 = \psi_{C_i}$, we need to find appropriate importance density function. Thus, for the initial particles, we are going to use two constraints: i) each particle of the node must be inside its bounding box, and ii) the distance between each pair of the nodes in clique should be close to the mean value of the measured distance. Taking this into account, our importance density function $q_{C_i}^m$ ($m = 1$) for clique $i$, which includes nodes $t$ and $u$, is given by:

$$q_{C_i}^m(x_{C_i}) = q_{tu}(x_t, x_u) = \begin{cases} \psi_t(x_t)\psi_u(x_u), & \text{if } \|\mu_{tu} - \|x_t - x_u\|| < 2\sigma \\ 0, & \text{otherwise.} \end{cases}$$  \hspace{1cm} (6)

where $\mu_{tu}$ is the mean value of measured distance, and $\sigma$ is the standard deviation of the error distribution. The importance density for 3-node clique $j$, which includes nodes $t$, $u$, and $v$, is given by:

$$q_{C_j}^1(x_{C_j}) = \sqrt{q_{tu}^1(x_t, x_u)q_{tv}^1(x_t, x_v)q_{uv}^1(x_u, x_v)}$$  \hspace{1cm} (7)

To draw clique particle, we need to draw node particles within its boxes and accept the particle if the constraint is satisfied. If not, we reject the sample, and try again. The weights of the particles can be easily computed by $W_{C_i}^{k,1} = \psi_{C_i}(X_{C_i}^{k,1})/q_{C_i}^1(X_{C_i}^{k,1})$ and then normalized. In this way, we have created two types of particles: the edges (for 2-node cliques), and the triangles (for 3-node cliques). We illustrated initial set of particles in Fig. 3.
4.2 Computing messages

Having computed initial particles from beliefs, we can compute the particles from the messages. According to equation (3), we first need to marginalize the belief from previous iteration and then to divide the belief by the incoming message from previous iteration. Since all node particles within the clique have one common weight (e.g., \(\{W_{C_i}^{k,m} \cdot X_{C_i}^{k,m} \} = \{W_{C_i}^{k,m} \cdot X_{t}^{k,m}, X_{u}^{k,m}\}\)), we can simply pick the node which appears in the message (from clique that send the message), and compute the weight as reminder of (3). Since the separators sets can include one or two nodes, there exist 1-node and 2-node messages. For example, the weighted particles of the 1-node message from \(C_i(t, u)\) to \(C_j(t, v)\), at iteration \(m + 1\), are given by:

\[
\{X_{S_{ij}}^{k,m+1}, W_{S_{ij}}^{k,m+1}\} = \{X_t^{k,m}, m_{ij}^m(X_t^{k,m})\}
\]

(8)

The 2-node message can be found in same way. As we can see, we need the parametric form of the message \(m_{ij}^m\), so we estimate it using spherically symmetric Gaussian kernel [11]. For 2-node message, it is very expensive to estimate the parametric form directly from high-dimensional (4D) particles. However, there is dependency between the nodes within the message (noisy distance), so it can be written as:

\[
m_{ij}^m(x_t, x_u) = m_{ij}^m(x_t)\psi_{tu}(x_t, x_u)
\]

(9)

Finally, the messages from any anchor \(a\) to an unknown node \(t\) are simply given by parametric form:

\[
m_{at}(x_t) = \psi_{at}(X_t, x_t).
\]

4.3 Computing beliefs

According to (4), the belief of clique \(i\) is a product of its clique potential and all the messages coming into the clique. Before drawing particles, we need to solve two problems: i) the messages include information about different nodes within the clique, and ii) it is very expensive to draw samples from the product.

