

On Localization Issues of Mobile Devices

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades

“Doctor rerum naturalium”

der Georg-August-Universität Göttingen

im Promotionsprogramm Computer Science (PCS)

der Georg-August University School of Science (GAUSS)

vorgelegt von

Yali Yuan

aus China

Göttingen, November 2018

Betreuungsausschuss

Prof. Dr. Dieter Hogrefe,
Telematics Group, Institut für Informatik

Prof. Dr. Xiaoming Fu,
Computer Networks Group, Institut für Informatik

Mitglieder der Prüfungskommission

Referent: Prof. Dr. Dieter Hogrefe,
Telematics Group, Institut für Informatik

Korreferent: Prof. Dr. Xiaoming Fu,
Computer Networks Group, Institut für Informatik

Weitere Mitglieder der Prüfungskommission

Prof. Dr. Jens Grabowski,
Software Engineering for Distributed Systems Group, Institut für Informatik

Prof. Dr. Carsten Damm,
Theoretical Computer Science Group, Institut für Informatik

Prof. Dr. Marcus Baum,
Data Fusion Group, Institut für Informatik

Prof. Dr. Winfried Kurth,
Computer Graphics and Ecological Informatics Group, Institut für Informatik

Tag der mündlichen Prüfung

30. August 2018

Statement

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules. In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [University of Goettingen]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Göttingen, November 2018

Abstract

Mobile devices, such as sensor nodes, smartphones and smartwatches, are now widely used in many applications. Localization is a highly important topic in wireless networks as well as in many Internet of Things applications. In this thesis, four novel localization schemes of mobile devices are introduced to improve the localization performance in three different areas, like the outdoor, indoor and underwater environments.

Firstly, in the outdoor environment, many current localization algorithms are based on the Sequential Monte Carlo Localization (MCL), the accuracy of which is bounded by the radio range. High computational complexity in the sampling step is another issue of these approaches. Tri-MCL is presented, which significantly improves on the accuracy of the Monte Carlo Localization algorithm. To do this, three different distance measurement algorithms based on range-free approaches are leveraged. Using these, the distances between unknown nodes and anchor nodes are estimated to perform more fine-grained filtering of the particles as well as for weighting the particles in the final estimation step of the algorithm. Simulation results illustrate that the proposed algorithm achieves better accuracy than the MCL and SA-MCL algorithms. Furthermore, it also exhibits high efficiency in the sampling step.

Then, in the Global Positioning System (GPS)-denied indoor environment, Twi-Adaboost is proposed, which is a collaborative indoor localization algorithm with the fusion of internal sensors such as the accelerometer, gyroscope and magnetometer from multiple devices. Specifically, the datasets are collected firstly by one person wearing two devices simultaneously: a smartphone and a smartwatch, each collecting multivariate data represented by their internal parameters in a real environment. Then, the datasets from these two devices are evaluated for their strengths and weaknesses in recognizing the indoor position. Based on that, the Twi-AdaBoost algorithm, an interactive ensemble learning method, is proposed to improve the indoor localization accuracy by fusing the co-occurrence information. The performance of the proposed algorithm is assessed on a real-world dataset. The experiment results demonstrate that Twi-AdaBoost achieves a localization error about 0.39 m on average with a low deployment cost, which outperforms the state-of-the-art indoor localization algorithms.

Lastly, the characteristics of mobile Underwater Sensor Networks (UWSNs), such as low communication bandwidth, large propagation delay, and sparse deployment, pose

challenging issues for successful localization of sensor nodes. In addition, sensor nodes in UWSNs are usually powered by batteries whose replacements introduce high cost and complexity. Thus, the critical problem in UWSNs is to enable each sensor node to find enough anchor nodes in order to localize itself, with minimum energy costs. An Energy-Efficient Localization Algorithm (EELA) is proposed to analyze the decentralized interactions among sensor nodes and anchor nodes. A Single-Leader-Multi-Follower Stackelberg game is utilized to formulate the topology control problem of sensor nodes and anchor nodes by exploiting their available communication opportunities. In this game, the sensor node acts as a leader taking into account factors such as 'two-hop' anchor nodes and energy consumption, while anchor nodes act as multiple followers, considering their ability to localize sensor nodes and their energy consumption. I prove that both players select best responses and reach a socially optimal Stackelberg Nash Equilibrium. Simulation results demonstrate that the proposed EELA improves the performance of localization in UWSNs significantly, and in particular the energy cost of sensor nodes. Compared to the baseline schemes, the energy consumption per node is about 48% lower in EELA, while providing a desirable localization coverage, under reasonable error and delay. Based on the EELA scheme, an Adaptive Energy Efficient Localization Algorithm using the Fuzzy game theoretic method (Adaptive EELA) is proposed to solve the environment adaptation problem of EELA. The adaptive neuro-fuzzy method is used as the utility function of the Single-Leader-Multi-Follower Stackelberg game to model the dynamical changes in UWSNs. The proposed Adaptive EELA scheme is able to automatically learn in the offline phase, which is required only once. Then, in the online phase, it can adapt to the environmental changes, such as the densities of nodes or topologies of nodes. Extensive numerical evaluations are conducted under different network topologies and different network node densities. The simulation results demonstrate that the proposed Adaptive EELA scheme achieves about 35% and 66% energy reduction per node on average comparing the state-of-the-art approaches, such as EELA and OLTC, while providing a desirable localization coverage, localization error and localization delay.

Acknowledgements

I would like to express my utmost gratitude and appreciation to the people who supported me in my work on this thesis.

First and foremost, I sincerely thank my first supervisor Prof. Dr. Dieter Hogrefe for his patient guidance, support and encouragement throughout my graduate studies. He is very nice and knowledgeable Professor. He created an enjoyable working environment for us that stimulates original thinking and initiative. It has been an honor to be his PhD student. I appreciate all his contributions of time, suggestions and funding to make my PhD experience productive.

I would like to express my gratitude to Prof. Dr. Xiaoming Fu and Dr. Parisa Memar-moshrefi, who contributed to many discussions that helped to find the right structure for the work. They always gave me valuable encouragement and feedback. Furthermore, I want to thank the thesis committee Prof. Dr. Dieter Hogrefe, Prof. Dr. Xiaoming Fu, Prof. Dr. Marcus Baum, Prof. Dr. Jens Grabowski, Prof. Dr. Carsten Damm and Prof. Dr. Winfried Kurth for investing their valuable time.

In addition, I want to thank all my co-authors, Prof. Dr. Dieter Hogrefe, Associate Prof. Dr. Megumi Kaneko, Prof. Dr. Xu Chen, Arne Bochem, Chencheng Liang, Christian Melching and Yachao Yuan. Without their help and contributions, I can not achieve these research work.

The members of the Telematics group have contributed immensely to my personal and professional time at Goettingen. The group has been a source of friendships as well as good advice and collaboration. They are good discussion partners and give me suggestions to see things also from other points of view. They always help me when I need help.

I gratefully acknowledge the funding sources that made my PhD work possible. The China Scholarship Council provides me three years scholarship to support my PhD study.

Lastly, I would like to thank my family for all their love, support and encouragement. My parents raised me with a love of science and supported me in all my pursuits. My sisters and brothers always support me to achieve my dream. And most of all for my loving, encouraging, and patient husband Yang Liu whose faithful support during the final stages of this PhD is so appreciated.

Contents

1	Introduction	2
1.1	Overview of Localization with Mobile Devices in Wireless Environment	2
1.2	Scope of the Thesis	3
1.3	Organization	3
2	Outdoor Localization with Wireless Sensor Networks	6
2.1	Motivation	6
2.2	Contributions	7
2.3	State of the Art	7
2.4	Localization Scheme	9
2.4.1	Symbols	9
2.4.2	Initialization	9
2.4.3	Range-Free Distance Estimation	9
2.4.4	Prediction	13
2.4.5	Filtering in Monte Carlo Localization (MCL)	13
2.4.6	Filtering in Tri-MCL	14
2.4.7	Position Estimation	15
2.5	Evaluation	17
2.5.1	Results	18
2.6	Summary and Future Work	19
3	Indoor Localization with Mobile Devices	20
3.1	Motivation	20
3.2	Contributions	21
3.3	State of the Art	22
3.4	Methodology	23
3.4.1	Building a Weak Learner Based on the CART Algorithm	23

3.4.2	AdaBoost.R2 Regression Model	25
3.4.3	Two-AdaBoost Fusion Strategy	25
3.5	Experiments	29
3.5.1	Dataset Analysis	29
3.5.2	Dataset Preprocessing	29
3.5.3	Performance Metrics	31
3.5.4	General Results and Analysis	32
3.5.5	Comparison Results and Analysis	35
3.6	Summary and Future Work	38
4	Underwater Localization with Wireless Sensor Networks	40
4.1	Motivation	40
4.2	Contributions	41
4.3	State of the Art	43
4.3.1	Localization in Underwater Sensor Networks (UWSNs)	43
4.3.2	Topology Control	43
4.4	System Model	44
4.4.1	System Overview	44
4.4.2	Propagation Model	45
4.5	Underwater Localization by Stackelberg Game Theory	47
4.5.1	Problem Formulation and Solution	47
4.5.2	Numerical Evaluations	62
4.6	Underwater Localization by Adaptive Fuzzy Stackelberg Game Theory	71
4.6.1	Problem Formulation	71
4.6.2	Proposed Adaptive EELA Model	81
4.6.3	Numerical Evaluations	82
4.7	Summary and Future Work	92
5	Conclusion and Future Work	94

Chapter 1

Introduction

1.1 Overview of Localization with Mobile Devices in Wireless Environment

Positioning or localization systems estimate the location of a person or an object with the help of the Global Positioning System (GPS) or other sensory information collected by mobile devices, such as smartphones, smartwatches or sensors. Localization systems are crucial in many applications, such as habitat monitoring [1], animal tracking [2] and vehicle tracking [3] for outdoor environment; medical assistance (patient tracking) [4], elderly care (aged pedestrian tracking) [4] and underground mining safety [5] for indoor situation; aquatic environment monitoring, target tracking [6], and pollution control for underwater environment. The requirement for highly-accurate, low-power and low-cost localization schemes for different applications under different environments have motivated me to further improve the current localization performance. In this thesis, I would like to answer several questions. How to improve the performance of current localization algorithms to achieve a higher accuracy in outdoor or indoor environment? How to improve the localization energy efficiency without affecting some other aspects of localization performance, like coverage, localization accuracy and localization delay in UWSNs?

1.2 Scope of the Thesis

In this thesis, I propose four novel localization algorithms, i.e., Tri-MCL, Twi-AdaBoost, EELA and Adaptive EELA, in order to improve the performance of the localization systems. The proposed Tri-MCL algorithm [6] and Twi-AdaBoost [7] method were published in IEEE 41st Conference on Local Computer Networks (LCN) in 2016 and Journal of IEEE Access in 2018, respectively. The proposed EELA algorithm [8] is under the second review in Journal of IEEE Transactions on Vehicular Technology and I will submit one paper related to the Adaptive EELA algorithm soon. The four proposed algorithms are used in three different environments, like indoor, outdoor or underwater scene. In different localization scenarios, different contents, such as the motivation, the state of the art, the proposed algorithm and evaluation results are presented. Hence, I introduce them in detail separately in this thesis.

1.3 Organization

The thesis is structured as follows. Chapter 2 presents one novel localization algorithm for mobile sensor networks and IoT applications, called Tri-MCL. In detail, I introduce the motivation and contribution of Tri-MCL in Section 2.1 and 2.2, respectively. Section 2.3 reviews related works of existing MCL-based algorithms. In Section 2.4, the proposed Tri-MCL scheme is described. Simulation results are shown in Section 2.5. Finally, I draw conclusions in Section 2.6.

In Chapter 3, a multi-device context indoor localization algorithm, named Twi-AdaBoost, which uses several kinds of sensors in both smart-phone and smart-watch to pursue the optimal utilization of sensors from multiple devices is proposed. The motivation and contribution of the proposed Twi-Adaboost are firstly introduced in Section 3.1 and 3.2. Then, in Section 3.3, the state of the arts are reviewed. Section 3.4 presents the proposed Twi-AdaBoost fusion strategy and its knowledge background. The datasets analysis and pre-processing, experimental results as well as performance evaluations are introduced in Section 3.5. Finally, Section 3.6 summarizes the Chapter 3.

An Energy-Efficient Localization Algorithm (EELA) and Adaptive EELA which are proposed to analyze the decentralized interactions among sensor nodes and anchor nodes in mobile UWSNs, are discussed in Chapter 4. In detail, Section 4.1 and 4.2 gives the description about the motivation and contributions. Section 4.3 discusses the related

works. Section 4.4 introduces the system model, which consists of the system overview and propagation model. The detailed description of the proposed EELA model is presented in Section 4.5, which is composed of the problem formulation and solution as well as the numerical evaluations of EELA. Based on the EELA scheme, an Adaptive Energy-Efficient Localization Algorithm (Adaptive EELA) using the fuzzy game theoretic method is proposed to solve the environment adaptation problem of EELA, which is introduced in Section 4.6 including the detailed contents of the proposed Adaptive EELA, such as the problem formulation, the description of Adaptive EELA and numerical evaluations. Finally, Section 4.7 presents the summary and future work.

Chapter 5 gives the conclusion and future work for all the proposed algorithms in this thesis.

Chapter 2

Outdoor Localization with Wireless Sensor Networks

2.1 Motivation

Node localization plays an important role in Wireless Sensor Networks (WSNs) and Internet of Things (IoT) applications since it is not only useful in many basic network applications but also necessary in network operation. Examples are applications such as habitat monitoring [1], animal tracking [2], vehicle tracking [3], and environment monitoring [9], as well as network operation methods such as location-based routing protocols saving significant energy by eliminating the need for route discovery [10–12] [6].

Global Positioning System (GPS) is the straightforward solution for sensor node localization. However, it has disadvantages such as high cost, high power use and no indoor operation. One reasonable solution is that only a small proportion of sensor nodes is equipped with a GPS module and the rest get their positions through another localization scheme. The sensor nodes equipped with a GPS are called seeds or anchors. Many localization algorithms have been proposed not only for static sensor networks [13–16], but also for mobile sensor networks [17–20] in the past several years [6].

A popular representative of localization algorithms for mobile sensor networks is MCL [18]. The key idea of MCL is that the positions of sensor nodes are determined by a set of weighted samples and each sample, usually called particle, represents a possible location of the node. The most important contribution of MCL is that it is especially designed for mobile WSNs, i.e. all nodes including anchors are allowed to move arbitrarily during network operation time. However, the sampling phase and filtering phase need to be

repeated in order to obtain each particle, so it always suffers from high computational cost which will shorten the network life time significantly [6].

2.2 Contributions

The contributions of the proposed Tri-MCL algorithm are listed as follows [6],

- 1) The design and an evaluation of a new algorithm, called Tri-MCL, are presented for mobile sensor networks and IoT applications.
- 2) In order to improve the localization accuracy and sampling efficiency, Tri-MCL employs three different, synergistic distance measurements based on range-free methods and historical information to measure the distances between sensor nodes and anchor nodes. These distances are then used for filtering and weighting the particles in a more precise manner in the final estimation step of the algorithm.
- 3) Tri-MCL is an interactive process operating over multiple distance estimation values to form a consolidated fusion by interactively exploiting the synergies in these range-free distance measurement approaches, which is the key difference from the traditional MCL approach.
- 4) Numerical evaluation results prove that the proposed Tri-MCL works well compared with the baseline schemes.

The structure of this chapter is organized as follows: Section 2.3 reviews related works of existing MCL-based algorithms. In section 2.4, I describe the proposed Tri-MCL scheme. Simulation results are shown in Section 2.5. Finally, I draw conclusions in section 2.6.

2.3 State of the Art

Many localization algorithms have been designed for mobile sensor networks [18,19,21–27] [6].

In 2004, the Monte Carlo method is firstly introduced by L. Hu and D. Evans for localizing nodes in wireless sensor networks, called MCL [18]. MCL-based localization represents the posterior distribution of a node's location by a set of weighted particles, and in each time unit, the particles are updated based on new observations about beacons from anchor nodes. The authors proposed the localization method for a general network

environment where nodes and anchors can move arbitrarily. It does not require any additional hardware. This makes the approach suitable for both mobile ad hoc sensor networks and IoT applications. The main drawback of the MCL algorithm is that it has to redo the sampling and filtering phases for each particle. Usually, it will iterate many times in order to obtain enough valid particles, which is really time consuming. This makes it less suitable for sensor networks with limited computational abilities. The values of the particle weights are only 1 or 0, making this part of the algorithm coarse-grained [6].

In [19], A. Baggio and K. Langendoen proposed the Monte Carlo localization Boxed (MCB) algorithm. The sampling area was generated by building boxes in the intersection of the anchor nodes' communication scope, which improves the time efficiency significantly in the prediction phase of MCL. However, when the particle number of MCB equals to that of MCL, the localization error will not be improved. Even worse is that the number of valid particles will increase with the increasing number of the anchor nodes. As a result, the set of valid particles will be much larger than necessary for estimating a node's location [6].

M. Rudafshani and S. Datta [25] proposed the MSL*, MSL to improve the filtering phase of MCL using the location information of sensor nodes within two hops, but the additional communication was needed to pass samples or accuracy information. S. Hartung et al. [28] proposed the Sensor-Assisted Monte Carlo Localization (SA-MCL) method to solve the problem of temporary connection loss to anchor nodes due to changing network topologies. They used three different additional sensors to estimate the localization of sensor nodes. In [23], SMC was proposed to improve the localization accuracy by using the Angle of Arrival (AoA) measurements. Another range-based scheme [22] assumed that the distance or angle between anchor nodes and sensor node can be measured based on signal measurements such as Received Signal Strength Indication (RSSI), Time of Arrival (ToA), or AoA. However, the authors in [22, 23, 28] all need additional hardware support to improve the accuracy or solve problems of MCL [6].

In [29], weighted MCL (WMCL) was proposed. WMCL can improve the localization accuracy and sampling efficiency with low anchor densities, but the communication cost is much higher than for the original MCL algorithm. The RDMCL method was proposed in [30], which is based on the Received Signal Strength (RSS), distance and direction of the moving anchor nodes and MCL. RDMCL used three methods based on the number of nodes' one-hop neighbor anchors to build a more effective sampling area. The authors in [31] proposed a Weighted Monte Carlo Localization based on the Smallest Enclosing

Circle algorithm to solve the localization problem of node mobility in IoT scenarios. This algorithm generates the smallest enclosing circle of anchor nodes by using the hop counts from anchor nodes [6].

2.4 Localization Scheme

In this section, the proposed localization scheme, Tri-MCL, is described in detail. There are three phases in Tri-MCL: initialization phase, sampling phase and filtering phase. However, in Tri-MCL, the re-sampling phase is simplified, which effectively reduces the computational cost. The Tri-MCL filtering phase is also different from that of traditional MCL-based algorithms, which are given as,

- 1) Instead of using only the radio range of anchors to do the filtering, I use ring areas with three different distances around anchors as the filter area to filter particles. This helps to improve the localization accuracy.
- 2) Each particle has a different contribution to the final position estimate of the sensor node, as I weight each particle using a distance error penalty and a range free based distance estimation method failure penalty.

2.4.1 Symbols

For reference purposes, I list all symbols used in the following description of the proposed Tri-MCL scheme in Table 2.1.

2.4.2 Initialization

Before the start of the simulation, Tri-MCL is initialized with a set of N particles, distributed randomly over the simulation area.

2.4.3 Range-Free Distance Estimation

The distances estimated between sensor nodes and anchor nodes are employed to aid Tri-MCL to improve the performance of the location model. For this reason, three different schemes with different strengths are employed to estimate distances leading to the

Symbol	Meaning
V_{Max}	Maximum possible node speed also the definition of radius around nodes
e_t	HistDR position estimate at time t
L_t	Set of particles at time t
N	Number of particles in L_t
p_k	k -th particle
A	Set heard anchor nodes
a_i	i -th heard anchor node
ϕ	Number of distance estimation methods
$r_{\text{RingWidth}}$	Filtering ring width parameter
$r_{j,i}$	Distance estimate to anchor i according to method j
$d(a, b)$	Distance between points or particles a and b
$p_{i,k}$	Copy of p_k for anchor a_i
σ	$\sigma > 0$, used to avoid dividing by zero
α	Weighting factor $\alpha = 0.75$
$\omega_{\text{Fail},i,k}$	Penalty factor for $p_{i,k}$, relating to number of failed methods
$\omega_{\text{Range},i,k}$	Penalty factor for $p_{i,k}$, relating how well the distance estimate matches
$\omega_{i,k}$	Weighting factor for $p_{i,k}$
(x_t, y_t)	Final position estimate at time t

Table 2.1: Table of symbols, ©2016 IEEE.

synergistic qualities of Tri-MCL. Tri-MCL does not require the extra cost of the hardware due to all of the different schemes based on range-free technology. For a given situation, if the range estimates made with methods that are unsuitable, each method will return a failure state to penalize this situation. Otherwise, a good result will be returned.

2.4.3.1 DVH-Dist

Based on the the principles of the well known DV-Hop localization algorithm [32], a distance estimation algorithm named DVH-Dist is proposed in Tri-MCL. The steps of DV-Hop are listed as,

- 1) The minimum hop count from anchor nodes to sensor nodes are determined.
- 2) The distance between the sensor nodes and anchor nodes are computed by multiplying the minimum hop count and average distance of each hop.

- 3) The node estimates its position through triangulation algorithm or maximum likelihood estimators.

It is interesting to notice that I do not use the the costly calculations required to calculate the positions from the distance estimates in DV-Hop, because only the distances are relevant in the context of Tri-MCL. DVH-Dist is a simplification method of DV-Hop.

This approach works well in scenarios with a high density of sensor nodes, the same with that of DV-Hop, so that fine grained multi-hop distance estimates to a smaller number of anchor nodes can be made.

This method returns a failure state in a given situation, such as no seed nodes having been heard, even indirectly.

2.4.3.2 Cent-Dist

Based on the Centroid localization scheme, Cent-Dist works by calculating positions of nodes according to [33]. In Cent-Dist, a location estimate on sensor nodes is calculated by the center position between the received anchor node beacons, where very little computational costs results from calculating this position estimate.

For Cent-Dist, this position estimate is calculated and used to determine estimated distances towards all known anchor nodes (i.e. also those received through DVH-Dist flooding). This approach works best, when there are multiple anchor nodes in the immediate vicinity of the sensor node attempting to calculate its position.

Cent-Dist returns a failure state in a given situation, such as less than one direct seed (anchor) node having been heard.

2.4.3.3 HistDR-Dist

Historical Dead Reckoning Distance (HistDR-Dist) is the final method employed in Tri-MCL to calculate range estimates. For HistDR-Dist, the steps are given as,

- 1) The last three position estimates made by Tri-MCL are stored.
- 2) By using the stored information, an estimate of the current acceleration and angular acceleration of the node can be derived.

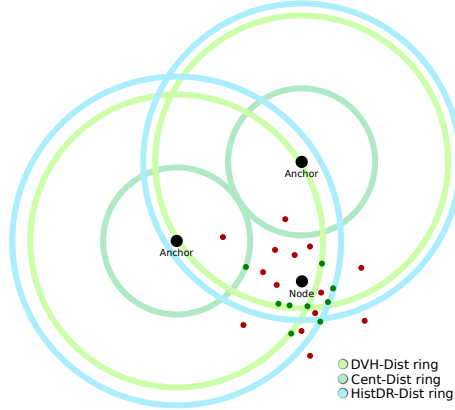


Figure 2.1: Sample situation with two anchor nodes and one sensor nodes. Red are filtered particles and green are admissible particles, ©2016 IEEE.

- 3) Using these values, by means of dead reckoning, the current position of the node is estimated by HistDR-Dist.

The mathematical description of HistDR-Dist is given in the following part. Let $e_{t-1} = (x_{t-1}, y_{t-1})$ be the previous estimate generated by Tri-MCL, $e_{t-2} = (x_{t-2}, y_{t-2})$ the one before that and so on. Using these values, HistDR will estimate the current position, which can then be used to determine a distance estimate $e_t = (x_t, y_t)$,

$$v_{t-1} = d(e_{t-1}, e_{t-2}), v_{t-2} = d(e_{t-2}, e_{t-3}), \quad (2.1)$$

$$\gamma_{t-1} = \angle(e_{t-1}, e_{t-2}), \gamma_{t-2} = \angle(e_{t-2}, e_{t-3}), \quad (2.2)$$

$$\Delta v = v_{t-1} - v_{t-2}, \Delta \gamma = \gamma_{t-1} - \gamma_{t-2}, \quad (2.3)$$

$$x_t = x_{t-1} + (v_{t-1} + \Delta v) \cos(\gamma_{t-1} + \Delta \gamma), \quad (2.4)$$

$$y_t = y_{t-1} + (v_{t-1} + \Delta v) \sin(\gamma_{t-1} + \Delta \gamma). \quad (2.5)$$

This approach can give good results if the previous estimates are reasonably accurate. It is not reliant on other nodes for the current time step, so it bridges short intervals without connectivity to the rest of the network.

The mobility model plays the key role in the performance of the approach. I use the random waypoint mobility model, rather than the random walk mobility model to optimize the performance of HistDR-Dist. Further research work related to HistDR-Dist can be done by going one derivation deeper and working with the differential of acceleration and angular acceleration. In this case, HistDR-Dist can perform well in simulation using a Gaussian mobility model as well as in a real world implementation.

Just like DVH-Dist and Cent-Dist, the computational cost of HistDR-Dist is very low due to the simple computation and fixed number of components. DVH-Dist returns a failure state in a given situation, for example, less than three samples exist in the history, this method returns a failure state.

All in all, the computational cost of the three improved distance estimation methods can be assumed to have less than the computational cost of DV-Hop alone.

2.4.4 Prediction

The maximum speed, which any node in the network can attain is defined by the parameter V_{Max} given in m s^{-1} . Like the prediction step of the original MCL, the set of particles L_t at the current time t is determined by iterating over the set of previous particles L_{t-1} . For each particle $l_i \in L_{t-1}$, a new particle is drawn from its surroundings within a radius of V_{Max} , reinterpreted in m to describe an area, around it. If the prediction is not performed once per second, the the radius has to be adjusted correspondingly, both in MCL and in Tri-MCL, e.g. for a 0.5 s interval, the radius in which particles may move should be halved.

2.4.5 Filtering in MCL

In MCL, filtering is done by discarding particles that do not lie within one radio range r_{Range} around any of the directly heard anchor nodes and within the ring from one to two radio ranges around indirectly heard (two-hop) anchor nodes. This means that particles $p \in L_t$ are kept by satisfying the following conditions:

$$\exists a \in \text{MCL}_A : d(p, a) \leq r_{\text{Range}}, \quad (2.6)$$

$$\wedge \exists a \in \text{MCL}_I : r_{\text{Range}} < d(p, a) \leq 2 * r_{\text{Range}}. \quad (2.7)$$

The set of directly heard anchor nodes in MCL and the set of two-hop anchors are represented by MCL_A and MCL_I , respectively. As for the re-sampling efficiency, in MCL, filtering is implemented in the following steps,

- 1) One particle is drawn from L_{t-1} .
- 2) The prediction step is run on this particle.
- 3) The decision is made by MCL whether to keep the particle or not.

4) The steps from one to three are repeated until that the new set of particles L_t is full.

This can lead to a high number of iterations of the costly prediction step in order to get enough admissible particles.

2.4.6 Filtering in Tri-MCL

Regarding Tri-MCL, the first part is eschewed and the second part is extended. Only particles $p_k \in L_t$ are kept with the following conditions. Let A be the set of anchor nodes heard over any number of hops, $a_i \in A$ be the i th anchor node and $r_{\{DVHDist, CentDist, HistDRDist\}}$, $i \in RF_i$ be the corresponding distance estimate according to the three different distance estimation methods:

$$i \in \{1, \dots, |A|\}, \quad (2.8)$$

$$k \in \{1, \dots, N\}, \quad (2.9)$$

$$\exists a_i \in A, \exists r \in RF_i : \quad (2.10)$$

$$r - r_{RingWidth} \leq d(p_k, a_i) \leq r + r_{RingWidth}, \quad (2.11)$$

where $d(p_k, a_i)$ refers to the euclidean distance between the position of the particle's and anchor node's position and $r_{RingWidth}$ is one of the parameters of the algorithm, referring to the tolerance with which particles are kept, even if their range does not exactly match that of any range estimates.

After filtering, if the set L_t is less than the required N particles, the following cases are used to remedy the particles until it reaches to the required N number.

1. If it contains no particles at all, it will be reinitialized with the positions of all directly heard anchor nodes as particles.
2. If no directly heard anchor nodes are available, the positions of indirectly (two-hop) anchor nodes are used to seed the set of particles.
3. If still no particles are in the set, it is reset to its state before filtering took place.