The first problem can be solved by filling the message with information about nodes which appear in destination clique, but not in the message. For example, for the message \(m_{ij}^{m+1}(x_t, x_u)\), from \(C_i(t, u, v)\) to \(C_j(t, u, r)\), we can form the joint message:

\[
M_{ij}^{m+1}(x_t, x_u, x_r) = m_{ij}^{m+1}(x_t, x_u)\psi_{tu}(x_t, x_r)\psi_{ru}(x_u, x_r).
\]

Using (9) and the definition of clique potential, the joint message can be written as:

\[
M_{ij}^{m+1}(x_C) = m_{ij}^{m+1}(x_t)\psi_{Cj}(x_C)
\]

(10)

where node \(t\) must be in appropriate separator set \((t \in S_{ij})\), and if \(|S_{ij}| > 1\) we can pick one node randomly. Taking this into account, it can be easily proved, that equation (10) is valid for any clique. Thanks to the particles from standard messages, we already have few (one or two) node particles from each joint message. The remained node particles can be drawn by shifting given node particles in random direction for an amount which represents the observed distance, and by checking (only in case of 3-node clique) another distance constraint (more details in [5]). Of course, the weights of the particles from joint messages are equal to the weights of the particles from standard messages. However, due to the sample depletion, we resample with replacement [2] so as to produce the particles with same weights: \(\{1/N, X_{i}^{k,m+1}\}\).

Finally, due to the problem ii), instead of product, we make the sum of joint messages (i.e., using mixture importance sampling (MIS) [2]). Therefore, the final importance density for the belief of clique \(j\), and corresponding particles, are respectively given by:

\[
q_{C_j}^{m+1}(x_C) = \sum_{i \in G_j} M_{ij}^{m+1}(x_C)
\]

(11)

\[
\{W_{C_i,j}^{k,m+1}, X_{C_i,j}^{k,m+1}\} = \frac{1}{|G_j| \cdot N_c} \bigcup_{i \in G_j} X_{ij}^{k,m+1}
\]

(12)

Finally, we can find the set of particles from the beliefs \(\{W_{C_j}^{k,m+1}, X_{C_j}^{k,m+1}\}\) (\(k = 1, ..., N_C\)):

\[
X_{C_j}^{k,m+1} = \text{choose}(X_{C_j}^{k,m+1}[G_j])
\]

(13)

\[
W_{C_j}^{k,m+1} = W_{C_j,corr}^{k,m+1} \cdot \psi_{Cj}(X_{C_j}^{k,m+1}) \prod_{c \in G_j} m_{at}(X_t^{k,m+1})
\]

(14)

\[
W_{C_j}^{k,m+1,corr} = W_{C_j}^{k,m+1} \frac{\prod_{c \in G_j} m_{ij}^{m+1}(X_{S_{ij}}^{k,m+1})}{q_{C_j}^{m+1}(X_{S_{ij}}^{k,m+1})}
\]

(15)

where \(W_{C_j}^{k,m+1,corr}\) is correction of the weights due to the MIS. \(X_{C_j}^{k,m+1}\) particle from node \(t\), and function \text{choose} chooses randomly one particle from \(|G_j|\).

The final estimates of the nodes within the cliques, are given as the mean of the beliefs in last iteration. Since the most of the nodes appear in more than one clique, we simply average multiple estimates.

5 Simulation Results

We placed \(N_a + N_u = 60\) nodes in 20m x 20m area. The minimum number of anchors, which are placed near the edges, is \(N_{a,min} = 3\). This realistic constraint helps the unknown nodes near the edges which suffer from low connectivity. The rest of the anchors and the unknowns are randomly deployed within the area. The number of iteration is set to \(N_{iter} = 3\), which means that any node/clique will receive all the information 3-hop away from itself. We assume that the distance is obtained using RSS measurements, so we choose the standard deviation \(\sigma_{dB} = 5\text{dB}\) (i.e., the parameter of log-normal
distribution of the distance is \( \sigma_{\log(d)} = \sigma_{dB}/10n_p = 0.25 \), where \( n_p \approx 2 \). All simulations are performed for \( N_c = 6 \) and \( N_a = 12 \) anchor nodes, and with respect to transmission radius \( R \), which varies from 5m to 12m. All previous parameters are same both for NBP and NGBP-PJT. However, the last parameter, the number of particles, is set to \( N_{nbp} = 100 \) and \( N_{pjt} = 210 \), so as to make the same computational time for both methods\(^2\). Finally, all simulations results represent the average over 20 random networks.

Using the defined scenario, we compared accuracy of NBP and NBP-PJT algorithms. The error is defined as Euclidean distance between true and estimated location. We illustrated the results of both methods in Fig. 4. It’s obvious that, for almost all unknown nodes, NBP-PJT method overperforms the NBP method. Regarding RMS error and coverage (the percentage of located nodes with error less than predefined tolerance) shown in Fig. 5 and Fig. 6, the NBP-PJT significantly overperforms NBP, for all \( R \) and both values of \( N_a \). It’s also worth noting that number of anchors significantly affects accuracy and coverage.