If at this point $|L_t| < N$, until $|L_t| = N$, a particle $p_k \in L_t$ is drawn and from it a particle p'_k is sampled from its V_{Max} surroundings, as in the prediction step, and then inserted into

L_t :

$$d(p'_k, p_k) \leq V_{\text{Max}}, \quad (2.12)$$

$$L_t := L_t \cup \{p'_k\}. \quad (2.13)$$

In Tri-MCL, the re-sampling efficiency is very high, where it only needs to run once the prediction step on the contents of L_{t-1} and once per missing particle. Thus, the number of re-sampling runs in that case is bounded by $N - 1$. No further re-sampling step is required to keep the the necessary number of admissible particles in L_t .

In Figure 2.1, a graphical representation of Tri-MCL approach is described, which consisting of one sensor node and two anchor nodes. In this scenario, green particles are kept while red particles are filtered conversely. Each of the six colored rings, one per method and anchor node, is one of the admissible areas for particles and corresponds to the three different types of distance measurement methods I employ. The green particles lie only on the colored rings.

2.4.7 Position Estimation

The method of weighted average of the particles is used to estimate the position in the final step. Each particle is used once per anchor node that is heard by the sensor node and each such instance of a particle is weighted according to two penalty-factors:

$$\forall a_i \in A, \forall p_k \in L_t, \text{ let } p_{i,k} := p_k. \quad (2.14)$$

2.4.7.1 Distances estimation method failure penalty

Let $\omega_{\text{Fail},i,k}$ be the distance estimation method failure penalty factor, which relates to the number ϕ of distance estimation methods that succeeded in estimating a distance to an anchor node associated with certain particles $p_{i,k}$:

$$\epsilon_{i,k} = \frac{|RF_i| - \phi}{|RF_i|}, \quad (2.15)$$

$$\beta_{i,k} = \frac{\epsilon_{i,k} + \sigma}{1 - \epsilon_{i,k} + \sigma}, \quad (2.16)$$

$$\omega_{\text{Fail},i,k} = \frac{1}{\beta_{i,k}}, \quad (2.17)$$

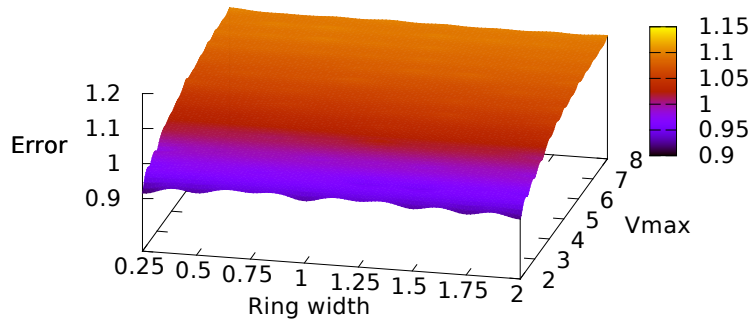


Figure 2.2: Tri-MCL error relative to radio range over varying ring width and V_{Max} , ©2016 IEEE.

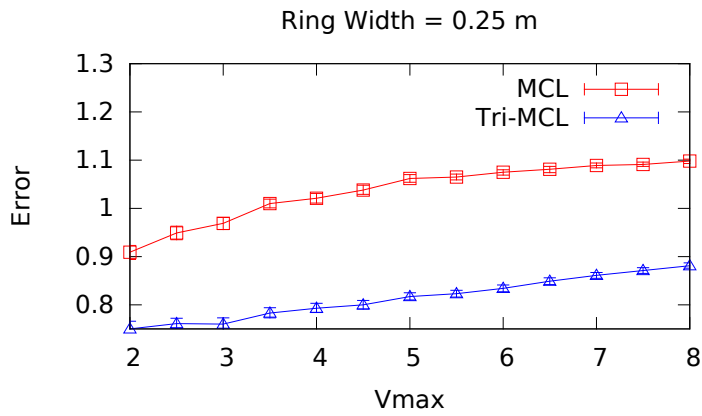


Figure 2.3: Tri-MCL error compared to MCL error over varying V_{Max} with a ring width of 0.25 m, ©2016 IEEE.

with a small $\sigma > 0$, used to avoid dividing by zero.

2.4.7.2 Range error penalty

Let $\omega_{\text{Range},i,k}$ be the range error penalty factor, which represents how well the particle's position matches the estimated ranges. It is computed as the average distance error over all three of the range free distance estimation methods for the given particle $p_{i,k}$, as follows:

$$\xi_{i,k}(r) = \begin{cases} 1, & \text{if } r \text{ failed} \\ \frac{|d(p_{i,k}, a_i) - r|}{\text{maxDistance}}, & \text{otherwise} \end{cases} \quad (2.18)$$

$$\omega_{\text{Range},i,k} = \frac{1}{3} * \sum_{r \in RF_i} \xi_{i,k}(r). \quad (2.19)$$

2.4.7.3 Overall weight

Finally, the final weight of the particle $p_{i,k}$ are formed by the combination of two weights:

$$\omega_{i,k} = \omega_{\text{Range},i,k} * \alpha + \omega_{\text{Fail},i,k} * (1 - \alpha), \quad (2.20)$$

where α is a weighting factor. α selects 0.75 as its value from the experience of different previous trials.

2.4.7.4 Weighted average

A weighted average over all the particles left in the set is employed to calculate the position estimation in the final step.

Let $p_{i,k}^x$ be the x component of the particle $p_{i,k}$ and $p_{i,k}^y$ be the y component, the final coordinates are given as:

$$\omega_{\Sigma} = \sum_{i=0}^{|A|} \sum_{k=0}^N \omega_{i,k}, \quad (2.21)$$

$$x_t = \frac{1}{\omega_{\Sigma}} \sum_{i=0}^{|A|} \sum_{k=0}^N \omega_{i,k} p_{i,k}^x, \quad (2.22)$$

$$y_t = \frac{1}{\omega_{\Sigma}} \sum_{i=0}^{|A|} \sum_{k=0}^N \omega_{i,k} p_{i,k}^y. \quad (2.23)$$

2.5 Evaluation

A set of simulations in an especially built simulation software with varying two simulation parameters is performed in order to evaluate the effectiveness of the proposed Tri-MCL approach. The total number of nodes is 150, where 15 are anchor nodes. All nodes are

randomly distributed over a $100 \text{ m} \times 50 \text{ m}$ simulation area. Each simulation is run 50 times and each time is 300 s. The radio communication range of all nodes is set 20 m.

The modified random waypoint model is utilized to model nodes moving around in the simulation area. To avoid a loss of velocity as described by Yoon et al. [34], the model constrains the admissible combinations of newly picked speed and waypoint combinations in such a way, that the picked combination must be reachable within five simulation seconds. Otherwise a new speed and waypoint combination is chosen until an admissible combination is found [34].

The maximum speed V_{Max} , which is attainable by nodes and the Tri-MCL parameter $r_{\text{RingWidth}}$, is varied over the simulation runs. V_{Max} is varied within $[2, 8]$ with a step size of 0.5, while $r_{\text{RingWidth}}$ is varied within $[0.25, 2]$ with a step size of 0.25.

In order to compare with the baseline schemes, MCL [18] and SA-MCL [28] are used to run in the same environment of Tri-MCL. After that, the final results are used to prove the performance of the proposed Tri-MCL algorithm.

2.5.1 Results

In Figure 2.2, it is clear to see that the error values for Tri-MCL over the different simulation runs. The error is given based on the communication range.

The confidence intervals (99 %) are given for each sample in both Figure 2.3 and 2.4. From them, we can see that Tri-MCL delivers the highest improvement upon MCL at high values for V_{Max} and low values for $r_{\text{RingWidth}}$, with a maximum improvement of 28 % during one simulation run. The highest, average improvement at 25 % over a simulation batch was found with $r_{\text{RingWidth}} = 0.25 \text{ m}$ and $V_{\text{Max}} = 5 \text{ m s}^{-1}$. Even in the worst batch, some slight but significant improvement over MCL can still be obtained, the value of which is 3.5 % at maximum tested $r_{\text{RingWidth}}$ and minimum tested V_{Max} . All in all, it is clear to see that the lower values of $r_{\text{RingWidth}}$ result in better location estimates due to higher precision during the filtering step.

It is interesting to mention that the best accuracy is achieved with both low values for V_{Max} and low values for $r_{\text{RingWidth}}$. However, when it has higher speeds, Tri-MCL performs more robust than MCL and SA-MCL. Besides, the error rate over MCL is higher with higher values for V_{Max} due to its performance deteriorating quickly. We can see that in Figure 2.4, even with a higher $r_{\text{RingWidth}}$ value of 2 m, Tri-MCL performs significantly

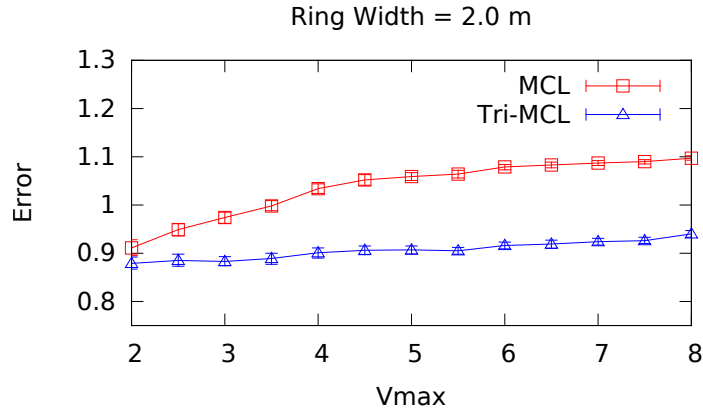


Figure 2.4: Tri-MCL error compared to MCL error over varying V_{Max} with a ring width of 2 m, ©2016 IEEE.

better than MCL.

SA-MCL performs not well in the given scenario, the performance of which is almost the same with that of MCL, so it has been omitted from Figures 2.3 and 2.4 to enhance readability.

2.6 Summary and Future Work

In this chapter, a new localization scheme called Tri-MCL is presented, which improves localization accuracy and increases the efficiency of sampling during the prediction step. The proposed method employs three different distance measurement approaches based on range-free methods to estimate distances between sensor nodes and anchor nodes. These distances are then used to filter out particles not lying within rings around the anchor nodes with a radius corresponding the distance estimates. The weights of different particles are also considered, which means that the weight of each particle is related to the distance between anchor node and sensor node. The results from the simulations and experiments validate the effectiveness of the proposed algorithms in improving localization accuracy and reducing computational costs during re-sampling.

In the future, I aim to implement the proposed algorithm in mobile ad hoc networks with real world deployments.

Chapter 3

Indoor Localization with Mobile Devices

3.1 Motivation

Typical tasks for indoor localization with mobile devices include many applications, such as medical assistance (patient tracking) [4], elderly care (aged pedestrian tracking) [4] and underground mining safety [5], which have attracted many researchers' attention in recent time. However, it is challenging to obtain the accurate pedestrian localization in indoor environment due to multiple reasons. First of all, it is difficult to measure the distance due to the complexity of human movements in the GPS-denied, crowded and cluttered indoor environment. Also any sensor system used by a pedestrian should be wearable and portable, which makes it difficult to use certain sensors, such as laser range scanners although they can be successfully used in robotic applications [35]. In addition, instead of localizing a target in some area sporadically or on demand, the localization of pedestrian should be continuously and possibly in real-time.

With the availability of new small and inexpensive sensors, which enables practical tracking of individuals (who must carry them at all times), the localization of pedestrian in indoor environment has been improved significantly. In recent years, there has been an increasing interest in the development of pedestrian navigation systems for satellite-denied scenarios. The popularization of smart-phones and smart-watches is an interesting opportunity to reduce the infrastructure cost of the positioning systems. If these devices compute their own positions using their internal sensors, it requires very little, if any, physical infrastructure to function. Moreover, this offers a degree of location privacy since users can select whether they share the information with any third party or not.

Some of the existing technological approaches for indoor location systems, such as the infrared light, ultrasonic sensors, Wireless Local Area Network (WLAN), Radio-Frequency Identification (RFID), Ultra Wide-band, ZigBee and computer vision, are not suitable for mobile devices [36] and [37]. Since a dedicated infrastructure or higher processing capabilities are necessary for these technologies, this hinders the systems miniaturization and scalability. In addition, the above technologies can lead to sub-optimal positioning because the communication access points are rarely deployed to provide the optimal location geometry and coverage overlap. Therefore, mobile devices equipped with a variety of sensors (e.g., accelerometer, gyroscope, magnetometer) have become popular in modern indoor localization systems [37].

3.2 Contributions

The contributions of the proposed Twi-AdaBoost algorithm are given as follows,

- 1) Motivated by the lack of a comprehensive approach in multi-device based context recognition research, I propose a multi-device context indoor localization system, named Twi-AdaBoost, exploring the accelerometer, gyroscope and magnetometer sensors on both smart-phone and smart-watch.
- 2) The proposed Twi-AdaBoost algorithm fuses the co-occurrence information to get a better performance for the indoor localization based on the real world data. The indoor localization datasets [38] with the multisource characteristics from the real world is utilized to support simulations.
- 3) Twi-AdaBoost reduces the mean localization errors of position x (0.387 m) by 51.26 % as compared to using Generalized Regression Neural Network (GRNN) algorithm [39] on the combined dataset, where the datasets of smart-phone and smart-watch are merged by simply combining all the features. As for the mean localization error of position y (0.398 m), the proposed fusion approach is improved by 62.56 % compared to GRNN [39]. The other state-of-the-art indoor localization algorithms, such as Support Vector Regression (SVR) [40] perform worse than Twi-AdaBoost and GRNN on the simply combined dataset.

The rest of this Chapter is organized as follows. In Section 3.3, it reviews related works. Section 3.4 presents the proposed Twi-AdaBoost fusion strategy and its knowledge background. The datasets analysis and pre-processing, experimental results as well as per-

formance evaluations are introduced in Section 3.5. Finally, Section 3.6 summarizes this Chapter.

3.3 State of the Art

In outdoor environments, GPS is one of the most popular way to localize mobile devices. However, in indoor environments where the GPS signals are not receivable or usable, different models were proposed to solve the indoor localization problems.

Chen et al. [41] proposed a Convolutional Neural Network that used the Channel State Information of only one access point and achieved an average localization error of 1.36 m, but has a high training complexity. In contrary, a low computational complexity model was proposed in [42] which achieved a localization error of 2.1 m. In that model, an AdaBoost algorithm with C4.5 method as a weak classifier was used to combine the Received Signal Strength (RSS) and orientation information to improve the accuracy of indoor localization. It included two phases, the offline phase and online phase. In the offline phase, a database of the RSS from different access points at each reference location for the target environment was built; in online phase, the localization was determined by means of a sample of RSS collected in a particular position and an estimation model that used database information. Although the proposed model was not more accurate than other models, it demonstrated that it was possible to execute such models on resource-constrained devices. GRNN was proposed in [39], where RSS data gathered at the access points from the referenced nodes were used to train the GRNN model and the target node position was calculated by the weighted centroid method. Wu et al. [40] used the SVR model to solve the missing value location estimation problem. Utilizing other machine learning technologies, such as Linear Regression (LR) [43] which is a RSS-based localizaiton method, localization accuracy was improved by correcting the distance circles using LR model.

In [44], a sensor fusion framework was proposed by combining WiFi, Pedestrian Dead Reckoning (PDR) and landmarks. It used the linear Kalman filter to simplify the sensor fusion problem on a smart-phone. The weighted path loss algorithm was used in the WiFi localization due to its simplicity and effectivity, while in the pedestrian dead reckoning approach the initial estimation error was amended by landmarks. A Kalman filter was used to fuse magnetometer and gyroscope records in order to improve the accuracy of walking direction estimation. The localization accuracy of this approach was 1 m on an average. However, the additional landmarks with the known positions should be

provided to help this approach restart when the users went through these landmarks. At the same time, Ma et al. [45] used the weighted fusion to improve the WiFi-based indoor localization. There were two steps in this algorithm: the offline acquisition and the online localization. In the offline acquisition process, the optimal parameters were selected to complete the signal acquisition. In addition, the fingerprints database was built. In the online localization process, a pre-match method was employed to select the candidate fingerprints to shorten the positioning time. Then, two intermediate results were obtained by using the improved Euclidean distance and the improved joint probability. The final results were calculated by fusing these two intermediate results with different weights. More similar work can be found in [46], [47] and [48]. However, the time required to install, configure and maintain the WiFi systems together with the expense of access points have limited the general deployment of these indoor algorithms.

Fusing the internal sensors is popular in human activity recognition [49, 50]. For example, in [49], the data coming from embedded sensors on the smart-phone and environmental sensors were fused by a decision tree based on multi-sensor data-stream. Then they used the Recurrent Neural Networks (RNNs) to model the RSS stream.

However, few previous researchers did work on sensor fusion from multi-device in indoor localization. In this chapter, to use the richer context information, I propose a Twi-AdaBoost algorithm which combines the data of self-contained sensors from multiple devices, like smart-phone and smart-watch.

3.4 Methodology

Indoor localization has been an important issue in recent time. To solve this problem, a Twi-AdaBoost fusion strategy is proposed, exploiting the intrinsic correlation between two conditional independent datasets from smart-phone and smart-watch to boost the ability of prediction of the pedestrian's location from a crowded and cluttered background.

3.4.1 Building a Weak Learner Based on the CART Algorithm

In 1984, Breiman et al. [51] proposed the CART method by building a binary decision tree according to some splitting rules based on the predictor variables to address the regression problem. Suppose that the CART method takes a training dataset with instances $(x_1, y_1), \dots, (x_n, y_n)$ as input, where each x_i belongs to the features space X (such as

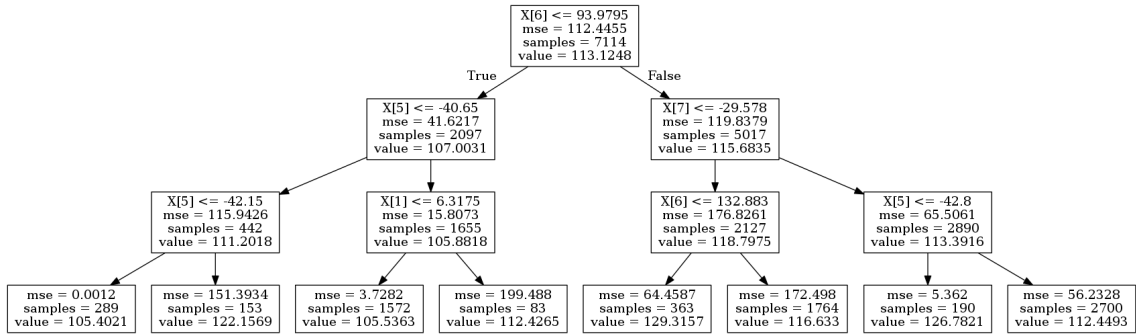


Figure 3.1: Example of cart regression tree.

accelerometer, gyroscope and magnetometer) and each label y_i is in the reference location dataset and $y_i \in R$. Figure 3.1 depicts an example of the cart regression tree based on the experiment dataset [38]. The subsets created by the splits are named nodes, otherwise, they will be named by terminal nodes. A regression tree partitions the X -space into disjoint regions A_k and provides a fitted value $E(Y|X \in A_k)$ within each region.

The tree is implemented recursively with the following steps in Algorithm 4.

Algorithm 1: Construction of Cart regression model

Input : Training dataset $(x_1, y_1), \dots, (x_n, y_n)$ and $y \in R; CART$.

for Each node $x_i \in X$ **do**

Examine every allowable split on each reference location variable y_i . Binary questions, like $I_s x_i > c?$, are used to generate the binary splits.

Select and execute the 'best' of these splits.

Stop splitting on a node when some stopping rule is reached.

end

Output: CART regression model $H(x)$

The CART regression method is selected as the weak learner based on the following two main reasons:

- It is simple and fast. In addition, it is not significantly impacted by outliers in the input variables.
- It is nonparametric and does not rely on the dataset distribution.

3.4.2 AdaBoost.R2 Regression Model

AdaBoost.R2 is one of most popular ensemble learning algorithms, which is designed to solve the regression problem [52]. In AdaBoost.R2, a set of weak classifiers are trained to form a strong classifier. Initially, each training instance receives a uniform weight w_i , which indicates the relative importance of each instance. After each iteration, the weight of the instance with the larger real-valued error $e_i = \frac{|y_i - h_t(x_i)|}{\max_{i=0}^n |e_i|}$ will be increased, otherwise, the weight will be decreased. In this case, the weaker learner is forced to focus on the “hard” examples in the training dataset. In particular, three loss functions can be selected in AdaBoost.R2: $e'_i = \frac{e_i}{D}$ (linear), $e'_i = (\frac{e_i}{D})^2$ (square), $e'_i = 1 - \exp(-e_i/D)$ (exponential). The pseudo code of AdaBoost.R2 is given in Algorithm 5.

Algorithm 2: AdaBoost.R2

Input : Training dataset $(x_1, y_1), \dots, (x_n, y_n)$ $y \in R$; *WeakLearner*; Iteration T ;
Initial weight distribution $D_t i = \frac{1}{n}$, $i \in [1, n]$

for each iteration $t \in [1, T]$ **do**

Call *WeakLearner*, providing it with a distribution D_t .

Build the regression model: $h_t(x) \rightarrow y$ for regression problems.

for each instance x_i **do**

| Calculate the adjusted error $e_i^t = \frac{1 - \exp(-|y_i - h_t(x_i)|)}{D_t}$. $D_t = \max_{j=1}^n |y_j - h_t(x_j)|$.

end

Calculate the adjusted error of h_t : $\epsilon_t = \sum_{i=1}^n e_i^t w_i^t$; if $\epsilon_t \geq 0.5$, stop and set
 $N = t - 1$.

$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$.

Update the weight vector: $w_i^{t+1} = \frac{w_i^t \beta_t^{1 - e_i^t}}{Z_t}$, where Z_t is a normalization factor
selected such that w_i^{t+1} will be a distribution.

end

Output: Strong classifier $H(x)$ is the weighted median of $h_t(x)$ for $t \in [1, T]$, using
 $\log \frac{1}{\beta_t}$ as the weight.

3.4.3 Twi-AdaBoost Fusion Strategy

Localization techniques based on individual dataset have their own strengths and weaknesses. In this chapter, I investigate the potential of fusing both smart-phone and smart-watch datasets to better infer the pedestrian’s indoor localization.

Figure 3.2 depicts the proposed Twi-AdaBoost algorithm based on the collaborative exploitation of smart-phone-smart-watch characteristics. The training datasets are first

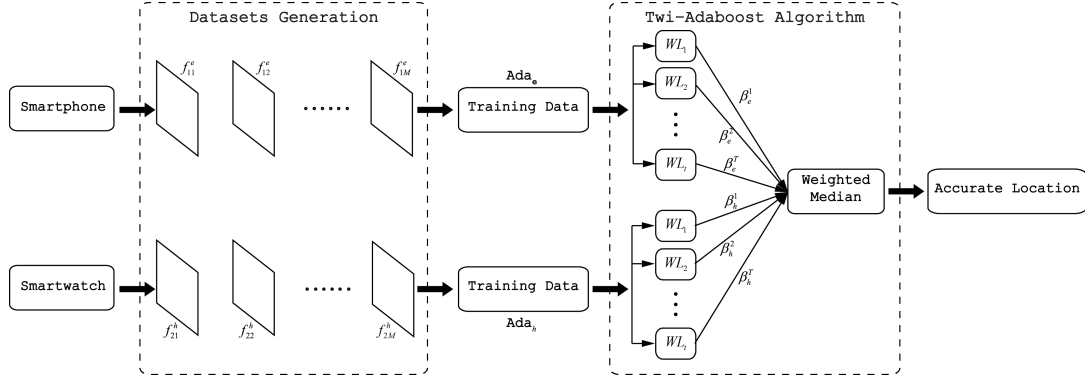


Figure 3.2: Proposed Twi-AdaBoost algorithm based on the collaborative exploitation of smart-phone and smart-watch.

extracted from smart-phone and smart-watch using their internal sensors, such as the accelerometer, gyroscope and magnetometer. Then, the Twi-AdaBoost strategy is used to improve the localization performance. Ultimately, the accurate location is obtained by combing all generated weak learners.

Figure 3.3 illustrates the interactive ensemble learning process across multiple datasets to form a consolidated fusion by interactively exploiting the complementary sensor features from different devices, which is the key difference from the traditional AdaBoost.R2 algorithm. The pseudo-code for Twi-AdaBoost is given in Algorithm 3. Twi-AdaBoost works by training the weak learner with an initial sample weight and evaluating its prediction by comparing the results to each other in form of the penalty factor. With this information new weights are generated and used for the next iteration.

In Algorithm 3, initially, each sample has a uniform weight $D_j^1(i) = 1/M$ of the i th training sample on the j th dataset, which indicates the relative contribution of each sample for the final prediction result. The weight will be changed after each iteration. The weight $D_j^{t+1}(x_{ji})$ of each sample in Twi-AdaBoost is decided by both the real-valued error $L_j^t(x_{ji})$ and the punishment factor $P_j^t(x_{ji})$, which is introduced to convey the complementary characteristics across the two datasets into the ensemble learning process. The penalty degree of the weight is controlled by the scale factor $P_j^t(x_{ji})$, which is decided by the value of $p_j^t(x_{ji})$ of all weak learners $f_j^t(x_{ji})$ achieving the agreement with both $f_k^t(x_{ji})$ and y_i at the t th iteration. With exploitation of both datasets from smart-phone and smart-watch, the “hardest” samples will be punished with the largest weights, which forces the new weak learners to focus on the “hardest” samples in the next generation and helps this algorithm to achieve better performance.

Algorithm 3: Multi-Device AdaBoost Algorithm

Input : The training dataset $S = \{(x_{ji}, y_i)_{j=1, \dots, N; i=1, \dots, M}\}$, where N is the number of different datasets from different devices and M the number of samples; *WeakLearner*; Iteration T ; Initial weight of each sample: $D_j^1(i) = 1/M$.

for $t = 1$ to T **do**

for $j = 1$ to N **do**

 Get a random integer $r \in [1, M]$ and generate a subset R , containing the r highest weighted samples of S .