To measure the communication cost, we count elementary messages, where one elementary message is defined as simple scalar data (e.g., one coordinate of one particle). According to Fig. 7, the NGBP-PJT not only outperforms NBP, but also this improvement is increasing as transmission radius increases. This is achieved thanks to the thin graph, which ensures low communication cost, and makes it nearly constant with respect to \( R \). This feature provides us more precise information about battery life. Regarding number of anchors, the larger fraction of anchors decreases communication cost due to the simple message that each anchor has to transmit. However, it’s interesting that for \( R < 7 \), the conclusion is opposite. This is caused by our constraint that the network is always connected (i.e., there is a path between each pair of the nodes), which is realistic constraint in the most applications. Basically, when the network is disconnected (it mostly happened for \( R < 7 \), \( N_a = 12 \)), we replaced that network with connected one, and as a consequence increase the communication between nodes.

6 Conclusions and Future Work
In this paper, we presented NGBP-PJT, a novel probabilistic approach for cooperative localization in loopy networks.
networks. Since the exact formation of junction tree is not tractable, we proposed the formation of pseudo-junction tree (PJT), which represents the approximated junction tree based on thin graph. In addition, in order to decrease the number of particles for NGBP-PJT method, we used a set of very strong constraints. The resulting localization method, NGBP-PJT, outperforms NBP in terms of accuracy and communication cost in any arbitrary network. Moreover, thanks to the thin graph, NGBP-PJT has nearly constant communication cost with respect to transmission radius. There remain many open directions for the future work. A comparison with other BP-based methods for loopy networks could be very useful. We are currently working on two alternative solutions for this problem: NBP based spanning tree, and tree-reweighted NBP [12]. Furthermore, it’s important to check if there is some cheaper (non-particle-based) message representation, which should be capable to handle all realistic uncertainties. Finally, target tracking using this method could be an interesting direction.

Acknowledgments
This work is supported by the FPU fellowship from Spanish Ministry of Science and Innovation. Furthermore, we thank partial support by project ICT-217033 WHERE, program CONSOLIDER-INGENIO 2010 CSD2008-00010 COMONSENS and National Project M3HF.

References
A.17 Optimized Edge Appearance probability for Cooperative Localization based on Tree-Reweighted Nonparametric Belief Propagation

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

OPTIMIZED EDGE APPEARANCE PROBABILITY FOR COOPERATIVE LOCALIZATION BASED ON TREE-REWEIGHTED NONPARAMETRIC BELIEF PROPAGATION

Vladimir Savic, Henk Wymeersch, Federico Penna, Santiago Zazo
OPTIMIZED EDGE APPEARANCE PROBABILITY FOR COOPERATIVE LOCALIZATION 
BASED ON TREE-REWEIGHTED NONPARAMETRIC BELIEF PROPAGATION

Vladimir Savic\textsuperscript{1}, Henk Wymeersch\textsuperscript{2}, Federico Penna\textsuperscript{3}, Santiago Zazo\textsuperscript{1}

\textsuperscript{1}Universidad Politecnica de Madrid, Spain, Email: \{vladimir, santiago\}@gaps.ssr.upm.es
\textsuperscript{2}Chalmers University of Technology, Gothenburg, Sweden, Email: henk.wymeersch@ieee.org
\textsuperscript{3}Politecnico di Torino, Italy, Email: federico.penna@polito.it

ABSTRACT
Nonparametric belief propagation (NBP) is a well-known particle-based method for distributed inference in wireless networks. NBP has a large number of applications, including cooperative localization. However, in loopy networks NBP suffers from similar problems as standard BP, such as over-confident beliefs and possible non-convergence. Tree-reweighted NBP (TRW-NBP) can mitigate these problems, but does not easily lead to a distributed implementation due to the non-local nature of the required so-called edge appearance probabilities. In this paper, we propose a variation of TRW-NBP, suitable for cooperative localization in wireless networks. Our algorithm uses a fixed edge appearance probability for every edge, and can outperform standard NBP in dense wireless networks.