 Train the weak classifier with R and D_j^t and build the regression model $f_j^t(x_j)$.

end

for $j = 1$ to N **do**

 Calculate the distance of each sample x_{ji} in S and the prediction with

$$l_j^t(x_{ji}) = |f_j^t(x_{ji}) - y_i|.$$

 Calculate the loss function $L_j^t(x_{ji})$ for each sample using the exponential loss

$$\text{function as } L_j^t(x_{ji}) = 1 - \exp\left(-\frac{l_j^t(x_{ji})}{\max_{i=1, \dots, M} (l_j^t(x_{ji}))}\right).$$

 Calculate the weighted loss as $\bar{L}_j^t = \sum_{i=1}^M L_j^t(x_{ji}) D_j^t(x_{ji})$.

$$\text{Set } \beta_j^t = \frac{\bar{L}_j^t}{1 - \bar{L}_j^t}.$$

 For each sample x_{ji} in S , calculate the punishment factor

$$P_j^t(x_{ji}) = 1 - \exp\left(-\frac{p_j^t(x_{ji})}{\max_{i=1, \dots, M} (p_j^t(x_{ji}))}\right), \text{ where}$$

$$p_j^t(x_{ji}) = \frac{1}{N} (|f_j^t(x_{ji}) - y_i| + \sum_{k=1}^N (|f_j^t(x_{ji}) - f_k^t(x_{ji})|)).$$

 For each sample x_{ji} in S , set $D_j^{t+1}(x_{ji}) = \frac{D_j^t(x_{ji}) \beta_j^t (1 - L_j^t(x_{ji})) (1 - P_j^t(x_{ji}))}{Z_j^t}$ where Z_j^t is the normalization factor such that D_j^{t+1} will be a distribution.

end

end

Output: The strong classifier $F(x)$ is the weighted median of

$$f_j^t(x_j)_{(t=1, \dots, T; j=1, \dots, N)}, \text{ with } \log\left(\frac{1}{\beta_j^t}\right) \text{ used as the weight.}$$

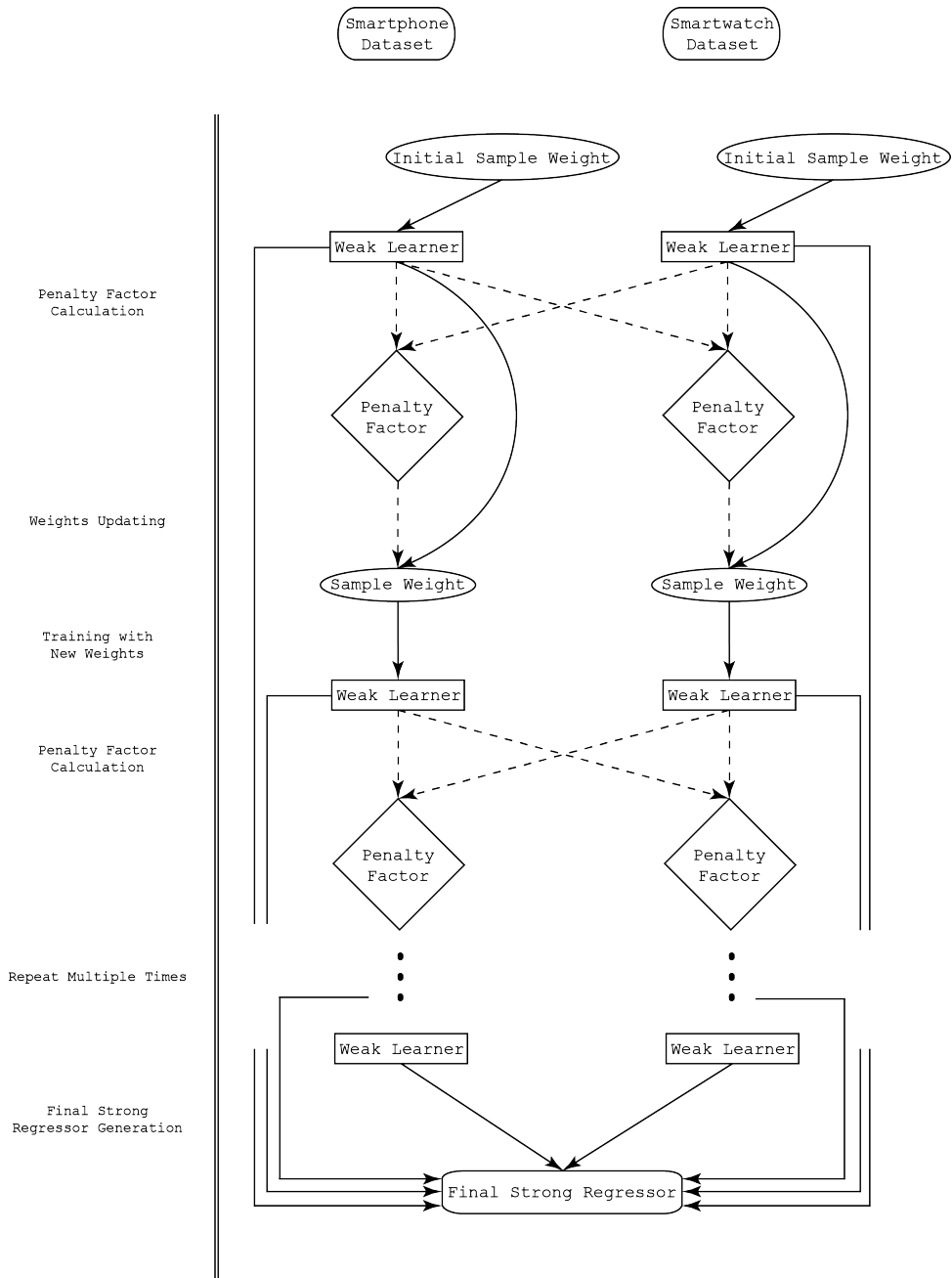


Figure 3.3: Interactive training process of Twi-AdaBoost across datasets from both smart-phone and smart-watch.

	smart-phone	smart-watch
Measurement 1	18355	58370
Measurement 2	17787	58370

Table 3.1: Number of total samples recorded.

3.5 Experiments

3.5.1 Dataset Analysis

In the experiment, the indoor localization datasets of paper [38] are employed to test the proposed algorithm. The datasets with over 36000 continuous samples are collected in a 185.12 m^2 real indoor environment. The user was wearing two devices simultaneously, such as a Sony Xperia M2 smart-phone and a LG W110G smart-watch, to collect the data in each campaign. Figure 3.4 from paper [38] depicts the overall map, where the data collection was performed. There are two rooms, two corridors and one small entrance hall inside this indoor office environment. Each dot in the map corresponds to a detection point and each dot is 0.6 meters far from another since each dot occupies $0.6 \text{ m} \times 0.6 \text{ m}$. For each of them, features of sensors in each device were collected. A zig-zag trajectory was performed by two different users who were wearing the same equipments to cover the entire map. The walking speed of each user was 0.6 m s^{-1} on an average. Each sample was collected about every 100 millisecond and the collection time is very short.

All the recorded datasets include the following features:

- Place ID, Timestamp;
- Accelerometer_X, Accelerometer_Y, Accelerometer_Z, MagneticField_X, MagneticField_Y, MagneticField_Z, X_Axis Angle (Pitch), Y_Axis Angle (Roll), Z_Axis Angle (Azimuth), Gyroscope_X, Gyroscope_Y, Gyroscope_Z.

The exact numbers of recorded samples in each measurement can be found in Table 3.1.

3.5.2 Dataset Preprocessing

The datasets [38] were collected by recording the internal sensor data of different devices about every 100 millisecond when the walking speed of each user was 0.6 m s^{-1} on an average. Thus, they might be not perfectly synchronized and have a slight offset in time.

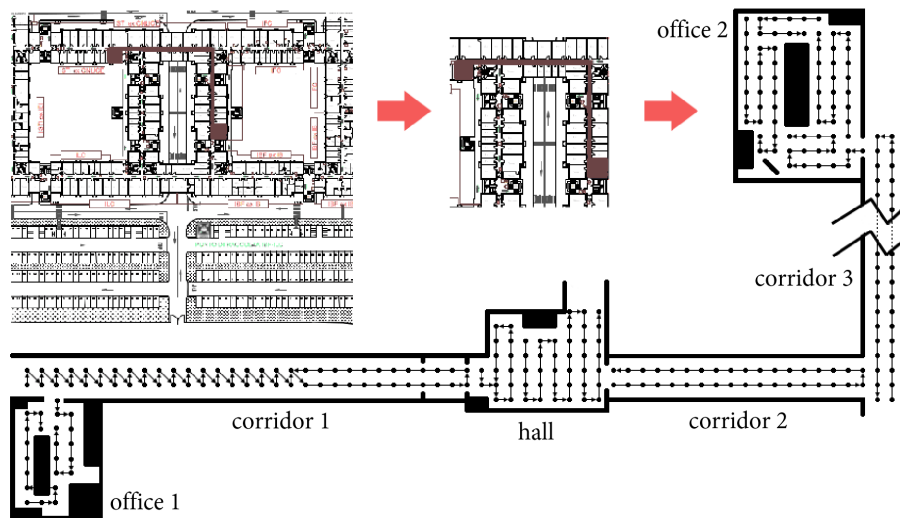


Figure 3.4: Map of the data collecting environment.

Therefore, I filter the provided data beforehand to enable their suitability for the proposed algorithm. Furthermore, some samples are not usable, as their precise recording position is unknown.

The samples of datasets are pre-processed and filtered out according to the following conditions:

1. As the datasets contain recorded samples that are not uniquely assignable to the given reference points, some samples are removed in order to assure the correct labeling of the data.
2. Each sample needs to have a counterpart-sample in all other datasets, which was recorded within a 50 ms sliding window, in order to make sure that the sample-pairs were recorded almost simultaneously. For example, a sample pair consists of two samples, one recorded on the smart-phone and one recorded on the smart-watch with max time difference of 50 ms.
3. Each sample can only be chosen either once or never to ensure that no sample is used twice and therefore unintentionally weighted higher than the other ones. This creates a one-to-one relationship between the selected samples of each dataset, illustrated in Figure 3.5.

After filtering out the datasets according to the above conditions, there are 14228 samples and 12608 samples from both smart-phone and smart-watch in the first measurement and

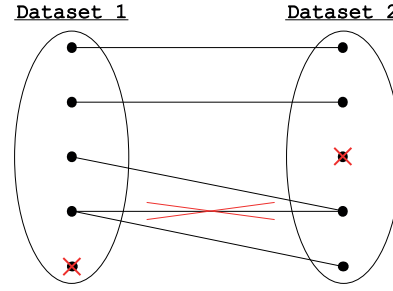


Figure 3.5: Illustration of a valid possible connection between samples of each dataset after pre-processing.

second measurement, respectively.

Finally, all features are normalized using min-max scaling technique [53]. The definition of it is given in equation (3.1), where $x \in X$ is the original and $x' \in [a, b]$ is the rescaled value,

$$x' = \frac{(b - a)(x - \min(X))}{\max(X) - \min(X)}. \quad (3.1)$$

3.5.3 Performance Metrics

In order to evaluate the results of Twi-AdaBoost algorithm, the performance metrics provided by scikit-learn [54] are employed in this chapter. For example Root Mean Squared Error (RMSE), which is used to measure the differences between the values estimated by a model and the values actually observed; Explained Variance Score (EVS), which is used to compute the explained variance regression score; Mean Absolute Error (MAE), which is a risk metric corresponding to the expected value of the absolute error loss as well as the box-and-whiskers plots, which can display the variation in samples of a statistical population and detect the outliers being plotted as individual points.

The RMSE estimated over $n_{samples}$ is defined as equation (3.2), where y'_i is the predicted value of the i -th sample and y_i is the corresponding true value. The smaller the RMSE value is, the better the performance of the proposed Twi-AdaBoost algorithm. The EVS is estimated as equation (3.3), where Var is the variance, i.e. the square of the standard deviation. The higher the value is, the better the performance. The best possible score is

1.0. Equation (3.4) depicts the MAE estimated over $n_{samples}$,

$$\text{RMSE}(y, y') = \sqrt{\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - y'_i)^2}, \quad (3.2)$$

$$\text{EVS}(y, y') = 1 - \frac{\text{Var}\{y - y'\}}{\text{Var}\{y\}}, \quad (3.3)$$

$$\text{MAE}(y, y') = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (|y_i - y'_i|). \quad (3.4)$$

3.5.4 General Results and Analysis

To verify the performance of the proposed Twi-AdaBoost algorithm, I use 12608 samples including 6304 samples of smart-phone and 6304 samples of smart-watch in the second measurement. About 85% samples are randomly selected as the training set and the rest samples as testing set. The metrics RMSE, EVS as well as box-and-whiskers plots are utilized to evaluate the performance. In all the figures, AdaBoost.R2 on SH denotes AdaBoost.R2 on smart-phone dataset; AdaBoost.R2 on SW denotes AdaBoost.R2 on smart-watch dataset while AdaBoost.R2 on HW denotes AdaBoost.R2 on the mixed dataset, where the datasets of smart-phone and smart-watch are merged by simply combining all the features.

3.5.4.1 Root Mean Square Error

In both Figure 3.6 and Figure 3.7, we can see that with the iteration increase, the RMSE decreases. At the 50th iteration, both AdaBoost.R2 and Twi-AdaBoost tend to be stable. Figure 3.6 depicts the RMSE of position x estimation using Twi-AdaBoost and AdaBoost.R2 methods with iteration T on the testing set. It is clear from Figure 3.6 that compared to AdaBoost.R2 on SH (1.42), the RMSE value achieved by Twi-AdaBoost (0.69) is 51.63% lower while the RMSE value obtained by Twi-AdaBoost is about 49.90% lower than that of AdaBoost.R2 on SW (1.37) at the 50th iteration. Compared with AdaBoost.R2 on HW, Twi-AdaBoost achieves 40.54% improvement. Figure 3.7 shows that RMSE of position y estimation using Twi-AdaBoost and AdaBoost.R2 methods with iteration T on the

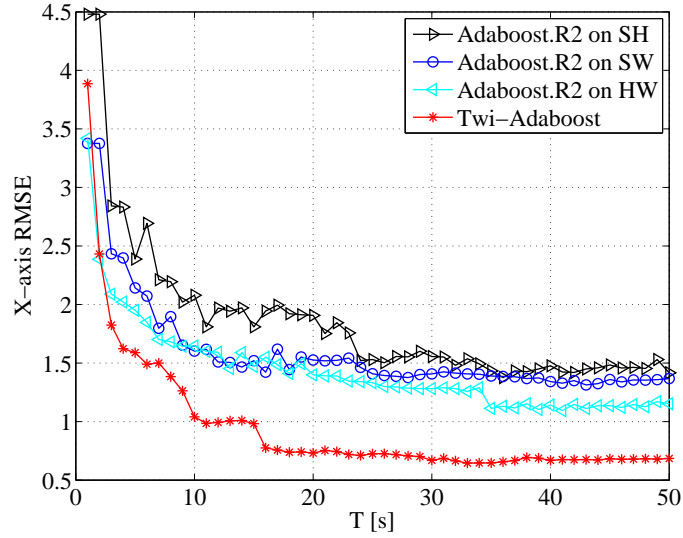


Figure 3.6: RMSE of position x estimation using Twi-AdaBoost and AdaBoost.R2 methods.

testing set. We notice that the RMSE of AdaBoost.R2 on SW (1.26) is better than that of AdaBoost.R2 on SH (1.81). However, the RMSE value achieved by Twi-AdaBoost (0.73) is 42.32 % lower compared to AdaBoost.R2 on SW at the 50th iteration. Compared with AdaBoost.R2 on HW (1.32), Twi-AdaBoost achieves 45.10 % improvement.

3.5.4.2 Explained Variance Score

As visible in Figure 3.8 and Figure 3.9, the EVS of both position x and y increase with the increase of iteration. However, after the 30th iteration, the performance of EVS becomes stable. Figure 3.8 describes that the EVS of position x estimation using Twi-AdaBoost and AdaBoost.R2 methods with iteration T on the testing set. It is demonstrated that Twi-AdaBoost outperforms AdaBoost.R2 on SH, SW and HW, respectively. Specifically, the EVS of position x estimation of it attains 1.19 %, 1.08 % and 0.66 % higher compared to AdaBoost.R2 on SH, AdaBoost.R2 on SW and AdaBoost.R2 on HW, respectively. In Figure 3.9, it shows the EVS of position y estimation using Twi-AdaBoost and AdaBoost.R2 methods with iteration T on the testing set. The EVS of position y estimation of AdaBoost.R2 on SH is worse than that of AdaBoost.R2 SW while the EVS of AdaBoost.R2 on HW has almost same performance with that of AdaBoost.R2 SW. The EVS of y position estimation attained by Twi-AdaBoost is the highest one with the EVS value 99.55 %.

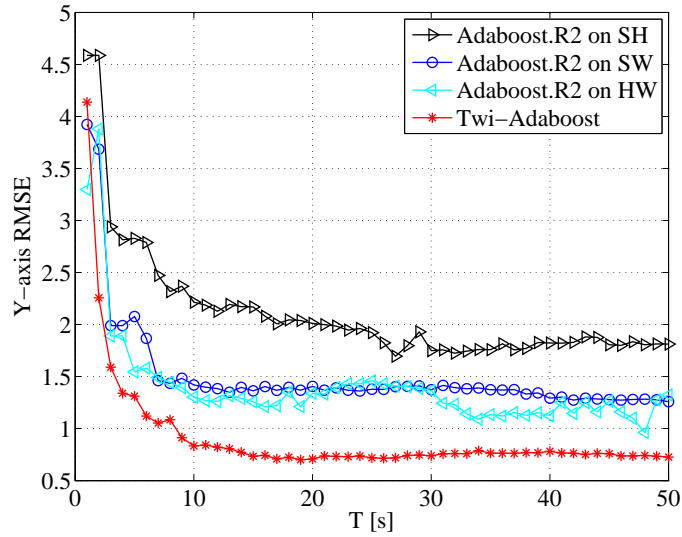


Figure 3.7: RMSE of position y estimation using Twi-AdaBoost and AdaBoost.R2 methods.

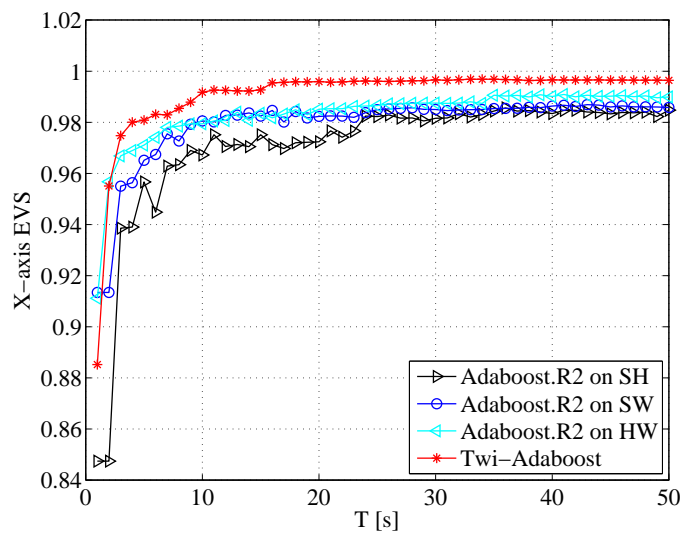


Figure 3.8: EVS of position x estimation using Twi-AdaBoost and AdaBoost.R2 methods.

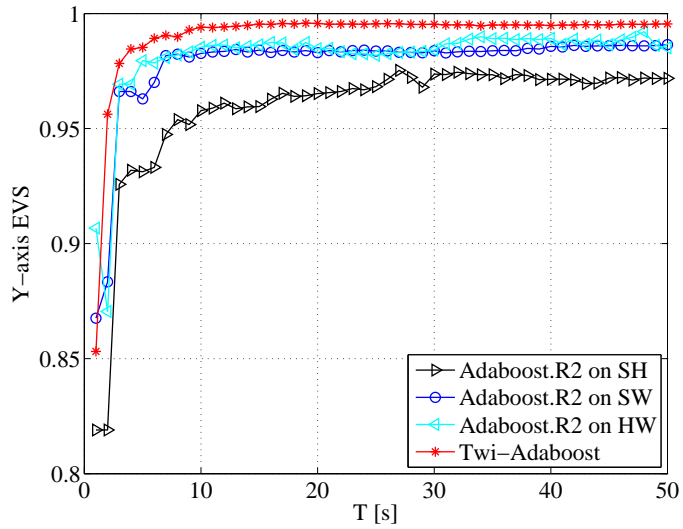


Figure 3.9: EVS of position y estimation using Twi-AdaBoost and AdaBoost.R2 methods.

3.5.4.3 Box-and-whiskers Plots

Box-and-whiskers plots of the predicted position offsets are shown in Figure 3.10, Figure 3.12 and Figure 3.13, which is a more complete performance analysis. The boxes refer to different values of the updated period T based on different datasets using AdaBoost.R2 and Twi-AdaBoost, where the boundaries of the box represent the 25th and 75th percentiles of the sample data, respectively; the line within the box shows the median; Whiskers above and below the box indicate the range from the 90th percentiles and 10th percentiles, respectively; the outliers are shown as dots. Notice that there are more outliers in Figure 3.10 and Figure 3.12, which are obtained from smart-phone and smart-watch dataset using AdaBoost.R2, respectively. We can see that Figure 3.13 obtains the best performance.

3.5.5 Comparison Results and Analysis

Table 3.2 illustrates the comparison results among the proposed Twi-AdaBoost method and the state-of-the-art indoor localization algorithms, where XRE denotes the RMSE on X coordinate; YRE denotes the RMSE on Y coordinates; Twi-Ada is Twi-AdaBoost; Ada.RT is AdaBoost.RT. It displays the performance of the different models, using the metrics introduced previously, given the HW dataset, where the datasets of smart-phone and smart-watch are merged by simply combining all the features. I conduct extensive numerical studies on randomly selected different initial data. It is clear to see that the

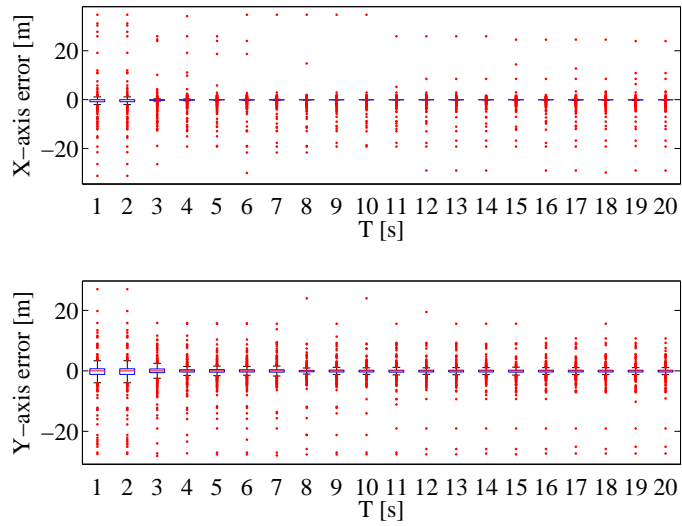


Figure 3.10: Box-and-whiskers plots of the position offsets using AdaBoost.R2 based on the smart-phone dataset.

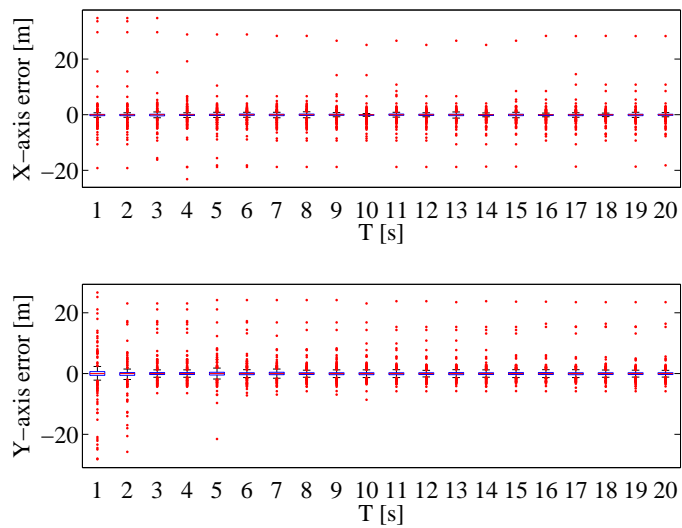


Figure 3.11: Box-and-whiskers plots of the position offsets using AdaBoost.R2 based on the smart-watch dataset.

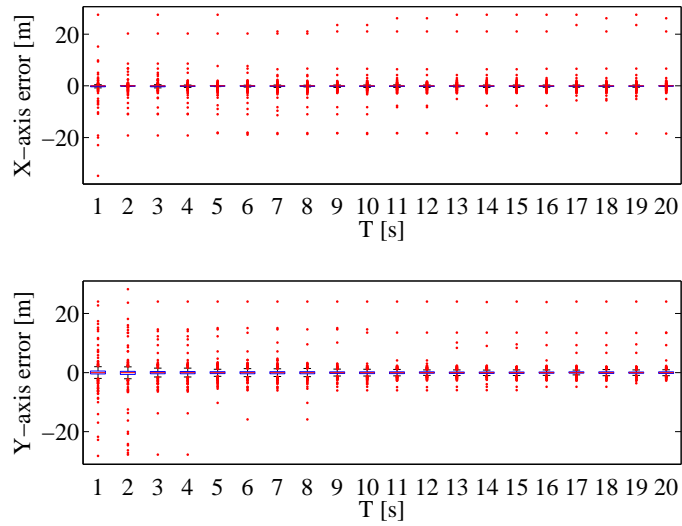


Figure 3.12: Box-and-whiskers plots of the position offsets using AdaBoost.R2 based on the mixed dataset.

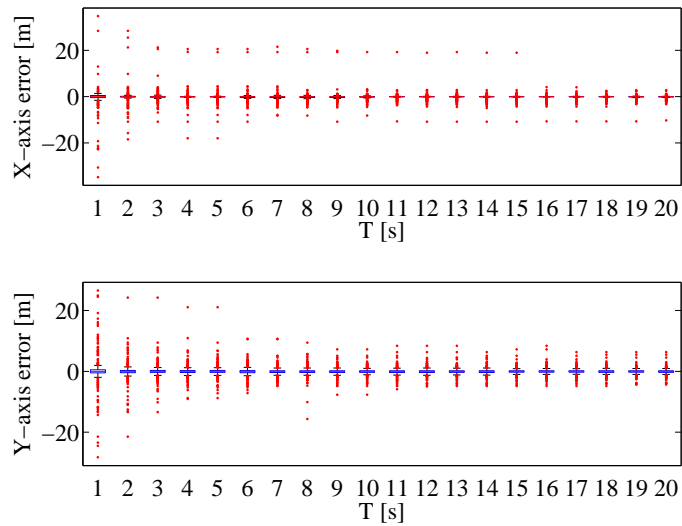


Figure 3.13: Box-and-whiskers plots of the position offsets using Twi-AdaBoost based on the mixed dataset.

Table 3.2: Comparison results among Twi-AdaBoost and the state-of-the-art.

	XRE	YRE	XMAE	YMAE	XEVS	YEVS
Twi-Ada	1.073	0.824	0.387	0.398	0.991	0.994
LR [43]	8.658	9.123	6.746	7.729	0.411	0.286
Ada.RT [55]	2.375	2.020	0.862	0.975	0.955	0.965
GRNN [39]	1.905	2.455	0.794	1.063	0.971	0.948
SVR [40]	7.328	7.908	5.071	6.475	0.584	0.464

proposed Twi-AdaBoost outperforms the other algorithms throughout all metrics on both coordinates. GRNN performs second best on the X coordinate, but worse than Ada.RT on the Y coordinate.

3.6 Summary and Future Work

In this chapter, I introduce Twi-AdaBoost, an indoor collaborative localization algorithm that explores the accelerometer, gyroscope and magnetometer sensors on both smart-phone and smart-watch. The key contribution of the proposed Twi-AdaBoost algorithm is fusing the co-occurrence information to get a better performance for the indoor localization based on the real world data. The indoor localization datasets [38] used in this chapter have the multi-source characteristics, which are supported by the presence of two different devices collecting data simultaneously from the surrounding environment: a smart-phone and a smart-watch, respectively. Each device collects multivariate data represented by their internal sensors, such as acceleration, orientation, and gyroscope. From the experiment results, it is obvious that Twi-AdaBoost convincingly outperforms the state-of-the-art indoor localization algorithms, taking advantage of the co-occurrence correlation across the sensors from multiple devices. Specifically, the localization error of position x and y achieved by Twi-AdaBoost is 0.387 m and 0.398 m, respectively.

Considering the future work, I plan to utilize the correlation between the position x and y in the same location to improve the performance of the indoor localization in this chapter. In addition, I will focus on exploiting the datasets combined by more co-occurrence information from multiple devices, like the Camera and WiFi, by machine learning methods to improve the localization accuracy in indoor environment.

Chapter 4

Underwater Localization with Wireless Sensor Networks

4.1 Motivation

Localization of mobile sensor nodes is indispensable for UWSNs. The gathered data is useless if it does not refer to a specific position of the sensor node. Many applications, such as aquatic environment monitoring, target tracking [6], geo-routing protocols [56] and pollution control, require the location information. GPS cannot be used in UWSNs because of the high attenuation of radio signal and their power hungry nature. UWSNs are mostly based on acoustic communication systems which suffer from many problems because of the aquatic conditions, such as frequency dispersion, multi-path fading, limited bandwidth and energy [57,58].

One important challenge is that underwater sensor nodes have limited resources due to the non-rechargeable batteries, which directly determines the network life time. Considering about the engineering hardship and monetary cost of battery replacement, the energy-efficient localization becomes more critical to extend the network lifetime in UWSNs. The energy of sensor nodes is consumed mainly by transmitting and receiving packets, which is much larger than that of the idle listening consumption [59], so adjusting the transmission power by topology control is one possible way to save energy in UWSNs. In most localization systems of UWSNs [58], multiple anchor nodes are required to help one sensor node to find its position. The performance of localization methods depend on the initial reference position, number of sensor nodes, number of anchor nodes, ranging technique as well as the position of anchor nodes.

These facts motivate us to pursue solutions to assist the sensor node finding enough anchor nodes with the minimum power consumption by topology control to extend the network lifetime. So far, limited number of schemes have been proposed for the localization service in UWSNs [60], [61], [62], especially for localization in sparse UWSNs by using topology control [63]. However, the proposed scheme in [63] only considers energy-saving for anchor nodes. Sensor nodes deployed underwater consume plenty of energy, which is a major cause of decrease in UWSNs.

In this chapter, unlike above works, such as Opportunity Localization Topology Control (OLTC) [63] and three-Dimensional Underwater Localization (3DUL) [64], I leverage the benefits of topology control to achieve a high localization performance and a low energy consumption. In localization systems of UWSNs, in order to successfully localize a single sensor node, multiple anchor nodes are required. This motivates us to model this problem as a Single-Leader-Multi-Follower Stackelberg game. The energy-efficient utilities and game scheme design make the proposed EELA scheme distinct from the existing state-of-the-art, such as OLTC [63] and 3DUL [64]. Based on EELA scheme, the Adaptive EELA scheme is designed by considering the localization scenario with a sensor node (unaware of its location) and its surrounding neighbor anchor nodes (aware of their locations) as an oligopoly and a few hours of training with no need for prior knowledge, new equipment, or extra cost.

4.2 Contributions

In this chapter, I aim to design the EELA scheme and Adaptive EELA scheme by considering the localization scenario with a sensor node and its surrounding neighbor anchor nodes as an oligopoly. The sensor node needs the help of its neighbor anchor nodes to get its position. As for EELA, the Single-Leader-Multi-Follower Stackelberg game is employed to model this scenario, where the sensor node will act as the leader while the anchor node will be the follower. As for Adaptive EELA, the Single-Leader-Multi-Follower Stackelberg game using the adaptive neuro-fuzzy model as its payoff function to model this scenario, where the sensor node will act as the leader while the anchor node will be the follower. In both these two proposed schemes, each sensor node attempts to exploit its maximum opportunities to communicate with potential anchor nodes to get localized with the minimum localization delay and energy cost. Each anchor node, on the other hand, exploits its maximum opportunities to interact with the potential sensor nodes with

its maximum individual utility.

In summary, the contributions of the proposed EELA are given as follows:

- 1) I propose one novel algorithm, named EELA, which utilizes the Single-Leader-Multi-Follower Stackelberg game to model the localization scenario and build a strategic interaction between a sensor node and multiple anchor nodes for solving the energy-efficient localization problem in UWSNs.
- 2) As for sensor nodes, 'two-hop' anchor nodes and energy consumption are considered to select an optimal transmission power to find the maximum number of anchor nodes with minimum overhead consumption.
- 3) As for anchor nodes, both the ability to localize sensor nodes and the energy cost are considered to select an optimal transmission power to localize maximum number of sensor nodes with minimum energy cost.
- 4) I conduct extensive numerical evaluations to show that the proposed EELA scheme achieves about 48 % energy reduction for all nodes on average compared to the state-of-the-art approach.

The contributions of the proposed Adaptive EELA are given as,

- 1) The Adaptive EELA using an Adaptive fuzzy Single-Leader-Multi-Follower Stackelberg game is proposed to model the localization scenario and build a strategic interaction between a sensor node and multiple anchor nodes. This solves the energy efficient localization problem in UWSNs.
- 2) The proposed Adaptive EELA scheme is able to automatically learn in the offline phase, which is required only once. Then, in the online phase, it can adapt to the environmental changes or dynamical changes, such as the densities of nodes or topologies of nodes.
- 3) I conduct extensive numerical evaluations under different network topologies and different network node densities. The simulation results demonstrate that the proposed Adaptive EELA scheme achieves about 35 % and 66 % energy reduction per node on average comparing the state-of-the-art approaches, such as Fixed-EELA and OLTC, while providing a desirable localization coverage, localization error and localization delay.

The rest of the Chapter is organized as follows. Section 4.3 discusses the related works.

Section 4.4 introduces the system model. The detailed description of the proposed EELA scheme is presented in Section 4.5. Simulation results and performance evaluation are shown in Section 4.6.3. Finally, Section 4.7 presents the summary and future work.

4.3 State of the Art

4.3.1 Localization in UWSNs

A set of localization techniques has been proposed for UWSNs in recent years. A detailed survey about these works was given in [65] and [66]. One localization scheme was presented in [67], which worked well in high latency networks and improved the energy efficiency as well. One-way and two-way MAC-layer message delivery were combined in this paper. However, it is only suitable for static UWSNs due to its assumption of constant propagation delays among sensor nodes. In [64], 3DUL described a distributed, iterative and dynamic solution to the localization problem in the underwater acoustic sensor network with only three anchor nodes at the surface of the water. Trilateration algorithm was used to estimate the sensor node location. However, the error accumulated with the iteration increase, leading to inaccurate positioning of the later sensor node in 3DUL. One localization scheme [61] was proposed in a hierarchical underwater sensor networks consisting of surface buoys, anchor nodes, and ordinary nodes. Sensor nodes were localized by the trilateration method. However, the node density is assumed to be high in this scheme due to the long distance acoustic communication between the anchor node and the surface buoy. In [62], a novel scheme was proposed for long-term maritime surveillance monitoring tasks in ocean sensor networks. Liu et al. [68] proposed a joint solution for localization and time synchronization in mobile underwater sensor networks. The stratification effect of underwater medium was taken into account in localization. Schemes utilizing Autonomous Underwater Vehicles (AUVs) [69], [70] and [71] as beacon nodes result in additional cost to the network.

4.3.2 Topology Control

The problem of topology control for WSNs has been extensively studied in recent years. The topology control scheme presented in [72] and [73] started with neighbor finding, where all nodes transmit at their maximum transmission power. Later, each node computes the minimum transmission power required to maintain network connectivity. A

game theoretic model of topology control to analyze the decentralized interactions among heterogeneous sensors was given in [74]. The connectivity, the success rate and the power consumption were considered to achieve desirable network performances. Zhu et al. [75] took into account the signal to interference plus noise ratio and power efficiency to solve the distributed power control issues in cognitive wireless sensor network with imperfect information. A game-theoretic power control mechanism based on the Hidden Markov Model (HMM) was employed to maximize the network lifetime.

Although there are increasing interests in UWSNs in the past several years, few works have been investigated on approaches for topology control in UWSNs, especially for the localization by using topology control. In [76], a distributed radius determination algorithm is designed for the mobility-based topology control problem. However, the energy consumption of message reception was not considered in this paper. A scale-free network model for calculating edge probability was employed to generate the initial topology randomly by Liu et al. [77]. In order to ensure the connectivity and coverage of the network, two kinds of cluster-heads were constructed by using a topology control strategy based on complex network theory. A Single-Leader-Multi-Follower Stackelberg game, called OLTC scheme, was proposed in [63] to build a localization model with the high coverage and less energy consumption in sparse underwater sensor networks. Trilateration algorithm was employed to localize the sensor node. However, in this paper, sensor node always uses the maximum transmission power to broadcast the 'Request' message, which results in more energy consumption. Besides, only the energy-saving for anchor nodes is considered in OLTC. However, in many scenarios of UWSNs, energy-saving for sensor nodes is much more important, which can directly affect the lifetime of the whole network. Because they have a limited battery and are deployed underwater, which leads to engineering hardship and monetary cost of battery replacement.

4.4 System Model

4.4.1 System Overview

The proposed EELA is implemented in the three Dimension (3D) Underwater Sensor Network (UWSN), where $\{N_a\}$ is the set of anchor nodes deployed on the surface of water and $\{N_s\}$ denotes the set of sensor nodes deployed underwater. All nodes move passively given the water wave and underwater currents. Figure 4.1 depicts the deployment scenario

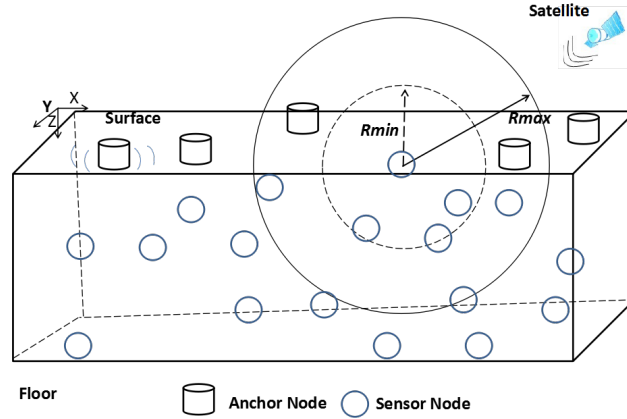


Figure 4.1: Deployment scenario of the proposed EELA scheme.

of EELA, where the cylinders represent anchor nodes on the water surface while dotted circles express sensor nodes which are randomly positioned underwater. Each sensor node i has multiple neighboring anchor nodes within the current transmission range under power P_i , which is expressed by $n_{neig}(P_i)$. Both sensor nodes and anchor nodes can change their transmission power to maximize their own benefits. The transmission power value for each sensor node is within $[0, P_{max}]$, corresponding to different transmission ranges within $[0, R_{max}]$ (see Eq. (4.1)). The transmission power value for each anchor node is within $[0, Q_{max}]$, corresponding to different transmission ranges within $[0, R_{max}]$ (see Eq. (4.1)). I set $P_{max} = Q_{max}$.

For reference purposes, a list of symbols used in the description of the proposed scheme is given in Table 4.1. I have the following assumptions as in [63] in the design of EELA.

- 1) All nodes are time-synchronized.
- 2) Sensor nodes are randomly deployed underwater while anchor nodes randomly float on the water surface.
- 3) Sensor nodes are aware of their depth.

4.4.2 Propagation Model

According to the underwater propagation model [78], the transmission power required by a sending node to a receiving node is given in Eq. (4.1), where P_0 is the received signal

Table 4.1: Symbols used in the proposed EELA scheme.

Parameter	Description
$\{N_s\}$	The set of sensor nodes
$\{N_a\}$	The set of anchor nodes
$\{Nbr_i^s\}$	The set of neighboring sensor nodes of i th sensor node
$\{Nbr_j^a\}$	The set of neighboring anchor nodes of j th anchor node
n_h^{req}	Additional number of anchor nodes required by h th sensor node
E_{tl}	Total remaining energy per node
C_j	Transmission energy cost per unit power of j th anchor node
E_i	Transmission energy cost per unit power of i th sensor node
P_i	Transmission power of i th sensor node
Q_j	Transmission power of j th anchor node
P_{max}	Maximum transmission range of sensor node
Q_{max}	Maximum transmission range of anchor node
OA_j	Localization ability of j th anchor node
OS_i	Ability of sensor node i to find anchor nodes
n_{min}^{req}	Minimum number of required anchor nodes for sensor node to get localized
\tilde{n}_h^{req}	Additional number of anchor nodes required by i th sensor node (fuzzy variable)
\tilde{E}_{tl}	Total current energy of one node (fuzzy variable)
$\sum_{h=1}^{n_{srx}} \tilde{Q}_h$	Sum-transmission power received from other nodes (fuzzy variable)
$\tilde{n}_{hd}(Q_j)$	Number of requests that can be resolved by anchor node j with transmission power Q_j (fuzzy variable)
$\tilde{n}_{neig}(P_i)$	Number of neighbor anchor nodes of i th sensor node (fuzzy variable)
\tilde{E}_{tl}	Total energy of the node (fuzzy variable)
\tilde{OA}_j	Localization ability of the j th anchor node (fuzzy variable)

strength,

$$P = A(R, f) + P_0. \quad (4.1)$$

In general, acoustic communications are used in underwater environment due to the small attenuation of acoustic signals [79]. The attenuation $A(R, f)$ in an underwater acoustic channel for a signal with frequency f over a distance R is given as

$$A(R, f) = A_{norm} R^k a(f)^R, \quad (4.2)$$

where A_{norm} is a normalization constant; R is the distance in meters between the sender and the receiver; f is the frequency; $a(f)$ is the absorption coefficient in dB/m. The spreading factor k is an expression of the geometry of propagation where typically $k = 1.5$ [79]. Eq. (4.3) describes the absorption coefficient with values in dB/km [80],

$$10 \log a(f) = \begin{cases} 0.003 + 0.11 \frac{f^2}{1+f^2} \\ + 44 \frac{f^2}{4100+f^2} + 2.75 \cdot 10^{-4} f^2 & f \geq 0.4 \\ 0.002 + 0.11 \frac{f^2}{1+f^2} \\ + 0.011 f^2 & f < 0.4 \end{cases} \quad (4.3)$$

where T is the Thorp's approximation for absorption loss in dB/km, and f is center frequency in kHz.

Different communication ranges correspond to different bandwidths. For example, if the distance range is 10 km to 100 km, the bandwidth is limited to few kHz. However, 10 kHz matches the short range (from 1 km to 10 km) and a few hundred kHz bandwidth is available for ranges below 100 m [81].

4.5 Underwater Localization by Stackelberg Game Theory

4.5.1 Problem Formulation and Solution

4.5.1.1 Proposed Single Leader Multi-Follower Stackelberg Game Formulation

In Stackelberg game [82], two types of players (leader and followers) are used to model the hierarchy of actions. The leader moves first and selects a strategy. Based on the action of the leader, the followers choose best response strategies that maximize their utilities. Then, the leader selects one strategy to maximize its utility based on the strategies of followers. In a distributed localization scenario, a sensor node can localize itself after receiving enough location beacons from multiple anchor nodes. However, due to the random and sparse node deployment, and the mobility of nodes, a sensor node may not find enough neighbor anchor nodes. A Single-Leader-Multi-Follower Stackelberg game [82] is employed, where the sensor node acts as the single leader while anchor nodes are multiple followers. The sensor node acts first and chooses a transmission power to send a request message, which is similar to the leader releasing a price in Stackelberg

game. Each anchor node reacts, i.e, it selects a transmission power to send reply message, after the action of the sensor node.

4.5.1.2 Utility Function of Anchor Nodes

Anchor nodes are followers. They decide their strategies to handle the maximum number of requests from sensor nodes with minimum energy consumption.

Let n_i^{req} be the additional number of anchor nodes required by sensor node i for localization, which is defined by,

$$n_i^{req} = \begin{cases} n_{min}^{req} - |\mathcal{V}_i|, & \text{if } |\mathcal{V}_i| < n_{min}^{req} \\ 0, & \text{otherwise,} \end{cases} \quad (4.4)$$

where n_{min}^{req} represents the number of anchor nodes required for one sensor node to get its location and \mathcal{V}_i is the set of the anchor nodes in the communication range of sensor node i .

The localization ability $OA_j(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ of anchor node j is composed of several terms expressing different effects,

$$OA_j(Q_j, P_1, P_2, \dots, P_{n_{arx}}) = \frac{n_{hd}(Q_j)}{n_{arx}} + \frac{n_{hd}(Q_j)}{\sum_{k=1}^{n_{arx}} n_k^{req}} - \frac{\sum_{i=1}^{n_{arx}} P_i}{Q_j}. \quad (4.5)$$

In (4.30), the first two terms $\frac{n_{hd}(Q_j)}{n_{arx}}$ and $\frac{n_{hd}(Q_j)}{\sum_{k=1}^{n_{arx}} n_k^{req}}$ are similar to those in the utility function in [63]. $\frac{n_{hd}(Q_j)}{n_{arx}}$ is the ‘ability of j th anchor node to resolve sensor requests’, where $n_{hd}(Q_j)$ is the number of requests that can be handled by anchor node j with transmission power Q_j and n_{arx} is the total number of request messages received from sensor nodes. From Proposition 5 below, we can see that $n_{hd}(Q_j)$ of a follower (anchor node) j is non-decreasing with the increase of the transmission power Q_j . The second term $\frac{n_{hd}(Q_j)}{\sum_{k=1}^{n_{arx}} n_k^{req}}$ is the ‘ability of j th anchor node to serve additional demands’. It means that only $n_{hd}(Q_j)$ requests can be served among the total sum demand for additional anchor nodes $\sum_{k=1}^{n_{arx}} n_k^{req}$ from sensor nodes. Finally, the third term $\frac{\sum_{i=1}^{n_{arx}} P_i}{Q_j}$ expresses the relation between the sum-transmit power received from sensor nodes and the transmission power of the anchor node j . If the transmission power $\sum_{i=1}^{n_{arx}} P_i$ received from sensor nodes increases, anchor node j has to handle more sensor nodes. Therefore, the localization ability $OA_j(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ decreases.

Proposition 1. For each anchor node j , the number of neighboring sensor nodes is higher or at

least equal with the increase of transmission power Q_j .

Proof. Let us assume that there are n sensor nodes uniformly deployed in the area with size d^3 . $n_{hd}(Q_j)$ can be calculated by,

$$n_{hd}(Q_j) = \rho v_j = \frac{4\pi n R_j^3}{3d^3}, \quad (4.6)$$

where ρ is the density of sensor nodes and v_j is the volume of j th anchor node with the transmission range R_j .

According to Eqs. (4.1) and (4.2), the transmission power Q_j of anchor node j is given as,

$$Q_j(R_j) = A_{norm} R_j^k a(f)^{R_j} + Q_{j0}. \quad (4.7)$$

The inverse function $g^{-1}(Q_j) = R_j$ exists and is monotonically strictly increasing with the transmission power Q_j , because $\frac{\partial Q_j(R_j)}{\partial R_j} = A_{norm} \left(k R_j^{k-1} a(f)^{R_j} + R_j^k a(f)^{R_j} \ln a(f) \right) > 0$. Therefore, $n_{hd}(Q_j)$ can be represented by Eq. (4.33). Then, the first order partial derivative of $n_{hd}(Q_j)$ is given in Eq. (4.34),

$$n_{hd}(Q_j) = \frac{4\pi n (g^{-1}(Q_j))^3}{3d^3}, \quad (4.8)$$

$$\frac{\partial n_{hd}(Q_j)}{\partial Q_j} = \frac{4\pi n}{d^3} (g^{-1}(Q_j))^2 \frac{\partial g^{-1}(Q_j)}{\partial Q_j}. \quad (4.9)$$

Hence, $\frac{\partial n_{hd}(Q_j)}{\partial Q_j} > 0$, which proves Proposition 5. \square

Next, I define the payoff function of any anchor node j by considering various factors such as energy cost, the ability to localize sensor nodes and the transmission power of sensor nodes as well as anchor nodes. It is hence defined as the weighted sum of the remaining energy ratio after transmission with Q_j and its localization ability $OA_j(Q_j, P_1, P_2, \dots, P_{n_{arx}})$,

$$U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}}) = w_{1j} \frac{E_j^{tl} - C_j Q_j}{E_j^{tl}} + w_{2j} OA_j(Q_j, P_1, P_2, \dots, P_{n_{arx}}). \quad (4.10)$$

In the first term of Eq. (4.35), E_j^{tl} is the total energy of the j th anchor node and C_j is the transmission energy cost per unit power. Weights w_{1j} and w_{2j} define the trade-off between the energy consumption of anchor node and the localization ability of anchor

node, and satisfy $w_{1j} + w_{2j} = 1$, $w_{kj} \in (0, 1)$.

4.5.1.3 Utility Function of Sensor Nodes

In the considered localization problem, sensor nodes act as leaders. They watch for the decision of anchors which act as followers, and based on the response of followers, maximize their profits. The strategy of the leader is to minimize the energy consumption and to localize the maximum number of sensor nodes during the allowed localization delay, which is defined in Section 4.5.2.3. A sensor node broadcasts a 'Request' message to explore the maximum number of anchors. After sensor nodes receive enough beacon locations from neighbor anchor nodes, it will localize itself.

The 'ability of sensor node i to find anchor nodes' $OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}})$ is composed of two terms,

$$OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}}) = \frac{n_{neig}(P_i)}{n_{srx}} - \frac{\sum_{j=1}^{n_{srx}} Q_j}{P_i}. \quad (4.11)$$

In Eq. (4.36), the first term $\frac{n_{neig}(P_i)}{n_{srx}}$ is the ratio of the number of anchor nodes $n_{neig}(P_i)$ within 'one-hop' and 'two-hop' ranges of transmission power P_i and the total number of anchor nodes n_{srx} received, where $n_{neig}(P_i)$ is a non-decreasing function of P_i , the proof of which is similar to that of Proposition 5. The second term $\frac{\sum_{j=1}^{n_{srx}} Q_j}{P_i}$ is the ratio of the sum-transmission power of received anchor nodes and the transmission power of sensor node i . If the transmission power of anchor nodes $\sum_{j=1}^{n_{srx}} Q_j$ increases, more anchor nodes' beacon messages are received so less anchor nodes can be reached with a given power P_i . Therefore, the 'ability of sensor node i to find anchor nodes' $OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}})$ is inversely proportional to this ratio.

The payoff of any sensor node i increases with the decrease in energy consumption. Also, it increases with the increase of the number of neighbor anchor nodes. In addition, the payoff of the leader decreases with each retry it does to send the 'Request' message. Therefore, the payoff function of sensor node i is defined as Eq. (4.37), which is the weighted sum of its remaining energy ratio and the 'ability of sensor node i to find anchor nodes',

$$U_L(P_i, Q_1, Q_2, \dots, Q_{n_{srx}}) = w_{1i} \frac{E_i^{tl} - E_i P_i}{E_i^{tl}} + w_{2i} OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}}). \quad (4.12)$$

In the first term of Eq. (4.37), E_i^{tl} is the total energy in the i th sensor node and E_i is the transmission energy cost per unit power. Weights w_{1i} and w_{2i} provide a trade-off

between energy consumption and ‘ability to find anchor nodes’, satisfying $w_{1i} + w_{2i} = 1$, $w_{1i}, w_{2i} \in (0, 1)$.

4.5.1.4 Existence and Uniqueness of Stackelberg Nash Equilibrium

The considered game achieves equilibrium when the sensor node (leader) selects the optimal transmission power to get its location with minimum energy consumption while anchor nodes (followers) choose their optimal transmission power to localize the maximum number of sensor nodes with minimum energy cost. At equilibrium, the benefit of each side can not be improved by unilaterally changing its own strategy. To find the Stackelberg equilibrium, each sensor node calculates the best reaction of anchor nodes to each of its mixed strategy and selects the mixed strategy that maximizes its own utility.

1. Best Response Strategy of Anchor Nodes: To define the strategy of the j th anchor node Q_j , the transmission power allocation problem can be cast as the optimization problem formulated below,

$$\begin{aligned}
\max_{Q_j} \quad & U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}}) = w_{1j} \frac{E_j^{tl} - C_j Q_j}{E_j^{tl}} \\
& + w_{2j} \left(\frac{n_{hd}(Q_j)}{n_{arx}} + \frac{n_{hd}(Q_j)}{\sum_{k=1}^{n_{arx}} n_k^{req}} - \frac{\sum_{i=1}^{n_{arx}} P_i}{Q_j} \right) \\
s.t. \quad & \\
& w_{1j} + w_{2j} = 1, \quad w_{1j}, w_{2j} \in (0, 1), \\
& Q_j \in [0, Q_{max}], P_j \in [0, P_{max}], \\
& C_j, n_{arx}, n_k^{req}, E_j^{tl} > 0.
\end{aligned} \tag{4.13}$$

All anchor nodes are non-cooperative. In Proposition 6, the existence of the best response strategy of each anchor node is proved and the unique equilibrium point is computed.

Proposition 2. Let Q_j be the strategy of the j th anchor node. The best response Q_j^* of each anchor node is given as,

$$\begin{aligned}
Q_j^*(P_1, P_2, \dots, P_{n_{arx}}) = \\
\left(\frac{w_{2j} E_j^{tl} \sum_{i=1}^{n_{arx}} P_i}{w_{1j} C_j - Z_j(g^{-1}(Q_j^*))^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j}} \right)^{\frac{1}{2}}, \tag{4.14}
\end{aligned}$$

with $w_F^* < w_{1j} < 1$, $w_{1j} + w_{2j} = 1$ and $w_{2j} \in (0, 1)$, where

$$Z_j = \frac{4(1 - w_{1j})\pi n E_j^{tl}}{d^3} \left(\frac{1}{n_{arx}} + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right), \quad (4.15)$$

$$w_F^* = \frac{d^3 E_j^{tl} \sum_{i=1}^{n_{arx}} P_i + A E_j^{tl} Q_{max}^2 B_j}{d^3 Q_{max}^2 C_j + d^3 E_j^{tl} \sum_{i=1}^{n_{arx}} P_i + A E_j^{tl} Q_{max}^2 B_j}, \quad (4.16)$$

where $A = 4\pi n \left(\frac{1}{n_{arx}} + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right)$ and $B_j = (g^{-1}(Q_j^*))^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j}$.

Proof. I prove that problem (4.13) is a standard form of convex optimization and determine the expression of the optimum.

The first order partial derivative of $U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ with respect to Q_j , for $j \in [1, N]$, is given as

$$\begin{aligned} \frac{\partial U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}})}{\partial Q_j} &= \frac{-w_{1j} C_j}{E_j^{tl}} + w_{2j} \left(\frac{\sum_{i=1}^{n_{arx}} P_i}{Q_j^2} \right. \\ &\quad \left. + \frac{4\pi n}{d^3} \left(\frac{1}{n_{arx}} + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right) (g^{-1}(Q_j))^2 \frac{\partial g^{-1}(Q_j)}{\partial Q_j} \right). \end{aligned}$$

Let $\frac{\partial U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}})}{\partial Q_j} = 0$, and denoting its solution as Q_j^* , I get Q_j^{*2} as,

$$Q_j^{*2}(P_1, P_2, \dots, P_{n_{arx}}) = \frac{(1 - w_{1j}) E_j^{tl} \sum_{i=1}^{n_{arx}} P_i}{w_{1j} C_j - Z_j (g^{-1}(Q_j^*))^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j}},$$

where $Z_j = \frac{4(1 - w_{1j})\pi n E_j^{tl}}{d^3} \left(\frac{1}{n_{arx}} + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right)$, hence

$$Q_j^*(P_1, P_2, \dots, P_{n_{arx}}) = \left(\frac{(1 - w_{1j}) E_j^{tl} \sum_{i=1}^{n_{arx}} P_i}{w_{1j} C_j - Z_j (g^{-1}(Q_j^*))^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j}} \right)^{\frac{1}{2}}. \quad (4.17)$$

For existence of Q_j^* , the condition $w_{1j} C_j - Z_j (g^{-1}(Q_j^*))^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j} > 0$ should be

satisfied. This is guaranteed by the condition $w'_F < w_{1j} < 1$, where w'_F is given by,

$$w'_F = \frac{\frac{Z_j}{1-w_{1j}} \left(g^{-1}(Q_j^*)\right)^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j}}{\frac{Z_j}{1-w_{1j}} \left(g^{-1}(Q_j^*)\right)^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j} + C_j}. \quad (4.18)$$

First, I show that the objective function of Eq. (4.13) is concave. The second order partial derivative of $U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ is given as

$$\begin{aligned} \frac{\partial^2 U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}})}{\partial Q_j^2} &= \frac{4\pi n w_{2j}}{d^3} \left(\frac{1}{n_{arx}} \right. \\ &\quad \left. + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right) \left(2g^{-1}(Q_j) \left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j} \right)^2 \right. \\ &\quad \left. + (g^{-1}(Q_j))^2 \frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2} \right) - \frac{2w_{2j} \sum_{i=1}^{n_{arx}} P_i}{Q_j^3}. \end{aligned}$$

Here, I show that $G(Q_j) < 0$ in order to prove that $\frac{\partial^2 U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}})}{\partial Q_j^2}$ is negative, where

$$G(Q_j) = \left(2g^{-1}(Q_j) \left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j} \right)^2 + (g^{-1}(Q_j))^2 \frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2} \right). \quad (4.19)$$

Firstly, I prove $\frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2} < 0$. According to Eqs. (4.1) and (4.2), the transmission power Q_j of anchor node j is given as

$$Q_j(R_j) = A_{norm} R_j^k a(f)^{R_j} + Q_{j0}.$$

The inverse function $g^{-1}(Q_j) = R_j$ of anchor node j exists and is strictly monotonically increasing with the transmission power Q_j , i.e., $\frac{\partial g^{-1}(Q_j)}{\partial Q_j} > 0$, because $\frac{\partial Q_j(R_j)}{\partial R_j} = A_{norm} \left(k R_j^{k-1} a(f)^{R_j} + R_j^k a(f)^{R_j} \ln a(f) \right) > 0$ and $\frac{\partial Q_j(R_j)}{\partial R_j} = \frac{1}{\frac{\partial g^{-1}(Q_j)}{\partial Q_j}}$.

Then, $\frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2}$ can be calculated as

$$\begin{aligned}\frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2} &= -\frac{\frac{\partial^2 Q_j(R_j)}{\partial R_j^2} \frac{\partial g^{-1}(Q_j)}{\partial Q_j}}{\left(\frac{\partial Q_j(R_j)}{\partial R_j}\right)^2} \\ &= -\frac{\partial^2 Q_j(R_j)}{\partial R_j^2} \left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j}\right)^3.\end{aligned}\quad (4.20)$$

Then, I have

$$\begin{aligned}\frac{\partial^2 Q_j(R_j)}{\partial R_j^2} &= A_{norm} \left(k(k-1)R_j^{k-2}a(f)^{R_j} \right. \\ &\quad \left. + kR_j^{k-1}a(f)^{R_j} \ln a(f) + kR_j^{k-1}a(f)^{R_j} \ln a(f) \right. \\ &\quad \left. + R_j^k a(f)^{R_j} \ln^2 a(f) \right) > 0,\end{aligned}\quad (4.21)$$

from which I get for (4.43) $\frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2} < 0$.