Index Terms— Cooperative localization, nonparametric belief propagation, tree-reweighted belief propagation, wireless networks.

1. INTRODUCTION

Cooperative localization is an important problem in wireless networks, as the availability of positional information can enable many applications, such as search-and-rescue, asset tracking, and indoor navigation. State-of-the-art algorithms for cooperative localization rely essentially on nonparametric belief propagation (NBP) \cite{1,3,5}. NBP is a variation on BP, which is a well-known, low-complexity inference method, with applications in many fields, including artificial intelligence, wireless communication, coding theory, computer vision, and cognitive networks. For continuous random variables, messages can often not be computed or represented exactly. NBP was proposed in \cite{1} to address this problem through a particle-based representation. Both BP and NBP can be interpreted as message passing algorithms on a graphical model, and are inherently distributed, thus lending themselves well to distributed inference problems, such as cooperative localization.

For most applications, the main problem of NBP (and also BP) is that in graphs with cycles, there are no guarantees on the quality of the marginal beliefs, nor on the convergence of message passing. Solutions to this problem include generalized belief propagation (GBP) \cite{2}, NBP over spanning trees (NBP-ST) \cite{3}, and tree-reweighted belief propagation (TRW-BP) \cite{4}. GBP is able to improve on standard BP, but at high computational cost, making it unsuitable for large-scale networks. NBP-ST performs NBP on two (or more) spanning trees, outperforming NBP only in highly connected networks, with small performance gains. Finally, TRW-BP considers distributions over all spanning trees, through a concept known as edge appearance probabilities. TRW-BP has the potential to outperform NBP, but relies on the availability of valid edge appearance probabilities, which are themselves hard to obtain in a distributed manner. Hence, TRW-BP in its original form is not suitable for a distributed implementation.

In this paper, we propose a variation of tree-reweighted NBP for cooperative localization in wireless networks, where we (i) consider uniform edge appearance probabilities (characterized by the scalar variable $\rho$); (ii) allow invalid edge appearance probabilities (i.e., not corresponding to any distribution over spanning trees). We evaluate the proposed method as a function of $\rho$ in terms of two performance metrics: the root-mean square error (RMSE) of the position estimate and the Kullback-Leibler divergence (KLD) of the marginal belief with respect to the true marginal posterior. We propose an empirical function for the optimum edge appearance probability that minimizes the errors (in terms of KLD and RMSE) caused by loops.

The remainder of this paper is organized as follows. In Section 2, we describe cooperative localization using TRW-NBP. An empirical approach for finding $\rho$ is presented in Section 3. Finally, Section 4 provides some conclusions and suggestions for future work.

2. COOPERATIVE LOCALIZATION BASED ON TREE-REWEIGHTED NONPARAMETRIC BELIEF PROPAGATION

2.1. The Localization Problem
Consider $N_a$ anchors and $N_t$ targets scattered randomly in a planar region, and denote the two-dimensional (2D) location of node $t$ by $x_t$. The target $u$ obtains a noisy measurement $d_{tu}$ of its distance from node $t$ with some probability $P_d(x_t,x_u)$:

$$d_{tu} = \|x_t - x_u\| + v_{tu}, \quad v_{tu} \sim p_v. \quad (1)$$

For simplicity, we assume the noise $v_{tu}$ has a zero-mean Gaussian distribution, and ideal model for probability of detection:

$$P_d(x_1,x_u) = \begin{cases} 1, & \text{for } \|x_t - x_u\| \leq R, \\ 0, & \text{otherwise}. \end{cases} \quad (2)$$

where $R$ represents transmission radius. We will indicate with the binary variable $\alpha_{tu}$ whether an observation is available ($\alpha_{tu} = 1$) or not ($\alpha_{tu} = 0$). Finally, each node $t$ has some prior distribution denoted $p_t(x_t)$. The joint distribution is given by:

$$p(x_1,\ldots,x_{N_t},\{\alpha_{tu}\},\{d_{tu}\}) = \prod_{(t,u)} p(\alpha_{tu}|x_t,x_u) \prod_{(t,u)} p(d_{tu}|x_t,x_u) \prod_{t} p_t(x_t). \quad (3)$$
Our objective is to compute (or approximate) the marginal beliefs $p(x_t | \{o_{tu}, d_{tu}\})$, for every target $t$. Then, we can easily estimate the positions, e.g., as mean values of these marginals.

2.2. Graph Representation

The relationship between a graphical model and a distribution $q(x_1, ..., x_{N_t})$ may be represented in terms of potential functions $\psi$ which are defined over graph’s cliques. A clique ($C$) is a subset of nodes such that for every two nodes in $C$, there exists a link connecting the two. So the joint distribution is given by:

$$q(x_1, ..., x_{N_t}) \propto \prod_{\text{cliques } C} \psi_C(\{x_i : i \in C\}).$$

We can now define potential functions which can express the belief equations and message-update rule of TRW-BP are, respectively.

On the right-hand side, there is a product over all reweighted messages ($\rho_{tu}$). According to the product of the local evidence at that node $P_{\text{local evidence}} \propto \prod_{\text{neighbors of node } x_i} \psi_{\text{local evidence}}(x_i)$. We note that this is uniform over $\{\rho_{tu} \}$, i.e., $\psi_{\text{local evidence}}(x_i)$.

Finally, the joint posterior distribution is given by:

$$p(x_1, ..., x_{N_t} | \{o_{tu}, d_{tu}\}) \propto \prod_t \psi_1(x_t) \prod_{t,u} \psi_{\text{local evidence}}(x_t, x_u).$$

By marginalizing this joint distribution, we can find the true belief of each node. Exact marginalization is intractable, which is why we resort to near-optimal message-passing methods.

2.3. Tree-Reweighted Belief Propagation (TRW-BP)

In the standard TRW-BP algorithm the belief at a node $t$ is proportional to the product of the local evidence at that node $\psi(x_t)$, and all reweighted messages coming into node $t$:

$$M_t(x_t) \propto \psi(x_t) \prod_{u \in G_t} m_{ut}(x_t)^{\rho_{tu}},$$

where $x_t$ is a state of node $t$, $\rho_{tu} = \rho_{tu}$ is the appearance probability of the edge $(t, u)$, and $G_t$ denotes the neighbors of node $t$. The messages are determined by the message-update rule:

$$m_{ut}(x_t) \propto \int \psi_u(x_u) \psi_{\text{local evidence}}(x_t, x_u)^{1/\rho_{tu}} \prod_{k \in G_u \setminus t} \frac{m_{ku}(x_u)^{\rho_{ku}}}{m_{ku}(x_u)^{\rho_{ku}} - \rho_{ku}} \, dx_u,$$

where $\psi_u(x_u)$ is the pairwise potential between nodes $t$ and $u$. On the right-hand side, there is a product over all reweighted messages going into node $u$ except for the one coming from node $t$. The update-rule (8) is carried out over the network. Upon convergence, the beliefs are computed through (7). In practice, it is more convenient to compute the beliefs at every iteration $i$. This leads to an equivalent form of TRW-BP: by replacing (7) in (8), we find that the belief equations and message-update rule of TRW-BP are, respectively, given by:

$$M_{i}^{(t)}(x_t) \propto \psi_t(x_t) \prod_{u \in G_t} m_{ut}^{(t)}(x_t)^{\rho_{tu}}$$

and

$$m_{ut}^{(t)}(x_t) \propto \int \psi_u(x_u) \psi_{\text{local evidence}}(x_t, x_u)^{1/\rho_{tu}} \prod_{k \in G_u \setminus t} \frac{m_{ku}^{(t)}(x_u)^{\rho_{ku}}}{m_{ku}^{(t)}(x_u)^{\rho_{ku}} - \rho_{ku}} \, dx_u.$$
3.1. A 4-Node Clique

We consider fully-connected network with 4 targets in 1D space (see Figure 1a for 2D case). In addition, there are 4 anchor nodes (not depicted), each of them connected exactly to one target. Our goal is to estimate the true belief, TRW-NBP beliefs and estimated locations. The latter are given by the minimum mean square error (MMSE) estimate from the belief. We run TRW-NBP for different values of $\rho$ and, for each result, we compute KLD between true and TRW-NBP beliefs, and RMSE of estimated locations, all shown in Figure 2. According to Figure 2, we can make the following conclusions:

1. Both RMSE and KLD reach the minimum for the same $\rho < 1$. That means that it is sufficient to use only RMSE for learning the optimal $\rho$ in larger networks, where the computation of true beliefs (necessary for computing KLD) is intractable.