Secondly, I have $2g^{-1}(Q_j) < (g^{-1}(Q_j))^2$, because $g^{-1}(Q_j) = R_j$ and $R_j \gg 1$. In order to prove $G(Q_j) < 0$ given in Eq. (4.19), I need to prove $\left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j}\right)^2 < \left|\frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2}\right|$. According to Eq. (4.43),

$$\begin{aligned}&\left|\frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2}\right| - \left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j}\right)^2 \\ &= \left|\frac{\partial^2 Q_j(R_j)}{\partial R_j^2} \left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j}\right)^3\right| - \left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j}\right)^2 \\ &= \left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j}\right)^2 \left(\left|\frac{\partial^2 Q_j(R_j)}{\partial R_j^2} \frac{\partial g^{-1}(Q_j)}{\partial Q_j}\right| - 1\right),\end{aligned}$$

where $\frac{\partial^2 Q_j(R_j)}{\partial R_j^2}$ is given by Eq. (4.43) and $\frac{\partial g^{-1}(Q_j)}{\partial Q_j}$ as,

$$\frac{\partial g^{-1}(Q_j)}{\partial Q_j} = \frac{1}{A_{norm} \left(kR_j^{k-1}a(f)^{R_j} + R_j^k a(f)^{R_j} \ln a(f) \right)}.\quad (4.22)$$

From Eq. (4.3), I know $\ln a(f) \gg 1$, from which I get $\frac{\partial^2 Q_j(R_j)}{\partial R_j^2} \frac{\partial g^{-1}(Q_j)}{\partial Q_j} \approx \frac{2k}{R_j} + 1 > 1$,

where $k = 1.5$ and $R_j > 0$. Therefore, $\left(\frac{\partial g^{-1}(Q_j)}{\partial Q_j}\right)^2 < \left|\frac{\partial^2 g^{-1}(Q_j)}{\partial Q_j^2}\right|$ is proved.

Since the value of the second order partial derivative of $U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ is negative, the objective function in Eq. (4.13) is concave, and hence achieves its maximum at Q_j^* in Eq. (4.17).

Next, I prove that $Q_j^*(P_1, P_2, \dots, P_{n_{arx}})$ is in $[0, Q_{max}]$. From Eq. (4.17), I have $Q_j^* > 0$. I show that $Q_j^*(P_1, P_2, \dots, P_{n_{arx}}) - Q_{max} < 0$ or equivalently,

$$\begin{aligned} & Q_j^*(P_1, P_2, \dots, P_{n_{arx}}) - Q_{max} < 0 \\ \Rightarrow & \left(\frac{(1 - w_{1j})E_j^{tl} \sum_{i=1}^{n_{arx}} P_i}{w_{1j}C_j - Z_j(g^{-1}(Q_j^*))^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j}} \right)^{\frac{1}{2}} - Q_{max} < 0, \end{aligned}$$

which is verified if

$$w_{1j} > w_F^* = \frac{d^3 E_j^{tl} \sum_{i=1}^{n_{arx}} P_i + A E_j^{tl} Q_{max}^2 B_j}{d^3 Q_{max}^2 C_j + d^3 E_j^{tl} \sum_{i=1}^{n_{arx}} P_i + A E_j^{tl} Q_{max}^2 B_j}.$$

To ensure both the existence of $Q_j^*(P_1, P_2, \dots, P_{n_{arx}})$, i.e., $w_F' < w_{1j} < 1$ and the requirement that $Q_j^*(P_1, P_2, \dots, P_{n_{arx}}) \in [0, Q_{max}]$, it is sufficient to set $w_F' < w_F^*$. Hence, $\frac{w_F'}{w_F^*}$ is calculated as,

$$\frac{w_F'}{w_F^*} = \frac{H_j}{H_j + C_j d^6 E_j^{tl} \sum_{i=1}^{n_{arx}} P_i} \quad (4.23)$$

where H_j is defined as,

$$\begin{aligned} H_j &= \left(A E_j^{tl} Q_{max} B_j \right)^2 + A d^3 (E_j^{tl})^2 \sum_{i=1}^{n_{arx}} P_i B_j \\ &\quad + A C_j d^3 E_j^{tl} Q_{max}^2 B_j \end{aligned}$$

where $A = 4\pi n \left(\frac{1}{n_{arx}} + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right)$ and $B_j = (g^{-1}(Q_j^*))^2 \frac{\partial g^{-1}(Q_j^*)}{\partial Q_j}$.

From Eq. (4.23), I confirm that $\frac{w_F'}{w_F^*} < 1$. Thus, the maximum value of $U_F(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ can be achieved at $Q_j^*(P_1, P_2, \dots, P_{n_{arx}})$ in $[0, Q_{max}]$ provided $w_F' < w_{1j} < 1$, proving Proposition 6. \square

2. Best Response Strategy of Sensor Node: To define the strategy of the i th sensor node P_i , the transmission power allocation problem can be cast as the optimization problem formulated below,

$$\begin{aligned} \max_{P_i} U_L(P_i, Q_1, Q_2, \dots, Q_{n_{srx}}) &= w_{1i} \frac{(E_i^{tl} - E_i P_i)}{E_i^{tl}} \\ &+ w_{2i} \left(\frac{n_{neig}(P_i)}{n_{srx}} - \frac{\sum_{j=1}^{n_{srx}} Q_j}{P_i} \right) \end{aligned} \quad (4.24)$$

s.t.

$$\begin{aligned} w_{1i} + w_{2i} &= 1, \quad w_{1i}, w_{2i} \in (0, 1), \\ P_i &\in [0, P_{max}], Q_h \in [0, Q_{max}], \\ n_{srx}, E_i, E_i^{tl} &> 0. \end{aligned}$$

The existence and uniqueness of the sensor node's transmission power at Nash equilibrium is proved in Proposition 7.

Proposition 3. Let P_i be the strategy of the i th sensor node. The best response P_i^* of each sensor node is given as,

$$\begin{aligned} P_i^*(Q_1, Q_2, \dots, Q_{srx}) &= \\ &\left(\frac{w_{2i} d^3 E_i^{tl} n_{srx} \sum_{j=1}^{n_{srx}} Q_j}{w_{1i} A_i - Z_i (g^{-1}(P_i^*))^2 \frac{\partial g^{-1}(P_i^*)}{\partial P_i}} \right)^{\frac{1}{2}}, \end{aligned} \quad (4.25)$$

with $w_L^* < w_{1i} < 1$, $w_{1i} + w_{2i} = 1$ and $w_{2i} \in (0, 1)$, where $A_i = d^3 E_i n_{srx}$, $Z_i = 4\pi n w_{2i} E_i^{tl}$ and

$$w_L^* = \frac{1 + 4\pi n E_i^{tl} B_i P_{max}^2}{d^3 E_i^{tl} n_{srx} \sum_{j=1}^{n_{srx}} Q_j + A_i P_{max}^2 + 4\pi n E_i^{tl} B_i P_{max}^2}, \quad (4.26)$$

where $A_i = d^3 E_i n_{srx}$ and $B_i = (g^{-1}(P_i^*))^2 \frac{\partial g^{-1}(P_i^*)}{\partial P_i}$.

Proof. I prove that problem (4.24) is a standard form of convex optimization and determine the expression of the optimum.

The first order partial derivative of $U_L(P_i, Q_1, Q_2, \dots, Q_{n_{srx}})$ with respect to P_i is

given as

$$\frac{\partial U_L(P_i, Q_1, Q_2, \dots, Q_{n_{sr x}})}{\partial P_i} = -\frac{w_{1i} E_i}{E_i^{tl}} + w_{2i} \left(\frac{4\pi n}{d^3 n_{sr x}} (g^{-1}(P_i))^2 \frac{\partial g^{-1}(P_i)}{\partial P_i} + \frac{\sum_{j=1}^{n_{sr x}} Q_j}{P_i^2} \right).$$

Letting $\frac{\partial U_L(P_i, Q_1, Q_2, \dots, Q_{n_{sr x}})}{\partial P_i} = 0$, and denoting its solution as P_i^* , I get

$$P_i^{*2}(Q_1, Q_2, \dots, Q_{sr x}) = \frac{w_{2i} d^3 E_i^{tl} n_{sr x} \sum_{j=1}^{n_{sr x}} Q_j}{w_{1i} A_i - Z_i (g^{-1}(P_i^*))^2 \frac{\partial g^{-1}(P_i^*)}{\partial P_i}},$$

where $A_i = d^3 E_i n_{sr x}$ and $Z_i = 4\pi n w_{2i} E_i^{tl}$.

To guarantee the existence of $P_i^{*2}(Q_1, Q_2, \dots, Q_{sr x})$, I need to set the condition $w_{1i} A_i - Z_i (g^{-1}(P_i^*))^2 \frac{\partial g^{-1}(P_i^*)}{\partial P_i} > 0$. Then I get $w'_L < w_{1i} < 1$, where w'_L is calculated as,

$$w'_L = \frac{4\pi n E_i^{tl} (g^{-1}(P_i^*))^2 \frac{\partial g^{-1}(P_i^*)}{\partial P_i}}{d^3 E_i n_{sr x} + 4\pi n E_i^{tl} (g^{-1}(P_i^*))^2 \frac{\partial g^{-1}(P_i^*)}{\partial P_i}}. \quad (4.27)$$

First, I show that the objective function of Eq. (4.24) is concave. The second order partial derivative of $U_L(P_i, Q_1, Q_2, \dots, Q_{n_{sr x}})$ is given as

$$\begin{aligned} \frac{\partial^2 U_L(P_i, Q_1, Q_2, \dots, Q_{n_{sr x}})}{\partial P_i^2} &= \frac{4\pi n w_{2i}}{d^3 n_{sr x}} (2g^{-1}(P_i) \\ &\quad \left(\frac{\partial g^{-1}(P_i)}{\partial P_i} \right)^2 + (g^{-1}(P_i))^2 \frac{\partial^2 g^{-1}(P_i)}{\partial P_i^2}) \\ &\quad - \frac{2w_{2i} \sum_{j=1}^{n_{sr x}} Q_j}{P_i^3}. \end{aligned}$$

Here, I use the fact that $G(Q_j) < 0$ given in Eq. (4.19) as proved for Proposition 6. Therefore, the value of the second order partial derivative of $U_L(P_i, Q_1, Q_2, \dots, Q_{n_{sr x}})$ is negative.

Next, I prove that $P_i^*(Q_1, Q_2, \dots, Q_{sr x})$ is in $[0, P_{max}]$. From Eq. (4.46), I have $P_i^* > 0$

. I show that $P_i^*(Q_1, Q_2, \dots, Q_{srx}) - P_{max} < 0$ or equivalently,

$$\begin{aligned} & P_i^*(Q_1, Q_2, \dots, Q_{srx}) - P_{max} < 0 \\ \Rightarrow & \left(\frac{(1 - w_{1i})d^3 E_i^{tl} n_{srx} \sum_{j=1}^{n_{srx}} Q_j}{w_{1i} A_i - Z_i (g^{-1}(P_i^*))^2 \frac{\partial g^{-1}(P_i^*)}{\partial P_i}} \right)^{\frac{1}{2}} - P_{max} < 0, \end{aligned}$$

where $A_i = d^3 E_i n_{srx}$ and $B_i = (g^{-1}(P_i^*))^2 \frac{\partial g^{-1}(P_i^*)}{\partial P_i}$, which is verified if

$$w_{1i} > w_L^* = \frac{1 + 4\pi n E_i^{tl} B_i P_{max}^2}{d^3 E_i^{tl} n_{srx} \sum_{j=1}^{n_{srx}} Q_j + A_i P_{max}^2 + 4\pi n E_i^{tl} B_i P_{max}^2}.$$

Similarly as for Proposition 2, I can prove that $\frac{w_L'}{w_L^*} < 1$ to get the condition $w_L^* < w_{1i} < 1$ in order to guarantee the existence of $P_i^*(Q_1, Q_2, \dots, Q_{srx})$ and that $P_i^*(Q_1, Q_2, \dots, Q_{srx}) \in [0, P_{max}]$. Thus, the maximum value of $U_L(P_i, Q_1, Q_2, \dots, Q_{n_{srx}})$ can be achieved at $P_i^*(Q_1, Q_2, \dots, Q_{srx})$ in $[0, P_{max}]$ provided $w_F^* < w_{1i} < 1$, proving Proposition 7, for the existence and uniqueness of the Nash equilibrium of the considered Stackelberg game. \square

4.5.1.5 Algorithm Design

I now design the proposed EELA algorithm, where sensor nodes localize themselves once they receive enough location beacon information from neighboring anchor nodes. The proposed algorithm consists of several phases.

- Phase 1: each sensor node builds a neighbor list containing the ‘Wakeup’ (Type, ID, Time) message received from its neighbor anchor nodes. Each anchor node also builds its neighbor anchor list with its received ‘Wakeup’ message from its neighbor anchor nodes.
- Phase 2: anchor nodes which received ‘Wakeup’ messages from neighbor anchor nodes, broadcast their neighbor anchor list by using ‘AnchorNbr’ (Type, ID, Time, NbrAnchorNodes) message. Each sensor node updates its neighbor list information and adds the anchor node’s neighbor information. The game starts at the third phase of nodes communication.
- Phase 3: to start the opportunistic localization, each sensor node explores its maximum opportunities with the consideration of energy consumption and neighbor

anchor nodes by the procedures described in Algorithm 4. ‘One-hop’ neighbor anchor nodes are considered first. If the number of ‘one-hop’ neighbor anchor nodes is enough to localize the sensor node, it will not handle the ‘two-hop’ neighbors. This is because anchor nodes in ‘one-hop’ neighbor list have more accurate information, such as the one-way time delay. However, due to the node mobility and random deployment, ‘two-hop’ anchor nodes should be considered given the few ‘one-hop’ anchor nodes in UWSNs. Figure 4.2 depicts the initial ‘two-hop’ transmission power calculation, where AN_1 and AN_2 (black circles), which are ‘one-hop’ and ‘two-hop’ anchor nodes respectively, act as multiple followers. SN_3 is the sensor node acting as the single leader. If SN_3 uses the maximum transmission power P_{max} to have a transmission radius of R_{max} , OA is the opportunistic localization range. Proposition 4 is used to evaluate the transmission power required to reach the ‘two-hop’ anchor nodes.

Proposition 4. *Let p_{31} and p_{32} be the transmission powers of sensor node SN_3 required to send a Request message to anchor nodes AN_1 and AN_2 , respectively. Let q_{12} be the transmission power required at AN_1 to reach AN_2 . Then, if anchor nodes AN_1 and AN_2 are in the ‘one-hop’ and ‘two-hop’ neighbor list of SN_3 , respectively, I can set $p_{32} < p_{31} + q_{12}$. Moreover, node AN_2 in the ‘two-hop’ neighbor list of SN_3 can be moved to the ‘one-hop’ neighbor list, if I use $p_3 = p_{31} + q_{12}$ as the transmission power of SN_3 .*

Remark 1: q_{12} can be known at sensor node SN_3 by estimating the distance d_{12} from the received anchor nodes’ messages.

Remark 2: The final optimal transmission power of the sensor node will be selected by Proposition 7.

Proof. From Eq. (4.1), the transmission power $P(d)$ is an increasing function of the distance d . Given the triangle inequality $d_{32} < d_{31} + d_{12}$, and since q_{12} and p_{31} are sufficient to cover distances d_{12} and d_{31} respectively, the ‘two-hop’ neighbor anchor node AN_2 becomes a ‘one-hop’ neighbor anchor node by setting $p_3 = p_{31} + q_{12}$ as the new initial transmission power. \square

If I have multiple ‘two-hop’ anchor nodes, Proposition 4 is applied sequentially, until the required number of nodes is reached.

- **Phase 4:** After anchor nodes receive the ‘Request’ (Type, ID, Time, n_{req}) message from sensor nodes, an optimal transmission power will be selected by Proposition 6

Algorithm 4: Topology control of a sensor node (leader)

Input : $\{N_s\}, \{N_a\}, P_{ini}, \{Nbr_i^s\}, n_{min}^{req}$
Output: Optimized action P_i'
 $P_i' = 0, P_i = P_{ini}, U_i^{L*} = -\infty.$
for each message received from an anchor node $j \in \{Nbr_i^s\}$ **do**
 if $j \in \{Nbr_i^s\}^{one-hop'}$ **then**
 | Add j to $\mathcal{V}_i^{one-hop'}$.
 else
 | Add j to $\mathcal{V}_i^{two-hop'}$.
 end
end
if $|\mathcal{V}_i^{one-hop'}| \geq n_{min}^{req}$ **then**
 for each anchor node $j \in \mathcal{V}_i^{one-hop'}$ **do**
 | Calculate the utility U_i^L .
 end
 $P_i' \leftarrow \operatorname{argmax} U_i^L.$
else
 if $0 \leq (|\mathcal{V}_i^{one-hop'}| + |\mathcal{V}_i^{two-hop'}|) < n_{min}^{req}$ **then**
 | $n_i^{req} \leftarrow (n_{min}^{req} - |\mathcal{V}_i|),$
 | $P_i' \leftarrow P_{max}.$
 else
 for each anchor node $j \in \mathcal{V}_i$ **do**
 | Calculate the utility U_i^L .
 end
 $P_i \leftarrow \operatorname{argmax} U_i^L.$
 if $P_i > P_{max}$ **then**
 | $P_i' \leftarrow P_{max}.$
 else
 | $P_i' \leftarrow P_i.$
 end
 end
end
end
Broadcast 'Request' message at transmission power P_i' .

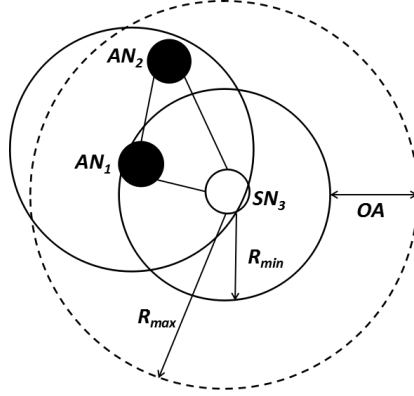


Figure 4.2: A scenario depicting the 'two-hop' transmission power calculation.

to broadcast the 'Reply' (Type, ID, Time, Location) message taking into account the factors of energy consumption and the ability to localize sensor node. The detailed steps are given in Algorithm 5.

Algorithm 5: Topology control of an anchor node (follower)

Input : $\{N_a\}, \{N_s\}, Q_{ini}, \{Nbr_j^a\}$

Output: Optimized action Q_j'

$Q_j' = 0, Q_j = Q_{ini}$.

Broadcast 'Wakeup' message at transmission power Q_j .

for each 'Wakeup' message received from each anchor node **do**

 | Build its neighbor anchor list $\{N_j^a\}$.

end

Broadcast 'AnchorNbr' message at the transmission power Q_j .

for each 'Request' message received from each sensor node $g \in \{Nbr_j^a\}$ **do**

 | Calculate the utility U_F^j .

end

$Q_j' \leftarrow \text{argmax } U_F^j$.

Broadcast 'Reply' message at transmission power Q_j' .

Finally, after sensor nodes receive the required number of beacon location information from its neighboring anchor nodes, they execute their localization procedure. Since the main purpose of this chapter is energy efficiency improvement of localization by topology control, I will assume the trilateration technique [83] in the next section, for node localization to illustrate the proposed EELA. In trilateration technique, each sensor node requires three anchor nodes in order to obtain its location, i.e., $n_{min}^{req} = 3$.

Parameter	Value
Node mobility model	Meandering current mobility model [86]
Channel Frequency	22 kHz [63]
Modulation technique	<i>FSK</i> [63]
Data rate	500 bps [63]
Speed of sound	1500 m s ⁻¹ [63]
Wave propagation model	Thorp's propagation model [85]
Receive and Idle power	0.1 watts
Sleep power	1 × 10 ⁻⁴ watts

Table 4.2: Simulation parameters.

4.5.2 Numerical Evaluations

4.5.2.1 Simulation Settings

I consider a network of 10-50 sensor nodes in a 3D underwater region of 2500 m × 2500 m × 2500 m with the proposed anchor nodes on the water surface. In the proposed simulation, the transmission range R_i is a continuous number in $(0, max_range]$. I set the $max_range = \sqrt{2500^2 + 2500^2 + 2500^2}m$ which is the max distance in the simulation region. Initially, the value of all nodes' transmission range are R_{ini} . For each simulation, Underwater Acoustic Network (UAN) models of NS-3 are utilized for generating the channels, PHY and MAC layers, as they are commonly used for modeling underwater networks [84]. The transmission power P_i or Q_j (TxPower attribute in UanPhyGen model) is initially set for a given range R_i or R_j using the Thorp's propagation model [85]. Weights of anchor nodes' utility function are taken as $w_{1j} = 0.4$ and $w_{2j} = 0.6$. On the other hand, sensor nodes' weights w_{1i} and w_{2i} are set to 0.1 and 0.9. This is because maximizing the ability of finding anchor nodes is crucial, while setting $w_{1i} = 0.1$ for energy consumption still ensures very high energy-efficiency, and as will show by the simulation results. I select the values of weights in the pre-process step according to the performance of EELA. Other simulation parameters are listed in Table 4.2.

In the simulations, sensor nodes are randomly deployed under water while anchor nodes are randomly deployed on the water surface in the simulation area. Any sensor node gets localized after receiving n_{min}^{req} number of replies from anchor nodes. All nodes move according to the velocity of ocean current, following the Meandering Current Mobility (MCM) model [86]. In MCM, the effect of the meandering sub-surface currents and velocities are considered for nodes moving. The sensor nodes mobility (v_m) is set

to 2.0 m s^{-1} , 3.0 m s^{-1} , 4.0 m s^{-1} [63]. Each simulation runs for 5000 times to obtain the average results.

4.5.2.2 Baseline Schemes

In the proposed EELA scheme, both anchor and sensor nodes can select their optimal transmission range to communicate with each other. The performance of the proposed scheme is compared to the five schemes listed below.

- 1) OLTC [63]: An anchor node dynamically selects a transmission range from $[0, Q_{max}]$ in order to maximize the number of neighbors yet to be localized. Only anchor nodes can adjust their transmission range while sensor nodes always use the maximum transmission range to send messages.
- 2) 3DUL-Min [64]: Surface anchor nodes initiates iterative three-dimensional localization scheme. Successful localized sensor nodes transform to anchor nodes to help other sensor nodes to localize themselves. Both anchor as well as sensor nodes use the fixed minimum transmission range to broadcast message.
- 2) 3DUL-Max [64]: Surface anchor nodes initiates iterative three-dimensional localization scheme. Successful localized sensor nodes transform to anchor nodes to help other sensor nodes to localize themselves. Both anchor as well as sensor nodes use the fixed maximum transmission range to broadcast message.
- 3) EELA-Min: the scheme without the dynamic transmission power optimization, i.e., both anchor and sensor nodes use the fixed minimum transmission range to broadcast message.
- 4) EELA-Max: the scheme without the dynamic transmission power optimization, i.e., both anchor and sensor nodes use the fixed maximum transmission range to broadcast message.

4.5.2.3 Performance Metrics

The following metrics are adopted to evaluate the performance of EELA.

- 1) Localization coverage: the ratio of the number of localized sensor nodes to the total

number of sensor nodes,

$$C = \frac{|\{N_{s,l}\}|}{|\{N_s\}|},$$

where $\{N_{s,l}\}$ is the set of sensor nodes which already have obtained their locations and the set of sensor node is $\{N_s\}$.

- 2) Average energy consumption per sensor node: the ratio of total energy consumption of sensor nodes to the number of sensor nodes,

$$\varepsilon_{\{N_{s,l}\}}^{avg} = \frac{\sum_{i=1}^{|\{N_{s,l}\}|} \varepsilon_i}{|\{N_{s,l}\}|}.$$

- 3) Average energy consumption per anchor node: the ratio of total energy consumption of anchor nodes to the number of anchor nodes,

$$\varepsilon_{\{N_a\}}^{avg} = \frac{\sum_{j=1}^{|\{N_a\}|} \varepsilon_j}{|\{N_a\}|},$$

where the set of anchor node is $\{N_a\}$.

- 4) Average energy consumption per node: the ratio of the total energy consumption of all nodes to the number of all nodes,

$$\varepsilon_{tl}^{avg} = \frac{\sum_{i=1}^{|\{N_s\}|} \varepsilon_i + \sum_{j=1}^{|\{N_a\}|} \varepsilon_j}{|\{N_s\}| + |\{N_a\}|}.$$

- 5) Average localization error,

$$\frac{\sum_{i=1}^{|\{N_{s,l}\}|} \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2 + (z_i - z'_i)^2}}{|\{N_{s,l}\}|}.$$

where for any localized sensor node node i , (x_i, y_i, z_i) and (x'_i, y'_i, z'_i) denote the original and the estimated locations, respectively.

- 6) Average localization delay: the time duration from a sensor node broadcasting a 'Request' message to the time of obtaining its location.

4.5.2.4 Results and Analysis

I compare the performance of EELA with that of OLTC [63], EELA-min, and EELA-max for UWSNs. For each simulation, the number of deployed sensor nodes is varied between 10 to 50. Each simulation runs for 5000 times to obtain the average results.

- 1) Localization Coverage: In Fig. 4.3, the average localization coverage in function of the number of sensor nodes is presented.

I observe that the average localization coverage of both EELA and OLTC are between that of EELA-Max and EELA-Min, as expected. 3DUL-Max [64] has the same performance with that of EELA-Max while 3DUL-Min [64] performs better than EELA. The proposed EELA scheme outperforms the reference OLTC scheme of [63]. For all schemes, an increased number of sensor nodes results in a better localization coverage. This is because the increase of the number of sensor nodes entails a higher spatial density, hence more sensor nodes may be localized by anchor nodes with a given power.

Compared to OLTC, the localization coverage achieved by EELA is 2.0 % higher on average. This is because in OLTC scheme, sensor nodes always send request with the maximum transmission power, which leads to a higher rate of packet collision. Thus, anchor nodes will receive fewer 'Request' messages. However, anchor nodes use the optimal transmission power $P_j < P_{max}$ to reply, so that some sensor nodes may not receive the required number of anchor nodes, hence decreasing coverage. By contrast, in EELA, sensor nodes request with the optimal transmission power $P_j < P_{max}$ instead of using the maximum transmission power, so that the required number of anchor nodes may be reached with minimum energy consumption. On the other hand, anchor nodes also utilize the optimal transmission power to reply. Both the optimal transmission power for anchor node and sensor node are selected to reach Stackelberg Nash Equilibrium in section 4.6.1.4. Hence, both anchor nodes and sensor nodes cannot improve their individual profit by single-sidedly changing their transmission power. Therefore, EELA can reduce the rate of packet collision, and achieve a higher coverage.

Compared to EELA-Min, the localization coverage achieved in EELA is about 54 % higher on average, because sensor nodes always use the minimum transmission power to send 'Request' messages while anchor nodes also use the minimum transmission power to reply, hence each sensor node receives the fewest beacon location

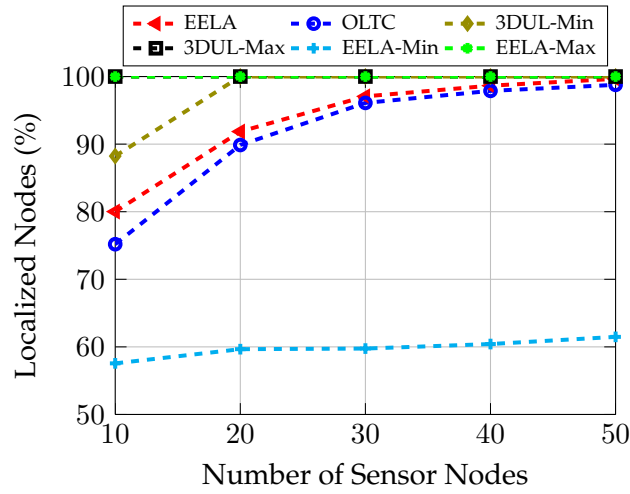


Figure 4.3: Localization coverage.

messages to localize itself. Next, the localization coverage achieved in EELA-Max is about 9% (on average) higher than that in EELA. For EELA-Max, although the ‘Request’ messages from sensor nodes have higher probability to have collisions, anchor nodes always use the maximum transmission power to send ‘Reply’ messages without considering received requests. Therefore, sensor nodes can always receive more beacon locations than those of other schemes. That is the reason why EELA-Max always has the highest coverage. However, the use of higher transmission power leads to higher energy consumption, as shown next. In 3DUL-Max and 3DUL-Min, the localized sensor nodes can be used as new anchor nodes to help other sensor nodes to get their locations, which leads to higher localization coverage. However, the energy consumption for sensor nodes is very high and the localization error as well as delay are accumulated due to the iterative localization design in 3DUL.

- 2) Average Energy Consumption Per Sensor Node: The average energy consumption results for sensor nodes are given in Fig. 4.4.

I observe that the performance of the proposed EELA scheme is between that of EELA-Min and EELA-Max, while OLTC has the same consumption as EELA-Max since all sensor nodes transmit with maximum power. 3DUL-Max consumes the highest energy for sensor nodes among all the schemes because after sensor nodes of 3DUL-Max get their locations, sensor nodes can transform to change to anchor nodes to help other sensor nodes to get their locations in order to achieve high

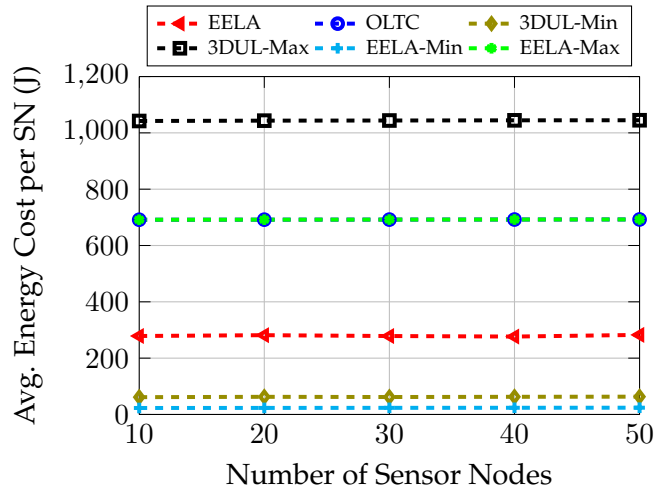


Figure 4.4: Average energy consumption per Sensor Node (SN).

coverage. In a similar way, the energy consumption of 3DUL-Min is higher than that of EELA-Min. In OLTC, EELA-Min, EELA-Max, 3DUL-Min and 3DUL-Max, the average energy consumption is not affected by the node density of sensor nodes because of the fixed transmission power. The variations for proposed EELA are also steady due to the strategy of the proposed game and the constant number of anchor nodes in this simulation setting.

From Fig. 4.4, I notice that the average energy consumption per sensor node in EELA (326 J) is about 53 % lower than that in OLTC (693 J). This is thanks to our transmission power optimization strategy given the ‘one-hop’ and ‘two-hop’ nodes, which enables to reach the same number of anchor nodes as by OLTC, but with much lower energy. This mechanism significantly reduces the energy consumption of sensor nodes, and reduces the collisions of requests at the same time.

- 3) Average Energy Consumption Per Anchor Node: Fig. 4.5 illustrates the average energy consumption of anchor nodes in function of the number of sensor nodes.

I observe that the energy cost of proposed EELA and OLTC lies between that of EELA-Min and EELA-Max, with a higher consumption for EELA compared to OLTC. 3DUL-Max consumes roughly the same energy with EELA-Max while 3DUL-Min consumes slightly higher energy than EELA-Min. Namely, we can see that the average energy consumption per anchor node in EELA (about 407 J) is nearly 38 % higher than that in OLTC (about 295 J). This is because anchor nodes in EELA need

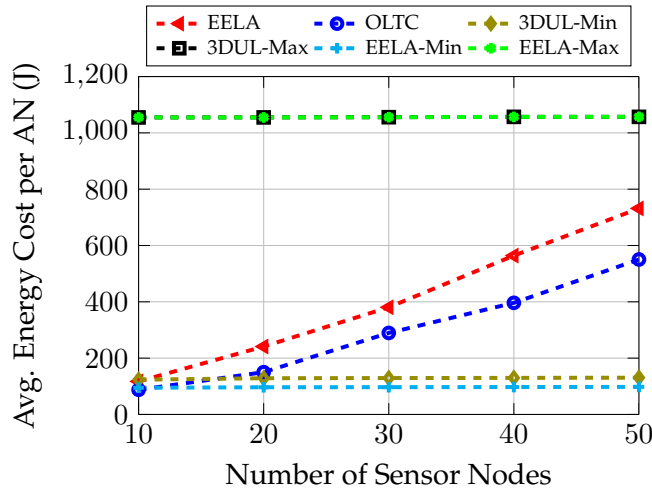


Figure 4.5: Average energy consumption per Anchor Node (AN).

to broadcast twice in order to build their ‘two-hop’ anchor neighboring list. As for OLTC, anchor nodes do not need to consider about their ‘two-hop’ anchor neighboring nodes. As shown next, EELA slightly increases the energy consumption of anchor nodes in order to improve the performance of sensor nodes, which eventually improves the average energy consumption of all nodes. Note also that saving energy of underwater sensor nodes is a more crucial issue than that for anchor nodes, since anchor nodes are specifically deployed at the surface for enabling localization.

- 4) Average Energy Consumption for Any Node: Fig. 4.6 shows the average energy consumption over anchor and sensor nodes.

Overall, I observe that 3DUL-Max consumes the highest energy among all the schemes. The proposed scheme largely reduces the average energy cost per node, compared to OLTC, i.e., around 48 % reduction. This shows that even if anchor nodes broadcast twice in the preprocessing phase, the proposed EELA still consumes much less energy in total, thanks to the energy-efficient power selection of sensor nodes. This is because in the deployment of practical localization systems, the number of sensor nodes is much larger than that of anchor nodes.

Note also that as shown in [87], the main source of energy consumption in underwater sensor nodes is transmission power, compared to any other functionality. Since the proposed and baseline schemes have similar algorithm complexities as shown in Section ??, entailing similar power consumption for processing, I can conclude that

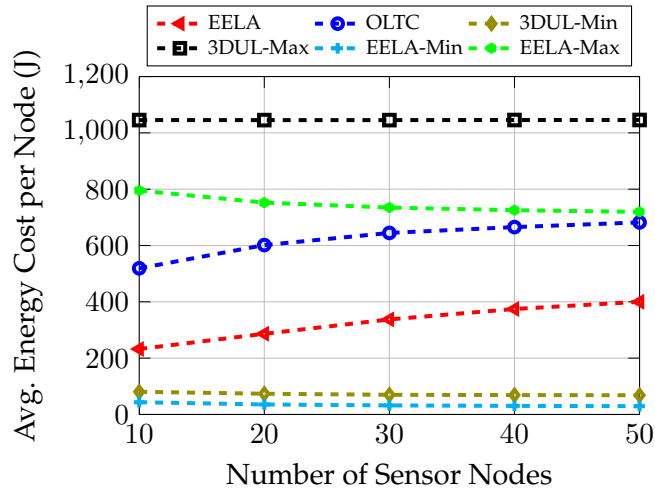


Figure 4.6: Average energy consumption per node.

the proposed EELA enables significant energy savings in a global manner.

- 5) Average Localization Delay and Error: Table 4.3 represents the Average Localization Delay (ALD) and Average Localization Error (ALE) of different schemes, where 3DMin and 3DMax are 3DUL-Min and 3DUL-Max, respectively. EMin and EMax are EELA-Min and EELA-Max, respectively.

I notice that the ALD of EELA is almost the same as that of OLTC. The ALD of EELA-Min is lower than that of EELA by nearly 11 % on average. This is because in EELA-Min, the communication distance travelled by the acoustic signal is shorter than that in other schemes, since it uses the minimum transmission power. As for EELA-Max, the longer communication distance leads to higher delay since it uses maximum transmission power. In 3DUL-Min and 3DUL-Max, the ALD is larger due to the iterative localization design.

Next, for evaluating ALE, each sensor node requires three beacon locations and three distances from anchor nodes since the trilateration technique is considered. It is assumed that anchor nodes broadcast their precise coordinates, so the localization error is generated by the mobility of nodes and depends on the distance between anchor and sensor nodes. From Table 4.3, we can see that EELA performs slightly better than OLTC, while the lowest and highest ALE are achieved by EELA-Min and 3DUL-Max, respectively. Overall, the ALEs are at comparable and reasonable levels for all algorithms.

Metric	EELA	OLTC	3DMin	3DMax	EMin	EMax
ALD (s)	6.87	6.85	6.95	8.96	6.11	7.15
ALE (m)	3.18	3.24	3.29	3.97	3.17	3.30

Table 4.3: Average localization delay and error

Metric	2 m s^{-1}	3 m s^{-1}	4 m s^{-1}
ALC (%)	96.24	96.16	96.26
AEN (J)	249.15	249.44	247.46
ALD (s)	6.90	6.89	6.89

Table 4.4: Average performance of EELA with different speed.

4.5.2.5 Environmental Influences in Simulation

In this section, I change the speed of the current ($v_m = 2 \text{ m s}^{-1}, 3 \text{ m s}^{-1}, 4 \text{ m s}^{-1}$) to observe the effects of environment changes. The number of anchor nodes is the proposed which are deployed on the surface of water while the number of sensor nodes varies from 10 to 50 which are deployed underwater.

Table 4.4 gives the average performance of EELA with different speed, including the Average Localization Coverage (ALC), Average Energy per Node (AEN) and Average Localization Delay (ALD). It is obvious that the average performance of EELA, such as ALC, AEN and ALD have very similar results with the changing of the current speed ($v_m = 2 \text{ m s}^{-1}, 3 \text{ m s}^{-1}, 4 \text{ m s}^{-1}$). This proves that the performance of the proposed EELA is stable with the variation in speed.

Finally, Figure 4.7 shows that, the higher the current velocity (from 2 m s^{-1} to 4 m s^{-1}), the higher the average localization error (from 3.2 m s^{-1} to 8.4 m s^{-1}). This is due to the delay between the time where anchor nodes' broadcast their messages used for location estimation, and the actual. For example, if this transmission delay is between 0 s to 3 s, the node may move 0 s to 6 s. When the current speed is 2 m s^{-1} , it is reasonable that the average transmission error is around 3.2 m. Similarly, when the speed of current is 4 m s^{-1} , the average localization error is around 8.4 m. However, the average localization errors remain constant with the number of sensors, which also shows the stability of the proposed EELA scheme in terms of ALE.

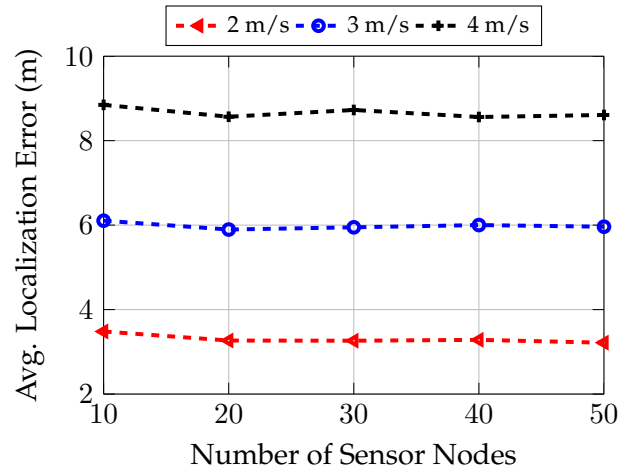


Figure 4.7: Average localization error of EELA with different speed.

4.6 Underwater Localization by Adaptive Fuzzy Stackelberg Game Theory

4.6.1 Problem Formulation

4.6.1.1 Proposed Single Leader Multi-Follower Fuzzy Stackelberg Game Formulation

In Stackelberg game [88], two types of players (leader and followers) are used to model the hierarchy of actions. The leader moves first and selects a strategy. Based on the act of the leader, the followers choose best response strategies that maximize their utilities. Then, the leader selects one strategy to maximize its payoff based on the strategies of followers. In a distributed localization scenario, a sensor node can localize itself after receiving enough location beacons from multiple anchor nodes. However, due to the random and sparse nodes deployment, and the mobility of nodes, the sensor node may not find enough neighbor anchor nodes. To model the proposed problem, a Single-Leader-Multi-Follower Stackelberg game [88] is employed, where the sensor node acts as the single leader while anchor nodes are multiple followers. The sensor node moves first and chooses a transmission power to send a request message, which is similar to the leader releasing a price in Stackelberg game. Anchor nodes move, i.e. selecting a transmission power to send reply message, after the movement of the sensor node.

In EELA scheme, the optimal weights in the payoff functions of both sensor and anchor nodes are required to be selected first in the offline phase, which can only handle one specific scenario. Besides, different appropriate weights need to be assigned to payoff functions in different scenarios. If I select some improper weights, the performances of EELA are poor. Therefore, the adaptive EELA is proposed in this chapter. It selects the Adaptive Neuro-Fuzzy Inference System (ANFIS) [89] to model its payoff functions. The adaptive neuro-fuzzy network is used to model the interaction among the leader and multiple followers in UWSNs. The proposed adaptive EELA adapts in an online learning approach to the encountered changes in the network topology, the node density and the environment.

ANFIS [89] is a fuzzy inference system implemented in the multilayer feedforward neural network with the supervised learning capability. It has some features, such as self-construction, parameter estimation and rule extraction. In the proposed Adaptive EELA scheme, the Stackelberg game uses the ANFIS to make a decision and select the optimal transmission power to interact with other nodes. Due to the adaptability and learning characteristics of neuro-fuzzy algorithm, it performs much better than the conventional fuzzy logic algorithm [90].

The original ANFIS structure is not suitable for the proposed design because the proposed input data is quite large which can produce many possible combination conditions. To solve this problem, the subtractive fuzzy clustering is used to generate the rule base relationship between the input and output variables. They have the following type: if (in1 is in1cluster 'k') and (in2 is in1cluster 'k') and (in3 is in3cluster 'k') then (out1 is out1cluster 'k'). The subtractive fuzzy clustering can automatically determine the number of clusters. The structure of sensor nodes is given in Figure 4.8, where $\tilde{n}_{neig}(P_i)$, \tilde{E}_i^{tl} and $\sum_{h=1}^{n_{srax}} \tilde{Q}_h$ are fuzzy variables which are input parameters of subtractive fuzzy clustering of sensor nodes (See Table 2.1). The output is the value of the payoff. The corresponding structure of anchor nodes is similar with that of sensor nodes, where fuzzy variables are $\tilde{n}_{hd}(Q_j)$, \tilde{E}_j^{tl} and $\sum_{k=1}^{n_{arax}} \tilde{P}_k$ (See Table 2.1) which are input parameters of subtractive fuzzy clustering of anchor nodes. The output is the value of the payoff.

In ANFIS structure models of both sensor nodes and anchor nodes, firstly, each input variable, which varies within a range, might be clustered into several class values to build up the fuzzy rules. Then, the output results are aggregated by some linear membership functions. For example, in Figure 4.8, three input fuzzy parameters, $\tilde{n}_{neig}(P_i)$ which is the number of neighbor anchor nodes of one sensor node, \tilde{E}_i^{tl} which is the total current

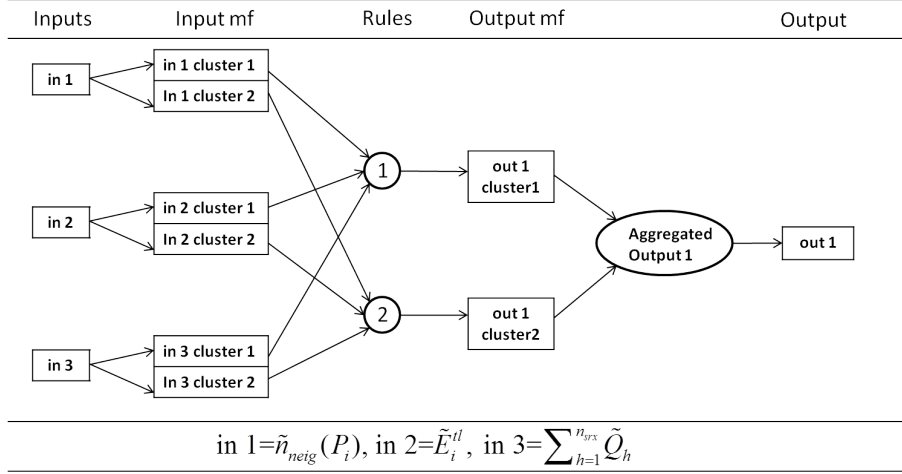


Figure 4.8: ANFIS structure of sensor nodes.

energy of one node and $\sum_{h=1}^{n_{srx}} \tilde{Q}_h$ which is the sum of transmission power received from anchor nodes, are clustered into two clusters in the ‘input mf’ layer. Then, two fuzzy rules are constructed in ‘Rules’ layer, like if (in1 is in1cluster ‘1’) and (in2 is in2cluster ‘1’) and (in3 is in3cluster ‘1’) then (out1 is out1cluster ‘1’). The output payoff value of sensor node is calculated by linear membership functions.

The input Gaussian-type membership functions are given in Eq. (4.28), where c_i and a_i are membership function parameters,

$$u_i(x) = \exp \left\{ -\frac{x - c_i}{a_i} \right\}. \quad (4.28)$$

4.6.1.2 Payoff Function for Anchor Nodes

Anchor nodes are followers. They decide their strategies to handle the maximum number of requests from sensor nodes with minimum energy consumption. \tilde{n}_i^{req} (fuzzy variable) is the additional number of anchor nodes required by sensor node and can be defined by,

$$\tilde{n}_i^{req} = \begin{cases} n_{min}^{req} - |V_i|, & \text{if } |V_i| < n_{min}^{req} \\ 0, & \text{otherwise} \end{cases} \quad (4.29)$$

where n_{min}^{req} represents the number of anchor nodes required for one sensor node to get its location; V_i is the set of the anchor nodes in the communication range of the sensor node i .

The localization ability $\tilde{O}\tilde{A}_j(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ of anchor node j consists of the 'ability of j th anchor node to resolve requests' from sensor nodes, the 'ability of j th anchor node to serve demand' and the 'ratio of the transmission power between sensor nodes $\sum_{i=1}^{n_{arx}}$ and anchor node j '. It can be calculated by,

$$\tilde{O}\tilde{A}_j(Q_j, P_1, P_2, \dots, P_{n_{arx}}) = \frac{\tilde{n}_{hd}(Q_j)}{n_{arx}} + \frac{\tilde{n}_{hd}(Q_j)}{\sum_{k=1}^{n_{arx}} n_k^{req}} - \frac{\sum_{k=1}^{n_{arx}} \tilde{P}_k}{Q_j}. \quad (4.30)$$

In (4.30), $\frac{\tilde{n}_{hd}(Q_j)}{n_{arx}}$ is the 'ability of j th anchor node to resolve requests', where $\tilde{n}_{hd}(Q_j)$ is the number of requests that can be handled by anchor node j with the transmission power Q_j and n_{arx} is the total number of request messages received from sensor nodes. From Proposition 5, we can see that $\tilde{n}_{hd}(Q_j)$ of a follower (anchor node) j is non-decreasing with the increase of the transmission power Q_j . The second term $\frac{\tilde{n}_{hd}(Q_j)}{\sum_{k=1}^{n_{arx}} n_k^{req}}$ is the 'ability of j th anchor node to serve demand'. It means that only $\tilde{n}_{hd}(Q_j)$ requests can be served among the total demand $\sum_{k=1}^{n_{arx}} n_k^{req}$ from sensor nodes. Finally, the third term $\frac{\sum_{k=1}^{n_{arx}} P_i}{Q_j}$ expresses the relation between the sum-transmit power of sensor nodes n_{arx} and the localization ability of the anchor node j . If the transmission power $\sum_{i=1}^{n_{arx}} P_i$ of sensor nodes increase, anchor node j has to handle more sensor nodes. Therefore, the localization ability $\tilde{O}\tilde{A}_j(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ decreases.

Proposition 5. For each anchor node j , the number of neighboring sensor nodes is higher or at least equal with the increase of transmission power Q_j .

Proof. Let us assume that there are n number of nodes uniformly deployed in the simulation area with the size d^3 . $\tilde{n}_{hd}(Q_j)$ can be calculated by,

$$\tilde{n}_{hd}(Q_j) = \rho v_j = \frac{4\pi n \tilde{R}_j^3}{3d^3}. \quad (4.31)$$

According to Eqs. (4.1) and (4.2), the transmission power Q_j of anchor node j is given as,

$$Q_j(\tilde{R}_j) = A_{norm} \tilde{R}_j^k a(f)^{\tilde{R}_j} + Q_{j0}. \quad (4.32)$$

The inverse function $f^{-1}(Q_j) = \tilde{R}_j$ of anchor node j exists and is strict monotone increase with the transmission power Q_j increase, because $\frac{\partial Q_j(\tilde{R}_j)}{\partial \tilde{R}_j} = A_{norm} \left(k \tilde{R}_j^{k-1} a(f)^{\tilde{R}_j} + \tilde{R}_j^k a(f)^{\tilde{R}_j} \ln a(f) \right) > 0$. Therefore, $\tilde{n}_{hd}(Q_j)$ can be represented

by Eq. (4.33). Then, the first order partial derivative of $\tilde{n}_{hd}(Q_j)$ is given in Eq. (4.34),

$$\tilde{n}_{hd}(Q_j) = \frac{4\pi n \left(f^{-1}(\tilde{Q}_j)\right)^3}{3d^3}, \quad (4.33)$$

$$\frac{\partial \tilde{n}_{hd}(Q_j)}{\partial Q_j} = \frac{4\pi n}{d^3} \left(f^{-1}(\tilde{Q}_j)\right)^2 \frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j}. \quad (4.34)$$

Hence, $\frac{\partial \tilde{n}_{hd}(Q_j)}{\partial Q_j} > 0$, which proves the Proposition 5. \square

Thus, the payoff function of any anchor node j considers factors, such as energy cost, the ability to localize sensor nodes and the transmission powers of sensor nodes and anchor nodes. It is defined as the weighted sum of the ratio of current remaining energy to the initial total energy of the j th anchor node $\left(\tilde{E}_j^{tl} - CQ_j\right)$ and the ability to localize sensor nodes $\tilde{O}A_j(Q_j, P_1, P_2, \dots, P_{n_{arx}})$,

$$U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}}) = \left(\tilde{E}_j^{tl} - CQ_j\right) + \tilde{O}A_j(Q_j, P_1, P_2, \dots, P_{n_{arx}}). \quad (4.35)$$

In the first term of Eq. (4.35), \tilde{E}_j^{tl} is the total energy of the j th anchor node and C is the transmission energy cost per unit power. The second term $\tilde{O}A_j(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ is the ability to localize sensor node. The value of payoff function increases, $\tilde{O}A_j(Q_j, P_1, P_2, \dots, P_{n_{arx}})$ increases.

4.6.1.3 Payoff Function for Sensor Nodes

In localization of UWSNs, sensor nodes are leaders. They watch for the decision of anchors which act as followers, and based on the response of the followers, they maximize their profits. The strategy of the leader is to minimize the energy consumption and localize maximum sensor nodes during the localization delay. A sensor node broadcasts 'Request' message to explore the maximum number of anchors. After sensor nodes receive enough neighbor beacon locations, it will localize itself.

The 'ability of sensor node i to find anchor nodes' $OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}})$ is composed of the ratio of the number of anchor nodes $\tilde{n}_{neig}(P_i)$ within 'one-hop' and 'two-hop' can be found with the transmission power P_i and the total number of anchor nodes n_{srx} received as well as the ratio of the sum of the anchor nodes' transmission power Q_j received by

sensor node i and the transmission power P_i of sensor node i , which can be defined as,

$$OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}}) = \frac{\tilde{n}_{neig}(P_i)}{n_{srx}} - \frac{\sum_{h=1}^{n_{srx}} \tilde{Q}_h}{P_i}. \quad (4.36)$$

In the first term $\frac{\tilde{n}_{neig}(P_i)}{n_{srx}}$ of Eq. (4.36), $\tilde{n}_{neig}(P_i)$ of a sensor node i is non-decreasing with the increase of the transmission power P_i , the prove of which is similar to Proposition 5. The second term $\frac{\sum_{h=1}^{n_{srx}} \tilde{Q}_h}{P_i}$ expresses the relation between the sum-transmit power of anchor nodes n_{srx} and the ‘ability of sensor node i to find anchor nodes’. If the transmission power of anchor nodes $\sum_{h=1}^{n_{srx}} \tilde{Q}_h$ increase, sensor node i can receive enough anchor nodes to get its location by a small transmission power. Therefore, the ‘ability of sensor node i to find anchor nodes’ $OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}})$ decreases.

The payoff of any sensor node i increases with the decrease in energy consumption. Also, it increases with the increase of neighbor anchor nodes number. In addition, the payoff of the leader (sensor node) i decreases with each retry it does to send the ‘Request’ message. As with each such retry, the leader has to transmit again. Therefore, the payoff function of sensor node i is calculated as Eq. (4.37), which is the weighted sum of the ratio of current remaining energy to the initial total energy of the i th sensor node $(\tilde{E}_i^{tl} - EP_i)$ and the ‘ability of sensor node i to find anchor nodes’ $OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}})$,

$$U_l(P_i, Q_1, Q_2, \dots, Q_{n_{srx}}) = (\tilde{E}_i^{tl} - EP_i) + OS_i(P_i, Q_1, Q_2, \dots, Q_{n_{srx}}). \quad (4.37)$$

In the first term of Eq. (4.37), \tilde{E}_i^{tl} is the total energy in the i th sensor node and E is the transmission energy cost per unit power of the i th sensor node.

4.6.1.4 Existence of Stackelberg Nash Equilibrium

- 1) Preliminaries: Some concepts and properties of fuzzy variable are introduced in this section, which will be used to prove that the proposed Adaptive EELA scheme reaches Nash equilibrium.

A triplet $(\Theta, p(\Theta), pos)$ is used to express a possibility space, where Θ , $p(\Theta)$ and pos denote a nonempty set, the power of Θ and a possibility measure, respectively. Based on that, the following axioms are given from [91] and [92],

Axiom 1. $Pos(\Theta) = 1, Pos(\phi) = 1.$

Axiom 2. $Pos(\cup_i A_i) = \sup_i Pos\{A_i\}$ for any collection $\{A_i\}$ in $p(\Theta)$, where $i = 1, 2, \dots, m$.

Axiom 3. Let Θ be nonempty sets on which $Pos_i\{\cdot\}$ ($i = 1, 2, \dots, n$) satisfy the first two axioms, respectively, and $\Theta = \prod_{i=1}^n \Theta_i$. Then

$$Pos(A) = \sup_{(\theta_1, \theta_2, \dots, \theta_n) \in A} Pos_1\{\theta_1\} \wedge Pos_2\{\theta_2\} \wedge \dots \wedge Pos_n\{\theta_n\}, \quad (4.38)$$

for each $A \in p(\Theta)$. In that case I can express $Pos = \wedge_{i=1}^n Pos_i$.

Lemma 1. If ξ_i ($i = 1, 2, \dots, n$) are a collection of independent fuzzy variables, and $f_i: R \rightarrow R$ ($i = 1, 2, \dots, n$) are a collection of functions, then, $f_i(\xi_i)$ ($i = 1, 2, \dots, n$) are also the independent fuzzy variables [92].

Definition 1. Let ξ be a fuzzy variable on the possibility space $(\Theta, p(\Theta), pos)$ and $\alpha \in (0, 1]$. Then,

$$\xi_\alpha^L = \inf\{r | Pos([\xi \leq r]) \geq \alpha\}, \xi_\alpha^U = \sup\{r | Pos([\xi \leq r]) \geq \alpha\},$$

are called the α -pessimistic value and the α -optimistic value of ξ , respectively.

Lemma 2. Let ξ_i ($i = 1, 2, \dots, n$) be a collection of independent fuzzy variables defined on the possibility space $(\Theta_i, p(\Theta)_i, pos_i)$ and $f: X \subset R^n \rightarrow R$ be a measurable function. If $f(x_1, x_2, \dots, x_n)$ is monotonic with respect to x_i , respectively, then [93]

(a) $f_\alpha^U(\xi) = f(\xi_{1\alpha}^V, \xi_{2\alpha}^V, \dots, \xi_{n\alpha}^V)$, where $\xi_{i\alpha}^V = \xi_{i\alpha}^U$, if $f(x_1, x_2, \dots, x_n)$ is nondecreasing with respect to x_i , then $\xi_{i\alpha}^V = \xi_{i\alpha}^L$. Otherwise,

(b) $f_\alpha^L(\xi) = f(\xi_{1\alpha}^{\bar{V}}, \xi_{2\alpha}^{\bar{V}}, \dots, \xi_{n\alpha}^{\bar{V}})$, where $\xi_{i\alpha}^{\bar{V}} = \xi_{i\alpha}^U$, if $f(x_1, x_2, \dots, x_n)$ is nondecreasing with respect to x_i , where $f_\alpha^U(\xi)$ and $f_\alpha^L(\xi)$ denote the α -optimistic value and the α -pessimistic value of the fuzzy variable $f(\xi)$, respectively.

Definition 2. Let ξ be a fuzzy variable with a finite expected value. Then [94],

$$E[\xi] = \frac{1}{2} \int_0^1 (\xi_\alpha^L + \xi_\alpha^U) d\alpha. \quad (4.39)$$

- 2) Best Response Strategy of Anchor Nodes: Corresponding to the strategy of the j th anchor node \tilde{Q}_j , the transmission power allocation problem can be cast as the optimization problem in Eq. (4.41). All anchor nodes are non-cooperative. In Proposition 6, the existence of the best response strategy of each anchor node is proved and the unique equilibrium point is computed.