2. The optimal $\rho$ ($\rho_{\text{opt}}$) is 0.5, which is the same as the theoretical value (Figure 1b), under a uniform distribution over spanning trees. NBP ($\rho = 1$) performs worse than optimum TRW-NBP in terms of both KLD and RMSE. For a comparison between the three different beliefs, see Figure 3.

3. A wide range of $\rho$ (in our example, 0.4-1) provides better performance than NBP in terms of both KLD and RMSE. That means that we can even use a coarse approximation of $\rho_{\text{opt}}$.

4. The RMSE is rather insensitive to $\rho$, for $\rho > \rho_{\text{opt}}$. Hence, care needs to be taken when interpreting RMSE figures as a function of $\rho$, as the effect on KLD may be much more pronounced.\footnote{This could be a problem for learning in larger networks, where it is practically impossible to obtain smooth curves. However, we always use only confident digits (by rounding-up RMSE) and, in case of more minimums, we use the minimum corresponding to the lowest value of $\rho$ (e.g., see Figure 5a). This approximation still keeps KLD quite close to minimum.}

Taking these conclusions into account, we now move on larger networks.

3.2. Grid and Random Topology Networks

We consider a network with 25 target nodes and 4 anchors in a 20m wide deployment area. We consider different values of the communication range \( R \), and the edge appearance probability $\rho$.

For the grid topology (where the distance between neighboring nodes is 0.6 m), Figure 4a shows the RMSE as a function of $\rho$, with

\[ \rho_{\text{opt}}(n_d) = \rho_0 \cdot e^{-k_p n_d}, \]

where parameters $\rho_0 = 3.187$ and $k_p = 0.199$ are found using least-square fitting. We did the same test for random topology (Figure 5), and obtained: $\rho_0 = 2.656$ and $k_p = 0.161$. Note that for random topology, it is harder to obtain sufficient statistics (Figure 5a), so the fitting is less confident compared with the grid topology. In any case, we conclude the following:

1. The difference between coefficients for random and grid topology is small, which means that the value of $\rho_{\text{opt}}$ depends more on the average node degree than the particular network configuration.

2. Though tempting to state that choosing $\rho = 1$ will lead to similar performance as $\rho = \rho_{\text{opt}}$, due to the almost flat curves for $\rho > \rho_{\text{opt}}$, this statement is not true when the performance is measured in terms of KLD (see Figure 2).

As an aside, when $n_d$ becomes very small, the fitted value for $\rho_{\text{opt}}$ can be larger than 1. This is merely a side-effect of the fitting. In practice, when $\rho_{\text{opt}} > 1$, one should set $\rho_{\text{opt}} = 1$.

Finally, it is worth noting that $n_d$ can easily and quickly be found using a consensus algorithm \[8\]. In that case, the computational/communication cost will be nearly the same as for NBP.
4. CONCLUSIONS

We presented a cooperative localization algorithm based on tree-reweighted nonparametric belief propagation (TRW-NBP), which combines the distributed nature of NBP and the improved performance of TRW-BP. In contrast to TRW-BP, we propose to use a constant edge appearance probability ρ. In contrast to NBP, we propose to set ρ < 1. Through Monte Carlo simulations, we have verified performance gains in terms of RMSE and KLD w.r.t. the true distribution. We have found that (i) the optimal ρ_{opt} does not depend on the particular criterion (RMSE or KLD); (ii) ρ_{opt} decreased in networks with more loops; (iii) ρ_{opt} can be expressed as a simple function of the average node degree (nd). Our future work will focus on the extension to 2D and 3D space. Preliminary results (not reported in this paper) indicate similar behavior of ρ_{opt} as a function of nd. Our final goal is the implementation and evaluation of different variations of belief propagation on wireless motes.

5. REFERENCES


References


[9] FP7-ICT-2009-4 WHERE2 D1.1 Scenarios and Parameters LINK NOT AVAILABLE YET!


[11] WHERE1 D2.4 Performance of WHERE cooperative positioning techniques