Proposition 6. Let Q_j be the strategy of the j th anchor node. The best response Q_j^* of each

anchor node is given as,

$$Q_j^*(P_1, P_2, \dots, P_{n_{arx}}) = \left(\frac{d^3 \sum_{k=1}^{n_{arx}} E[\tilde{P}_k]}{d^3 C - 4\pi n E \left[(f^{-1}(\tilde{Q}_j))^2 \frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j} \right] N} \right)^{\frac{1}{2}}, \quad (4.40)$$

where $N = \left(\frac{1}{n_{arx}} + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right)$.

Proof. The fuzzy variables \tilde{E}_j^{tl} , $\tilde{n}_{hd}(Q_j)$ and \tilde{P}_k in the optimization problem are all independent and non-negative. The optimization problem is a standard form of convex optimization problem, which is given as,

$$\begin{aligned} \max_{Q_j} E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})] &= \frac{1}{2} \int_0^1 \left\{ \tilde{E}_{ja}^{tlL} - CQ_j \right. \\ &+ \left(\frac{\tilde{n}_{hd\alpha}^L(Q_j)}{n_{arx}} + \frac{\tilde{n}_{hd\alpha}^L(Q_j)}{\sum_{k=1}^{n_{arx}} n_k^{req}} - \frac{\sum_{k=1}^{n_{arx}} \tilde{P}_{k\alpha}^U}{Q_j} \right) + \tilde{E}_{ja}^{tlU} \\ &\left. - CQ_j + \left(\frac{\tilde{n}_{hd\alpha}^U(Q_j)}{n_{arx}} + \frac{\tilde{n}_{hd\alpha}^U(Q_j)}{\sum_{k=1}^{n_{arx}} n_k^{req}} - \frac{\sum_{k=1}^{n_{arx}} \tilde{P}_{k\alpha}^L}{Q_j} \right) \right\} d\alpha. \end{aligned} \quad (4.41)$$

According to Lemma 2 and Definition 2, I represent the optimization problem as,

$$\begin{aligned} \max_{Q_j} E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})] &= E[\tilde{E}_j^{tl}] - CQ_j \\ &+ E[\tilde{n}_{hd}(Q_j)] \left(\frac{1}{n_{arx}} + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right) - \frac{\sum_{k=1}^{n_{arx}} E[\tilde{P}_k]}{Q_j} \end{aligned}$$

subject to

$$\begin{aligned} \tilde{E}_{j\alpha}^{tlL}, \tilde{E}_{j\alpha}^{tlU}, \tilde{n}_{hd\alpha}^L(Q_j), \tilde{n}_{hd\alpha}^U(Q_j) &> 0 \\ \tilde{P}_{k\alpha}^L, \tilde{P}_{k\alpha}^U > 0, n_{arx}, n_k^{req}, Q_j, C &> 0. \end{aligned}$$

The first order partial derivative of $E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})]$ with respect to the

j th anchor node, for $j \in [1, N]$, is

$$\begin{aligned} \frac{\partial E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})]}{\partial Q_j} &= -C + \frac{4\pi n}{d^3} E \left[(f^{-1}(\tilde{Q}_j))^2 \frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j} \right] \\ &\left(\frac{1}{n_{arx}} + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right) + \frac{\sum_{k=1}^{n_{arx}} E[\tilde{P}_k]}{Q_j^2}. \end{aligned}$$

The second order partial derivative of $E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})]$ is given as

$$\begin{aligned} \frac{\partial^2 E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})]}{\partial Q_j^2} &= \frac{4\pi n}{d^3} \left(\frac{1}{n_{arx}} \right. \\ &\left. + \frac{1}{\sum_{k=1}^{n_{arx}} n_k^{req}} \right) E \left[2f^{-1}(\tilde{Q}_j) \left(\frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j} \right)^2 \right. \\ &\left. + (f^{-1}(\tilde{Q}_j))^2 \frac{\partial^2 f^{-1}(\tilde{Q}_j)}{\partial Q_j^2} \right] - \frac{2 \sum_{k=1}^{n_{arx}} E[\tilde{P}_k]}{Q_j^3}. \end{aligned}$$

Here, I need to prove Eq. (4.42) feasibility in order to prove that $\frac{\partial^2 U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})}{\partial Q_j^2}$ is negative,

$$2f^{-1}(\tilde{Q}_j) \left(\frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j} \right)^2 + (f^{-1}(\tilde{Q}_j))^2 \frac{\partial^2 f^{-1}(\tilde{Q}_j)}{\partial Q_j^2} < 0. \quad (4.42)$$

Firstly, I prove $\frac{\partial^2 f^{-1}(\tilde{Q}_j)}{\partial Q_j^2} < 0$. According to Eqs. (4.1) and (4.2), the transmission power Q_j of anchor node j is given as,

$$Q_j(\tilde{R}_j) = A_{norm} \tilde{R}_j^k a(f)^{\tilde{R}_j} + Q_{j0}.$$

The inverse function $f^{-1}(\tilde{Q}_j) = \tilde{R}_j$ of anchor node j exists and is strict monotone increase with the transmission power Q_j increase, because $\frac{\partial Q_j(\tilde{R}_j)}{\partial \tilde{R}_j} = A_{norm} \left(k \tilde{R}_j^{k-1} a(f)^{\tilde{R}_j} + \tilde{R}_j^k a(f)^{\tilde{R}_j} \ln a(f) \right) > 0$ and $\frac{\partial Q_j(\tilde{R}_j)}{\partial \tilde{R}_j} = \frac{1}{\frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j}}$, then

$\frac{\partial^2 f^{-1}(\tilde{Q}_j)}{\partial Q_j^2}$ can be calculated as,

$$\begin{aligned}\frac{\partial^2 f^{-1}(\tilde{Q}_j)}{\partial Q_j^2} &= -\frac{\frac{\partial^2 Q_j(\tilde{R}_j)}{\partial \tilde{R}_j^2} \frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j}}{\left(\frac{\partial Q_j(\tilde{R}_j)}{\partial \tilde{R}_j}\right)^2} \\ &= -\frac{\partial^2 Q_j(\tilde{R}_j)}{\partial \tilde{R}_j^2} \left(\frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j}\right)^3.\end{aligned}\quad (4.43)$$

Due to Eq. (4.44), I get $\frac{\partial^2 f^{-1}(\tilde{Q}_j)}{\partial Q_j^2} < 0$,

$$\begin{aligned}\frac{\partial^2 Q_j(\tilde{R}_j)}{\partial \tilde{R}_j^2} &= A_{norm} \left(k(k-1) \tilde{R}_j^{k-2} a(f)^{\tilde{R}_j} + k \tilde{R}_j^{k-1} a(f)^{\tilde{R}_j} \right. \\ &\quad \left. \ln a(f) + k \tilde{R}_j^{k-1} a(f)^{\tilde{R}_j} \ln a(f) + \tilde{R}_j^k a(f)^{\tilde{R}_j} \ln^2 a(f) \right) > 0.\end{aligned}\quad (4.44)$$

Secondly, I prove Eq. (4.42) feasibility, as $2f^{-1}(\tilde{Q}_j) < \left(f^{-1}(\tilde{Q}_j)\right)^2$, where $f^{-1}(\tilde{Q}_j) = \tilde{R}_j$ and $\tilde{R}_j \gg 1$. Now, according to Eq. (4.43), I prove $\left(\frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j}\right)^2 < \left|\frac{\partial^2 f^{-1}(\tilde{Q}_j)}{\partial Q_j^2}\right|$ as follows,

$$\begin{aligned}&\left|\frac{\partial^2 f^{-1}(\tilde{Q}_j)}{\partial Q_j^2} \left(\frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j}\right)^3\right| - \left(\frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j}\right)^2 > 0 \\ &\Rightarrow \left|\frac{\partial^2 Q_j(\tilde{R}_j)}{\partial \tilde{R}_j^2} \frac{\partial f^{-1}(\tilde{Q}_j)}{\partial Q_j}\right| - 1 > 0.\end{aligned}$$

Therefore, I prove that the second order partial derivative $E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})]$ is negative.

Since the value of the second order partial derivative of $E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})]$ is negative, then the maximum value of $E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})]$ can be achieved at $Q_j^*(P_1, P_2, \dots, P_{n_{arx}})$ by (4.45),

$$\frac{\partial E[U_f(Q_j, P_1, P_2, \dots, P_{n_{arx}})]}{\partial Q_j} = 0.\quad (4.45)$$

□

3) Best Response Strategy of Sensor Node: The existence and uniqueness of Nash

equilibrium of the sensor node is given in Proposition 7, which implies the existence and uniqueness of the Nash equilibrium of the proposed Adaptive EELA game.

Proposition 7. *Let P_i be the strategy of the i th sensor node. The best response P_i^* of each sensor node is given as,*

$$P_i^*(Q_1, Q_2, \dots, Q_{srx}) = \left(\frac{d^3 n_{srx} \sum_{h=1}^{n_{srx}} E[\tilde{Q}_h]}{d^3 n_{srx} E - 4\pi n E \left[\left(f^{-1}(\tilde{P}_i) \right)^2 \frac{\partial f^{-1}(\tilde{P}_i)}{\partial P_i} \right]} \right)^{\frac{1}{2}}.$$

As for the proof of Proposition 7, the steps are similar to those in Section 4.6.1.4, hence omitted.

4.6.2 Proposed Adaptive EELA Model

4.6.2.1 Develop an Adaptive Fuzzy Game-theoretic System

The proposed Adaptive EELA scheme is to create a simple, non-iterative, efficient model to improve the energy-efficient localization in UWSNs. As discussed in Section 4.5, both the sensor node (leader) and multiple anchor nodes (followers) need to make a decision to select the optimal transmission power to interact with other nodes. However, in real environment, many factors can affect their decisions. For example, as for the sensor node, the number of neighbor anchor nodes with ‘one-hop’ and ‘two-hop’ situations, the energy cost, the weights of different factors and so on.

An overview architecture of Adaptive EELA is given in Figure 4.9. In the offline stage, the training data is created by the EELA scheme under different network topologies, which is used to train Adaptive EELA to build the fuzzy rules and membership functions. The training process is only needed for one time. In the online stage, each anchor node or sensor node transmits the collected neighbor information to its own trained Adaptive EELA scheme, which then returns the optimal transmission power for the anchor or sensor node. Then, sensor node uses the optimal transmission power to send the ‘Request’ message while the anchor node uses the optimal transmission power to reply. After sensor nodes get enough anchor node information, they start the localization process by trilateration or other localization algorithms. It is important to notice that the offline stage is only used once, then Adaptive EELA can adapt to the dynamic environment in the online stage.

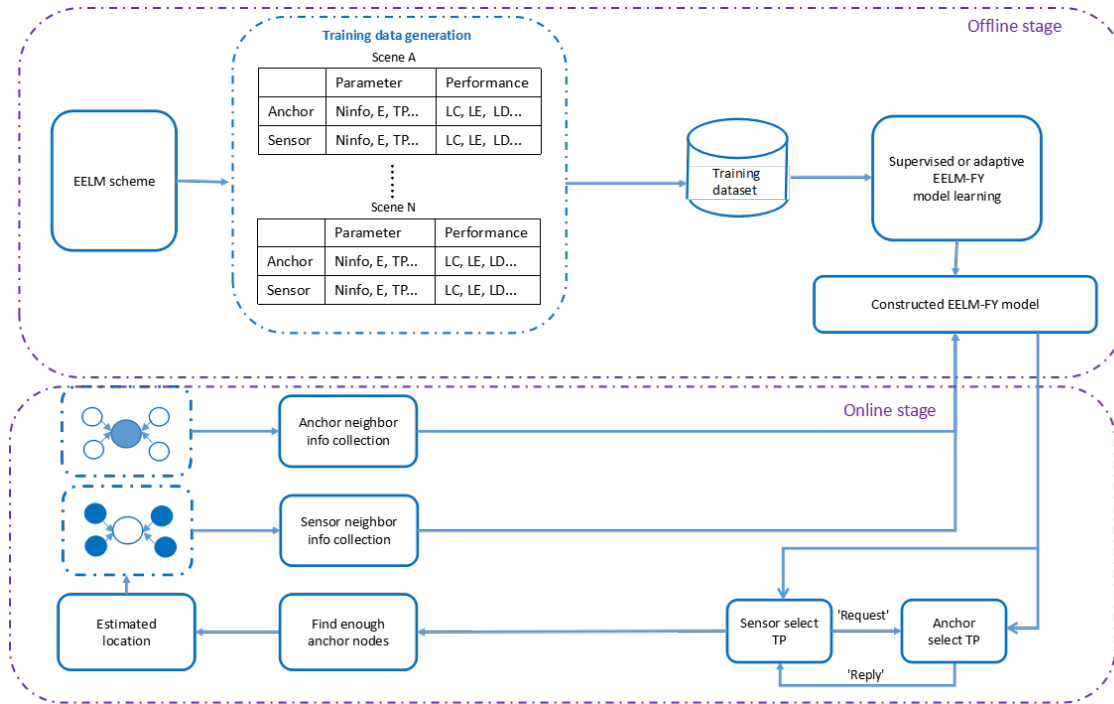


Figure 4.9: Architecture of the proposed Adaptive EELA scheme.

The algorithm design of the proposed Adaptive EELA can refer to that of EELA in Section 4.5.1.5.

4.6.3 Numerical Evaluations

4.6.3.1 Simulation Settings

I use NS-3 (<http://www.nsnam.org/>) simulation platform for evaluating Adaptive EELA. In order to verify the adaptive capacity of the proposed Adaptive EELA algorithm under different network topologies, I set three topological regions, A, B.1 and B.2. They are given in Table 4.5, where SE and ST means the scene and sensor type, respectively. AN and SN are anchor node and sensor node, respectively. All anchor nodes will be randomly deployed on the water surface while all sensor nodes will be randomly deployed underwater. The transmission range is a continuous value in $(0, max_range]$ and the initial transmission range is R_{ini} .

NS-3 UAN models are utilized for the proposed simulation, which is used to model a variety of underwater network scenarios and includes three main parts: the channel,

Table 4.5: Different network topologies used to verify Adaptive EELA.

SE	ST	Nodes Number	Simulation Area	max_rang	P_{ini}	Ocean Speed
A	AN	4	$2500^3 m^3$	$\sqrt{2500^2 \times 3}$	$\frac{max_range}{2}$	{2, 3, 4 m s ⁻¹ }
	SN	{10, 20, 30, 40, 50}				
B.1	AN	20	$10000^2 m^2 \times 2500 m$	5000 m	2500 m	{2, 3, 4 m s ⁻¹ }
	SN	{10, 20, 30, 40, 50}				
B.2	AN	{4, 8, 12, 16, 20}	$10000^2 m^2 \times 2500 m$	5000 m	2500 m	{2, 3, 4 m s ⁻¹ }
	SN	50				

PHY, and MAC models. In addition, both the UanPhyGen model, in which I can change the transmission power according to the given distance and the UanMacCw model [84] are employed as the physical layer and MAC layer, respectively. Notice that even if UanMacCw model provides a collision avoidance mechanism by the contention window, it cannot completely avoid the collision due to the hidden and exposed node problem. I take UanPropModelThorp [95] as the propagation model and AcousticModemEnergyModel [59] as the energy model. $TxPowerW$ in physical layer is changing with different distances. Other simulation parameters are listed in Table 4.2.

In each simulation, any sensor node gets localized after receiving n_{min}^{req} number of replies from anchor nodes. Since the main purpose of this chapter is energy efficiency improvement of localization by topology control and not for designing new localization algorithm, I use trilateration technique, as an example, for node localization to illustrate Adaptive EELA. In trilateration technique, each sensor node requires 3 anchor nodes in order to obtain its location, i.e. $n_{min}^{req} = 3$. I simulate the process for 1000 times, then take the average results to reduce impacts from the randomly deployment situation. All nodes move according to the velocity of ocean current, by following the MCM model [86]. In MCM, the effect of the meandering sub-surface currents and vortices are considered for nodes moving.

4.6.3.2 Simulation Models

In the proposed Adaptive EELA scheme, both anchor and sensor nodes can adjust their transmission power adaptively with fuzzy control algorithm. The performance of Adaptive EELA can be proved by comparing with the proposed schemes listed below.

- 1) Ideal-EELA: Both anchor and sensor nodes can adjust their transmission power with weights information in payoff functions. Weights will be selected in the offline phase

in different evaluation scenes. For example, when 10 sensor nodes and 4 anchor nodes are deployed in 2500 m^3 simulation area, the optimal weights will be selected for this scene in the offline phase.

- 1) Fixed-EELA [8]: Both anchor and sensor nodes can adjust their transmission power with weights information. Fixed weights will be used in different evaluation scenes.
- 2) OLTC [63]: Only anchor nodes can adjust their transmission power while sensor nodes always use the maximum transmission power to send messages.
- 3) EELA-Min [8]: Both anchor and sensor nodes use the fixed minimum transmission power to broadcast message.
- 4) EELA-Max [8]: Both anchor and sensor nodes use the fixed maximum transmission power to broadcast message.

4.6.3.3 Performance Metrics

The following metrics are adopted to evaluate the performance of Adaptive EELA.

- 1) Localization coverage: the ratio of the number of localized sensor nodes to the total number of sensor nodes.
- 2) Average energy consumption per node: the ratio of the total energy consumption of all nodes to the number of all nodes, which is calculated as

$$\varepsilon_{total}^{avg} = \frac{\sum_{i=1}^{|\{N_s\}|} \varepsilon_i + \sum_{j=1}^{|\{N_a\}|} \varepsilon_j}{|\{N_s\}| + |\{N_a\}|}.$$

- 3) Average localization error: calculation in Eq. (4.46),

$$\frac{\sum_{i=1}^{N_{sn_l}} \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2 + (z_i - z'_i)^2}}{|N_{sn_l}|}, \quad (4.46)$$

where for any localized sensor node node i , (x_i, y_i, z_i) and (x'_i, y'_i, z'_i) denote the original and the estimated locations, respectively.

- 4) Average localization delay: The time from a sensor node broadcasting a 'Request' message to the time of obtaining its location.

4.6.3.4 Results and Analysis

I compare the performance of Adaptive EELA with that of Ideal-EELA, Fixed-EELA [8], OLTC [63], EELA-Min [8], and EELA-Max [8] for UWSNs in different scenes shown in Table 4.5. The speed of current is 2 m^2 .

- 1) Localization Coverage: In this section, the average localization coverage (percentage of localized sensor nodes) along with changes of the number of sensor or anchor nodes in the scene A, B.1 and B.2 given in Table 4.5 are analyzed.

Overall, from Figure 4.10, Figure 4.11 and Figure 4.12, I notice that the average localization coverage in EELA-Min stays lowest comparing with other models, because sensor nodes always use the minimum transmission range to send 'Request' messages without considering neighbor information while anchor nodes use the minimum transmission range to reply, which results in fewer sensor nodes with enough beacon information to localize themselves. In contrast, the average localization coverage in EELA-Max achieves the highest comparing with other models, especially when the number of sensor nodes is low. However, the use of higher transmission range leads to higher energy consumption, which will be discussed in Section 4.6.3.4.

The average localization coverage in Ideal-EELA is higher than that in Adaptive EELA, Fixed-EELA and OLTC. Adaptive EELA performs better than Fixed-EELA and OLTC. This is because Ideal-EELA uses the optimal weights in payoff functions to assist both anchor nodes and sensor nodes to select optimal transmission ranges for each scene with different node densities and topologies. Adaptive EELA learned the advantages of Ideal-EELA can achieve a better performance compared to Fixed-EELA and OLTC. In Adaptive EELA, an existing sensor node can dynamically adapt to different scenes and select an optimal transmission range $R_i < R_{max}$ instead of using the maximum transmission range, which can reach as many anchor nodes as possible with the consideration of energy consumption. Similarly, anchor nodes also utilize the optimal transmission range to reply. Both the optimal transmission range for anchor node and sensor node are selected by Stackelberg Nash Equilibrium in section 4.6.1.4. In such a state, both the anchor node and sensor node cannot improve their individual profit by single-sidedly changing their transmission range.

In Figure 4.10, compared to Adaptive EELA, the localization coverage achieved in Fixed-EELA and OLTC is respectively about 6 % and 2 % lower, on average, when

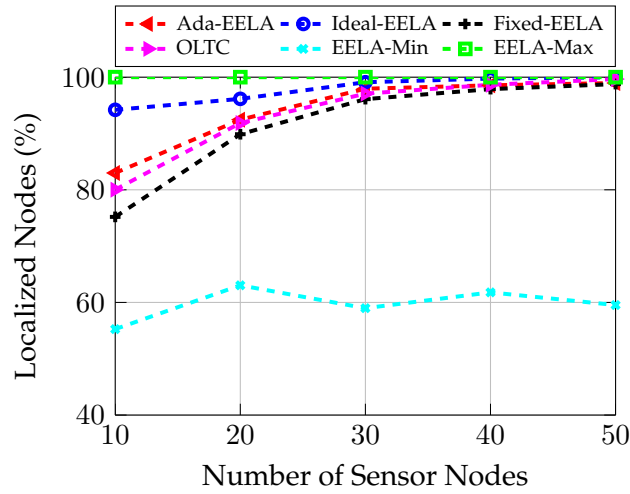


Figure 4.10: Localization coverage of scene A.

the number of sensor nodes is low (e.g. 10 and 20). Figure 4.11 and Figure 4.12 show that in case of scene B.1 and B.2, Adaptive EELA still performs better than Fixed-EELA and OLTC. Fixed-EELA can not select the optimal transmission range for different scenes, which affects its performance. In OLTC, sensor nodes always send request with the maximum transmission range, which leads to a higher rate of packet collision. Thus, anchor nodes will receive fewer 'Request' messages.

Clearly, in Figure 4.10 and Figure 4.11, an increased number of sensor nodes results in increased localization coverage, i.e. the number of localized nodes in the network. This observation is attributed to strategies of the game used in this chapter. The increase in the number of sensor nodes assists in attaining higher transmission range of anchor nodes, which helps more sensor nodes to find their locations. In Figure 4.12, there are up trends in the localization coverage for all models with the number of anchor nodes increase. With the number of anchor nodes increase, more sensor nodes can be localized.

- 2) Average Energy Consumption Per Node: The results of comparison for the average energy consumption per node in different scenes are given in Figure 4.13, Figure 4.14, and Figure 4.15. As a whole, the energy consumption of EELA-Min with the minimum transmission range R_{min} is the lowest among all the six schemes-Adaptive EELA, Ideal-EELA, Fixed-EELA, OLTC and EELA-Max. In contrast, EELA-Max with the maximum transmission range R_{max} consumes the highest energy. Adaptive EELA has almost the same performance with Ideal-EELA. Compared to Adaptive

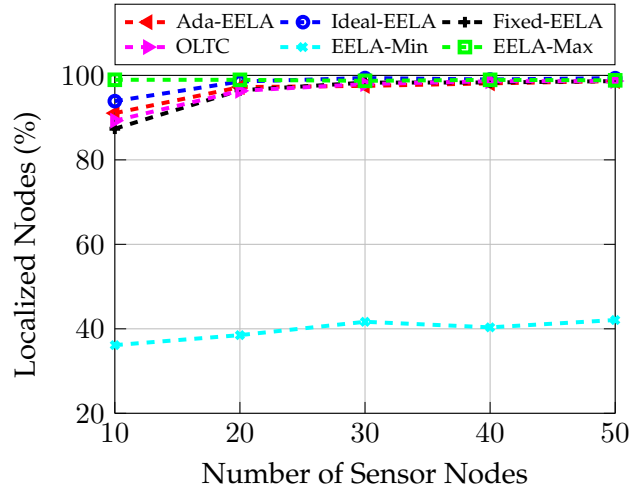


Figure 4.11: Localization coverage of scene B.1.

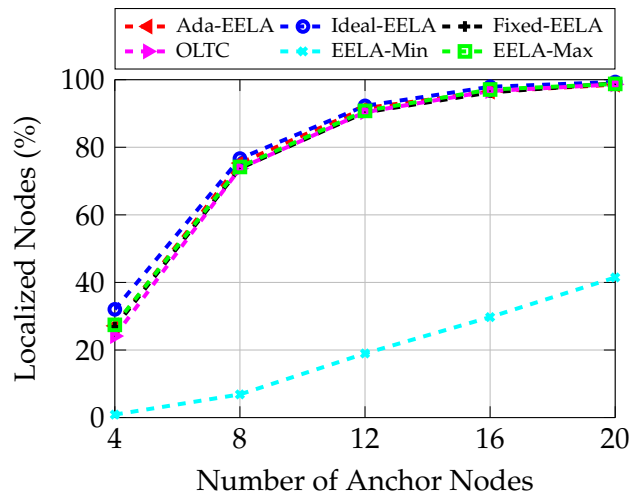


Figure 4.12: Localization coverage of scene B.2.

EELA and Ideal-EELA, Fixed-EELA and OLTC consume more energy. As I discussed in Proposition 4, a higher transmission range results in a higher transmission power, which finally increases the energy consumption of each node.

Figure 4.13 shows the results of simulation scene A described in Section 4.6.3.1. Compared to OLTC (about 623 J) and Fixed-EELA (about 326 J), energy consumption is nearly 66 % and 35 % lower in Adaptive EELA (about 214 J). In OLTC, sensor nodes always utilize the maximum transmission range to send 'Request' messages, which occupies the largest proportion of the energy consumption. In Fixed-EELA, anchor nodes can not perform the trade-off between the energy consumption and the ability to localize sensor nodes well. Similarly, sensor nodes can not perform the trade-off between the energy consumption and the ability to find anchor nodes well. Since for each scene with different node densities and topologies, fixed weights are employed in EELA. Although Ideal-EELA achieves the best performance in energy consumption per node, it requires the optimal weights for different scenes. The optimal weights needs to recalculate in the offline phase for each scene. It is noteworthy to mention that in Adaptive EELA, anchor nodes need to broadcast two times in order to build the 'two hop' anchor neighboring list, so the average energy consumption per anchor node in Adaptive EELA (about 382 J) is 30 % higher than that in OLTC (about 295 J) on average. In addition, the average energy consumption per sensor node in Adaptive EELA (about 176 J) is 75 % lower than that in OLTC (about 692 J) on average, respectively. That means even if anchor nodes broadcast two times in the pre-processing phase, Adaptive EELA still consumes much less energy than that in OLTC considering all nodes. This is because in localization systems, the number of sensor nodes is much larger than that of anchor nodes.

Figure 4.14 shows the results of comparison for the average energy consumption per node with the number of sensor nodes increase. Some characteristics in Figure 4.14, like the trend of different models, are similar in the explanations in Figure 4.13. Both in Figure 4.13 and Figure 4.14, as for Adaptive EELA, Ideal-EELA, Fixed-EELA and OLTC, the higher the density of sensor nodes is, the higher the average energy consumption per node. This is because more sensor nodes result in higher transmission range of anchor nodes, which finally leads to higher energy consumption of anchor nodes. As for EELA-Min and EELA-Max, a high density sensor nodes results in a low energy consumption. Reasons can be found in Eq. (4.46), which calculates the average energy consumption per node. Anchor nodes consumes much more energy than sensor nodes and the number of anchor nodes is fixed. With the number of

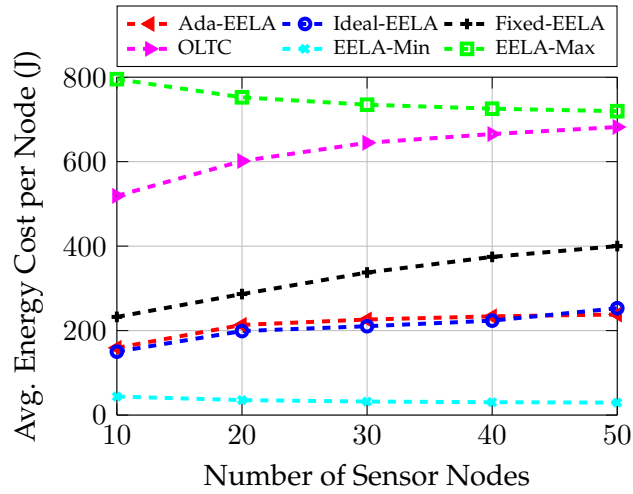


Figure 4.13: Average energy consumption per node of scene A.

sensor nodes increase, the average energy consumption per node becomes low.

Figure 4.15 depicts the results of comparison for the average energy consumption per node with the number of anchor nodes increase. Clearly, compared with Fixed-EELA and OLTC, the average energy consumption per node is approximately 13% and 23% lower in Adaptive EELA. I notice that with increase in the number of anchor nodes, the average energy consumption per node decreases in Adaptive EELA, Ideal-EELA and Fixed-EELA. As shown in Eq. , a higher density of anchor nodes results in a lower transmission range of sensor nodes, which finally reduces the average energy consumption per node. In addition, we can see that an increased density of anchor nodes results in increase of average energy consumption per node in OLTC, EELA-Max and EELA-Min. This is because sensor nodes use the fixed transmission range in these models, for example, sensor nodes in OLTC and EELA-Max use the maximum transmission range to send messages. When the number of anchor nodes increases, the average energy consumption per node increase, due to anchor nodes consumes more energy than sensor nodes.

- 3) Average Localization Delay: Table 4.6 depicts the average localization delay along with changes of the number of sensor or anchor nodes in the scene A, B.1 and B.2 given in Table 4.5, where AEELA, IEELA, FEELA, EELAn and EELAx denotes Ada-EELA, Ideal-EELA, Fixed-EELA, EELA-Min and EELA-Max, respectively.

Overall, the average localization delay of EELA-Min is lowest while EELA-Max

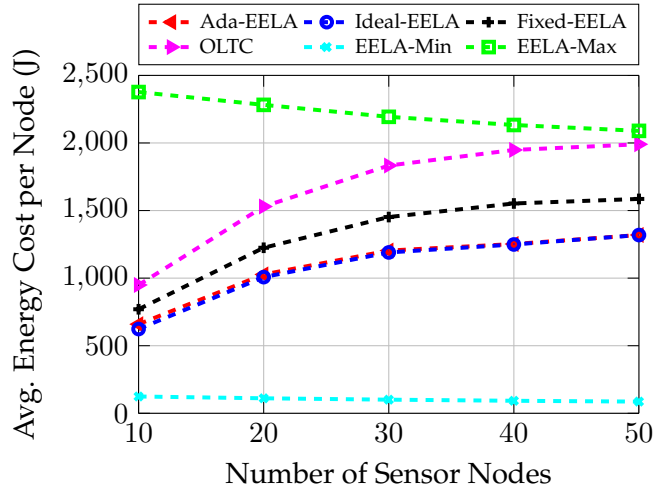


Figure 4.14: Average energy consumption per node of scene B.1.

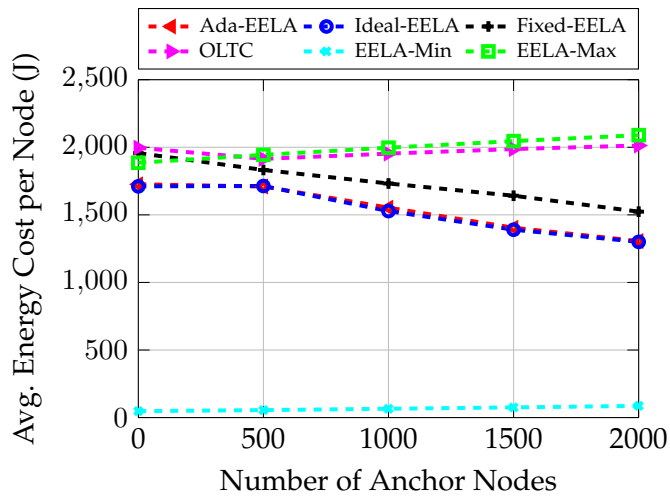


Figure 4.15: Average energy consumption per node of scene B.2.

Scene	AEELA	IEELA	FEELA [8]	OLTC [63]	EELAn [8]	EELAx [8]
A	6.11	5.86	6.88	6.80	5.41	6.95
B.1	28.07	27.14	29.52	29.56	22.75	30.84
B.2	17.08	16.68	18.86	18.78	14.37	18.82

Table 4.6: Average localization delay (s).

reaches the highest value of delay. It is because in EELA-Min, by using the minimum transmission range, the communication distance traveled by the acoustic signal is shorter than that in other schemes. In contrast, EELA-Max, due to using the maximum transmission range, has a longer communication distance than that in other schemes. Compared with Fixed-EELA and OLTC, Ada-EELA achieves a lower localization delay. Ideal-EELA performs slightly better than Ada-EELA.

In scene A, Ada-EELA results in 12% and 11% less localization delay compared to Fixed-EELA and OLTC. This behavior is attributed to the design goal of Ada-EELA-energy-efficient selection of transmission range. The smaller the transmission range is, the lower the average localization delay. Similarly, in scene B.1 and B.2, Ada-EELA performs better than Fixed-EELA and OLTC and the reason are the same with that in scene A.

- 4) Average Localization Error: Table 4.7 shows the comparison results of the average localization error induced during localization for Ada-EELA (AEELA), Ideal-EELA (IEELA), Fixed-EELA (FEELA), OLTC, EELA-Min (EELAn) and EELA-Max (EELAx). Since I use the trilateration technique for localization, for a sensor node, it requires three beacon locations and three distances from anchor nodes. As anchor nodes broadcast their precise coordinates, the localization error is generated by the mobility of nodes and the distance between anchor node and sensor node. In scene A and B.1, the average localization error occurring in Ada-EELA is slightly lower than that in Fixed-EELA and OLTC. The differences are small considering the simulation scene B.2, so I consider all models have very close performance (about 3.38 m on average) in this scene.

Although Adaptive EELA does not have much effect on the average localization error comparing with other schemes, I still discuss and visualize it. Because the average localization error is an important performance metric in UWSNs localization models, and I want to show that the performance of Adaptive EELA in this area are slightly better or at least not worse than other schemes, such as OLTC and Fixed-

Scene	AEELA	IEELA	FEELA [8]	OLTC [63]	EELAn [8]	EELAx [8]
A	2.80	2.51	3.19	3.24	3.29	3.27
B.1	3.75	3.61	4.01	3.98	3.88	3.72
B.2	3.32	3.25	3.47	3.46	3.46	3.37

Table 4.7: Average localization error (m).

EELA. At the same time, Adaptive EELA improves the energy consumption a lot as I discussed in Section 4.6.3.4.

4.7 Summary and Future Work

In this Chapter, firstly, I have considered the problem of energy-efficient sensor node localization using multiple anchor nodes, in underwater sensor networks where battery saving is essential. A Single-Leader-Multi-Follower Stackelberg game is used to model the considered localization problem, where anchor nodes act as followers of each sensor node, which acts as a leader. Considering the trade-off between localization ability and energy consumption, optimal transmission power strategies are devised for anchor and sensor nodes, which are shown to achieve Nash Equilibrium. Based on this analysis, I have proposed the EELA algorithm [8] defining the communication protocol among anchor and sensor nodes, for enabling energy-efficient localization. Simulation results demonstrate that compared to baseline schemes, the proposed EELA enables similar or better performance in terms of localization coverage, errors and delays, while drastically reducing the amount of consumed energy, i.e., down to half the consumption of reference OLTC [63].

In order to solve the environment adaptation problem of EELA, an Adaptive Energy-Efficient Localization Algorithm (Adaptive EELA) based on the Fuzzy game theoretic method is proposed. Adaptive EELA can adapt the dynamic environment changes, such as node densities or node topologies. It requires the offline learning only once with no need for prior knowledge, new equipment, or extra cost. Then, in the online phase, it can select the optimal transmission range both for anchor nodes and sensor nodes in different scenes automatically, which builds the energy-efficient localization scheme in UWSNs. Adaptive EELA works well in both sparse and dense mobile UWSNs. In many scenarios of UWSNs, sensor nodes with limited battery are deployed underwater which results in difficulty to replace the battery or to recharge them. Adaptive EELA is well fit this

kind of scenario and can be implemented in a real world. Numerical evaluation results demonstrate that Adaptive EELA achieves a high localization coverage compared with that in other schemes, such as OLTC [63] and EELA [8], by spending low battery power per node.

In the future works, I would like to consider the multi-path propagation problem caused by the reflections, temperature or salinity of the water into the proposed Adaptive EELA model.

Chapter 5

Conclusion and Future Work

In this thesis, four novel localization algorithms including Tri-MCL, Twi-AdaBoost, EELA and Adaptive EELA, are proposed to improve the performance of the localization. Firstly, Tri-MCL is presented to improve the localization accuracy in WSNs. The proposed method employs three different distance measurement approaches based on range-free methods to estimate distances between sensor nodes and anchor nodes. These distances are then used to filter out particles not lying within rings around the anchor nodes with a radius corresponding the distance estimates. The weights of different particles are also considered, which means that the weight of each particle is related to the distance between anchor node and sensor node. The results from our simulations and experiments validate the effectiveness of our proposed algorithms in improving localization accuracy and reducing computational costs during re-sampling.

Then, Twi-AdaBoost is introduced to explore the accelerometer, gyroscope and magnetometer sensors on both smart-phone and smart-watch. The key contribution of the proposed Twi-AdaBoost algorithm is fusing the co-occurrence information to get a better performance for the indoor localization based on the real world data. The indoor localization datasets [38] used in this chapter have the multi-source characteristics, which are supported by the presence of two different devices collecting data simultaneously from the surrounding environment: a smart-phone and a smart-watch, respectively. Each device collects multivariate data represented by their internal sensors, such as acceleration, orientation, and gyroscope. From the experiment results, it is obvious that Twi-AdaBoost convincingly outperforms the state-of-the-art indoor localization algorithms, taking advantage of the co-occurrence correlation across the sensors from multiple devices. Specifically, the localization error of position x and y achieved by Twi-AdaBoost is 0.387 m and 0.398 m,

respectively.

Lastly, I introduced two algorithms named EELA and adaptive EELA. Both of them discuss the topology control for energy-efficient localization in mobile underwater sensor networks using Stackelberg game. In EELA, I have considered the problem of energy-efficient sensor node localization using multiple anchor nodes, in underwater sensor networks where battery saving is essential. A Single-Leader-Multi-Follower Stackelberg game is used to model the considered localization problem, where anchor nodes act as followers of each sensor node, which acts as a leader. Considering the trade-off between localization ability and energy consumption, optimal transmission power strategies are devised for anchor and sensor nodes, which are shown to achieve Nash Equilibrium. Based on this analysis, I have proposed the EELA algorithm [8] defining the communication protocol among anchor and sensor nodes, for enabling energy-efficient localization. Simulation results demonstrate that compared to baseline schemes, the proposed EELA enables similar or better performance in terms of localization coverage, errors and delays, while drastically reducing the amount of consumed energy, i.e., down to half the consumption of reference OLTC [63]. Based on EELA, in order to solve the environment adaptation problem of EELA, an Adaptive Energy-Efficient Localization Algorithm (Adaptive EELA) based on the Fuzzy game theoretic method is proposed. Adaptive EELA can adapt the dynamic environment changes, such as node densities or node topologies. It requires the offline learning only once with no need for prior knowledge, new equipment, or extra cost. Then, in the online phase, it can select the optimal transmission range both for anchor nodes and sensor nodes in different scenes automatically, which builds the energy-efficient localization scheme in UWSNs. Adaptive EELA works well in both sparse and dense mobile UWSNs. In many scenarios of UWSNs, sensor nodes with limited battery are deployed underwater which results in difficulty to replace the battery or to recharge them. Adaptive EELA is well fit this kind of scenario and can be implemented in a real word. Numerical evaluation results demonstrate that Adaptive EELA achieves a high localization coverage compared with that in other schemes, such as OLTC [63] and EELA [8], by spending low battery power per node.

In the future, I plan to extend Tri-MCL algorithm and implement it in a real world environment. More co-occurrence information from multiple devices like the Camera or WiFi, are considered by machine learning methods, such as Adaboost, to improve the localization accuracy in indoor environment. In addition, I would like to consider the multi-path propagation problem caused by the reflections, temperature or salinity of the water into the proposed Adaptive EELA model.

Bibliography

- [1] Robert Szewczyk, Eric Osterweil, Joseph Polastre, Michael Hamilton, Alan Mainwaring, and Deborah Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, 2004.
- [2] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiu-an Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. *ACM Sigplan Notices*, 37(10):96–107, 2002.
- [3] Tian He, Sudha Krishnamurthy, John A Stankovic, Tarek Abdelzaher, Liqian Luo, Radu Stoleru, Ting Yan, Lin Gu, Jonathan Hui, and Bruce Krogh. Energy-efficient surveillance system using wireless sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 270–283. ACM, 2004.
- [4] Upkar Varshney. Pervasive healthcare and wireless health monitoring. *Mobile Networks and Applications*, 12(2-3):113–127, 2007.
- [5] Abdellah Chehri, Paul Fortier, and Pierre Martin Tardif. Uwb-based sensor networks for localization in mining environments. *Ad Hoc Networks*, 7(5):987–1000, 2009.
- [6] Arne Bochém, Yali Yuan, and Dieter Hogrefe. Tri-mcl: Synergistic localization for mobile ad-hoc and wireless sensor networks. In *IEEE 41st Conference on Local Computer Networks (LCN)*, pages 333–338. IEEE, 2016, ©2016 IEEE.
- [7] Yali Yuan, Christian Melching, Yachao Yuan, and Dieter Hogrefe. Multi-device fusion for enhanced contextual awareness of localization in indoor environments. *IEEE Access*, 6:7422–7431, 2018.
- [8] Yali Yuan, Chencheng Liang, Megumi Kaneko, Xu Chen, and Dieter Hogrefe. Topol-

- ogy control for energy-efficient localization in mobile underwater sensor networks using stackelberg game. *arXiv preprint arXiv:1805.12361*, 2018.
- [9] Wen-Zhan Song, Renjie Huang, Mingsen Xu, Andy Ma, Behrooz Shirazi, and Richard LaHusen. Air-dropped sensor network for real-time high-fidelity volcano monitoring. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 305–318. ACM, 2009.
- [10] Brad Karp and Hsiang-Tsung Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.
- [11] Young-Bae Ko and Nitin H Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wireless networks*, 6(4):307–321, 2000.
- [12] Martin Mauve, Jörg Widmer, and Hannes Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, 2001.
- [13] Hongyang Chen, Gang Wang, Zizhuo Wang, Hing-Cheung So, and H Vincent Poor. Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 11(1):108–116, 2012.
- [14] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps) using aoa. In *IEEE Societies Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'03)*, volume 3, pages 1734–1743. IEEE, 2003.
- [15] Yeong-Sheng Chen, Shih-Hao Chang, and Chun-Chien Teng. Location estimation based on convex overlapping communication regions in wireless ad hoc sensor networks. In *4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE'14)*, pages 1–5. IEEE, 2014.
- [16] Shrawan Kumar and DK Lobiyal. An advanced dv-hop localization algorithm for wireless sensor networks. *Wireless personal communications*, 71(2):1365–1385, 2013.
- [17] Harsha Chenji and Radu Stoleru. Toward accurate mobile sensor network localization in noisy environments. *IEEE Transactions on Mobile Computing*, 12(6):1094–1106, 2013.
- [18] Lingxuan Hu and David Evans. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 45–57. ACM, 2004.

- [19] Aline Baggio and Koen Langendoen. Monte-carlo localization for mobile wireless sensor networks. In *International Conference on Mobile Ad-Hoc and Sensor Networks*, pages 317–328. Springer, 2006.
- [20] Jang-Ping Sheu, Wei-Kai Hu, and Jen-Chiao Lin. Distributed localization scheme for mobile sensor networks. *IEEE Transactions on Mobile Computing*, 9(4):516–526, 2010.
- [21] Shigeng Zhang, Jiannong Cao, Lijun Chen, and Daoxu Chen. Locating nodes in mobile sensor networks more accurately and faster. In *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08)*, pages 37–45. IEEE, 2008.
- [22] Bram Dil, Stefan Dulman, and Paul Havinga. Range-based localization in mobile sensor networks. In *European Workshop on Wireless Sensor Networks*, pages 164–179. Springer, 2006.
- [23] Rui Huang and Gergely V Zaruba. Incorporating data from multiple sensors for localizing nodes in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 6(9):1090–1104, 2007.
- [24] Jiyoung Yi, Sungwon Yang, and Hojung Cha. Multi-hop-based monte carlo localization for mobile sensor networks. In *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'07)*, pages 162–171. IEEE, 2007.
- [25] Masoomeh Rudafshani and Suprakash Datta. Localization in wireless sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 51–60. ACM, 2007.
- [26] Sameer Tilak, Vinay Kolar, Nael B Abu-Ghazaleh, and K-D Kang. Dynamic localization control for mobile sensor networks. In *24th IEEE International Performance, Computing, and Communications Conference, (IPCCC'05)*, pages 587–592. IEEE, 2005.
- [27] Tsung-Han Lin, Polly Huang, Hao-Hua Chu, and Chuang-Wen You. Energy-efficient boundary detection for rf-based localization systems. *IEEE Transactions on Mobile Computing*, 8(1):29–40, 2009.
- [28] Salke Hartung, Arne Bochm, Andreas Zdziarstek, and Dieter Hogrefe. Applied sensor-assisted monte carlo localization for mobile wireless sensor networks. In *EWSN*, pages 181–192, 2016.

- [29] Shigeng Zhang, Jiannong Cao, Chen Li-Jun, and Daoxu Chen. Accurate and energy-efficient range-free localization for mobile sensor networks. *IEEE Transactions on Mobile Computing*, 9(6):897–910, 2010.
- [30] Yongtao Zhang, Lingguo Cui, and Senchun Chai. Energy-efficient localization for mobile sensor networks based on rss and historical information. In *27th Chinese Control and Decision Conference (CCDC'15)*, pages 5246–5251. IEEE, 2015.
- [31] Jiang Xu, Fanyu Bu, Wei Si, Yiteng Qiu, and Zhikui Chen. An algorithm of weighted monte carlo localization based on smallest enclosing circle. In *International conference on Internet of things (iThings/CPSCoM'11) and 4th international conference on cyber, physical and social computing*, pages 157–161. IEEE, 2011.
- [32] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps). In *IEEE Global Telecommunications Conference, (GLOBECOM'01)*, volume 5, pages 2926–2931. IEEE, 2001.
- [33] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, 7(5):28–34, 2000.
- [34] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Random waypoint considered harmful. In *IEEE societies twenty-second annual joint conference of the IEEE computer and communications (INFOCOM'03)*, volume 2, pages 1312–1321. IEEE, 2003.
- [35] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. *Massachusetts Institute of Technology, USA*, 2005.
- [36] AKM Mahtab Hossain and Wee-Seng Soh. A survey of calibration-free indoor positioning systems. *Computer Communications*, 66:1–13, 2015.
- [37] Robert Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys and Tutorials*, 15(3):1281–1293, 2013.
- [38] Paolo Barsocchi, Antonino Crivello, Davide La Rosa, and Filippo Palumbo. A multi-source and multivariate dataset for indoor localization methods based on wlan and geo-magnetic field fingerprinting. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN'16)*, pages 1–8. IEEE, 2016.
- [39] Mohammad Shaifur Rahman, Youngil Park, and Ki-Doo Kim. Rss-based indoor localization algorithm for wireless sensor network using generalized regression neural network. *Arabian journal for science and engineering*, 37(4):1043–1053, 2012.

- [40] Zhi-li Wu, Chun-hung Li, Joseph Kee-Yin Ng, and Karl RPH Leung. Location estimation via support vector regression. *IEEE Transactions on mobile computing*, 6(3):311–321, 2007.
- [41] Hao Chen, Yifan Zhang, Wei Li, Xiaofeng Tao, and Ping Zhang. Confi: Convolutional neural networks based indoor wi-fi localization using channel state information. *IEEE Access*, 5:18066–18074, 2017.
- [42] David Sánchez-Rodríguez, Pablo Hernández-Morera, José Ma Quinteiro, and Itziar Alonso-González. A low complexity system based on multiple weighted decision trees for indoor localization. *Sensors*, 15(6):14809–14829, 2015.
- [43] Frank Vanheel, Jo Verhaevert, Eric Laermans, Ingrid Moerman, and Piet Demeester. Automated linear regression tools improve rssi wsn localization in multipath indoor environment. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):38, 2011.
- [44] Zhenghua Chen, Han Zou, Hao Jiang, Qingchang Zhu, Yeng Chai Soh, and Lihua Xie. Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization. *Sensors*, 15(1):715–732, 2015.
- [45] Rui Ma, Qiang Guo, Changzhen Hu, and Jingfeng Xue. An improved wifi indoor positioning algorithm by weighted fusion. *Sensors*, 15(9):21824–21843, 2015.
- [46] Chenshu Wu, Zheng Yang, Zimu Zhou, Yunhao Liu, and Mingyan Liu. Mitigating large errors in wifi-based indoor localization for smartphones. *IEEE Transactions on Vehicular Technology*, 66(7):6246–6257, 2017.
- [47] Fei Gu, Jianwei Niu, and Lingjie Duan. Waipo: A fusion-based collaborative indoor localization system on smartphones. *IEEE/ACM Transactions on Networking*, 25(4):2267–2280, 2017.
- [48] Sebastian Hilsenbeck, Dmytro Bobkov, Georg Schroth, Robert Huitl, and Eckehard Steinbach. Graph-based data fusion of pedometer and wifi measurements for mobile indoor positioning. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pages 147–158. ACM, 2014.
- [49] Filippo Palumbo, Claudio Gallicchio, Rita Pucci, and Alessio Micheli. Human activity recognition using multisensor data fusion based on reservoir computing. *Journal of Ambient Intelligence and Smart Environments*, 8(2):87–107, 2016.

- [50] Muhammad Shoaib, Stephan Bosch, Hans Scholten, Paul JM Havinga, and Ozlem Durmaz Incel. Towards detection of bad habits by fusing smartphone and smartwatch sensors. In *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom'15 Workshops)*, pages 591–596. IEEE, 2015.
- [51] L Breiman, JH Friedman, RA Olshen, and CJ Stone. Classification and regression trees. monterey, calif., usa: Wadsworth, 1984.
- [52] Harris Drucker. Improving regressors using boosting techniques. In *ICML*, volume 97, pages 107–115, 1997.
- [53] Diansheng Guo, Jin Chen, Alan M MacEachren, and Ke Liao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE transactions on visualization and computer graphics*, 12(6):1461–1474, 2006.
- [54] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- [55] Dimitri P Solomatine and Durga L Shrestha. Adaboost. rt: a boosting algorithm for regression problems. In *IEEE International Joint Conference on Neural Networks*, volume 2, pages 1163–1168. IEEE, 2004.
- [56] Peng Xie, Jun-Hong Cui, and Li Lao. Vbf: vector-based forwarding protocol for underwater sensor networks. In *Networking*, volume 3976, pages 1216–1221. Springer, 2006.
- [57] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3):257–279, 2005.
- [58] Hwee-Pink Tan, Roe Diamant, Winston KG Seah, and Marc Waldmeyer. A survey of techniques and challenges in underwater localization. *Ocean Engineering*, 38(14):1663–1676, 2011.
- [59] Lee Freitag, Matthew Grund, Sandipa Singh, James Partan, Peter Koski, and Keenan Ball. The whoi micro-modem: An acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1086–1092. IEEE, 2005.
- [60] Amin Y Teymorian, Wei Cheng, Liran Ma, Xiuzhen Cheng, Xicheng Lu, and Zexin Lu.

- 3d underwater sensor network localization. *IEEE Transactions on Mobile Computing*, 8(12), 2009.
- [61] Zhong Zhou, Zheng Peng, Jun-Hong Cui, Zhijie Shi, and Amvrossios Bagtzoglou. Scalable localization with mobility prediction for underwater sensor networks. *IEEE Transactions on Mobile Computing*, 10(3):335–348, 2011.
- [62] Hanjiang Luo, Kaishun Wu, Yue-Jiao Gong, and Lionel M Ni. Localization for drifting restricted floating ocean sensor networks. *IEEE Transactions on Vehicular Technology*, 65(12):9968–9981, 2016.
- [63] Sudip Misra, Tamoghna Ojha, and Ayan Mondal. Game-theoretic topology control for opportunistic localization in sparse underwater sensor networks. *IEEE transactions on mobile computing*, 14(5):990–1003, 2015.
- [64] M Talha Isik and Ozgur B Akan. A three dimensional localization algorithm for underwater acoustic sensor networks. *IEEE Transactions on Wireless Communications*, 8(9), 2009.
- [65] Guangjie Han, Jinfang Jiang, Lei Shu, Yongjun Xu, and Feng Wang. Localization algorithms of underwater wireless sensor networks: A survey. *Sensors*, 12(2):2026–2061, 2012.
- [66] Gurkan Tuna and V Cagri Gungor. A survey on deployment techniques, localization algorithms, and research challenges for underwater acoustic sensor networks. *International Journal of Communication Systems*, 30(17), 2017.
- [67] Affan A Syed, John S Heidemann, et al. Time synchronization for high latency acoustic networks. In *INFOCOM*, volume 6, pages 1–12, 2006.
- [68] Jun Liu, Zhaohui Wang, Jun-Hong Cui, Shengli Zhou, and Bo Yang. A joint time synchronization and localization design for mobile underwater sensor networks. *IEEE Transactions on Mobile Computing*, 15(3):530–543, 2016.
- [69] Melike Erol, Luiz Filipe M Vieira, and Mario Gerla. Auv-aided localization for underwater sensor networks. In *International Conference on Wireless Algorithms, Systems and Applications (WASA'07)*, pages 44–54. IEEE, 2007.
- [70] Marc Waldmeyer, Hwee-Pink Tan, and Winston KG Seah. Multi-stage auv-aided localization for underwater wireless sensor networks. In *IEEE Workshops of International*

- Conference on Advanced Information Networking and Applications (WAINA'11)*, pages 908–913. IEEE, 2011.
- [71] Tamoghna Ojha and Sudip Misra. Hasl: High-speed auv-based silent localization for underwater sensor networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pages 128–140. Springer, 2013.
- [72] Ning Li and Jennifer C Hou. Localized topology control algorithms for heterogeneous wireless networks. *IEEE/ACM Transactions on Networking*, 13(6):1313–1324, 2005.
- [73] Harish Sethu and Thomas Gerety. A new distributed topology control algorithm for wireless environments with non-uniform path loss and multipath propagation. *Ad Hoc Networks*, 8(3):280–294, 2010.
- [74] Hongliang Ren and Max Q-H Meng. Game-theoretic modeling of joint topology control and power scheduling for wireless heterogeneous sensor networks. *IEEE Transactions on Automation Science and Engineering*, 6(4):610–625, 2009.
- [75] Jiang Zhu, Dingde Jiang, Shaowei Ba, and Yuping Zhang. A game-theoretic power control mechanism based on hidden markov model in cognitive wireless sensor network with imperfect information. *Neurocomputing*, 220:76–83, 2017.
- [76] Linfeng Liu, Ruchuan Wang, and Fu Xiao. Topology control algorithm for underwater wireless sensor networks using gps-free mobile sensor nodes. *Journal of Network and Computer Applications*, 35(6):1953–1963, 2012.
- [77] Linfeng Liu, Ye Liu, and Ningshen Zhang. A complex network approach to topology control problem in underwater acoustic sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(12):3046–3055, 2014.
- [78] Albert F Harris III and Michele Zorzi. Modeling the underwater acoustic channel in ns2. In *Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, page 18. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.
- [79] Melike Erol-Kantarci, Hussein T Mouftah, and Sema Oktug. A survey of architectures and localization techniques for underwater acoustic sensor networks. *IEEE Communications Surveys & Tutorials*, 13(3):487–502, 2011.
- [80] Milica Stojanovic. On the relationship between capacity and distance in an under-

- water acoustic communication channel. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):34–43, 2007.
- [81] Milica Stojanovic. Underwater acoustic communications. In *Professional Program Proceedings (Electro/95 International)*, pages 435–440. IEEE, 1995.
- [82] Hanif D Sherali, Allen L Soyster, and Frederic H Murphy. Stackelberg-nash-cournot equilibria: characterizations and computations. *Operations Research*, 31(2):253–276, 1983.
- [83] Dimitris E Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic systems*, 32(4):1239–1248, 1996.
- [84] Nathan Parrish, Leonard Tracy, Sumit Roy, Payman Arabshahi, and Warren LJ Fox. System design considerations for undersea networks: Link and multiple access protocols. *IEEE Journal on Selected Areas in Communications*, 26(9), 2008.
- [85] Leonid Brekhovskikh and Yury Lysanov. Scattering of sound at rough surfaces. In *Fundamentals of Ocean Acoustics*, pages 173–207. Springer, 1982.
- [86] Antonio Caruso, Francesco Paparella, Luiz Filipe M Vieira, Melike Erol, and Mario Gerla. The meandering current mobility model and its impact on underwater mobile sensor networks. In *27th IEEE Conference on Computer Communications (INFOCOM'08)*, pages 221–225. IEEE, 2008.
- [87] EvoLogics. S2cr 12/24 acoustic modem. Available: https://www.evologics.de/en/products/acoustics/s2cr_12_24.html, 2018. [Online]. Accessed: March 12, 2018.
- [88] Hanif D Sherali, Allen L Soyster, and Frederic H Murphy. Stackelberg-nash-cournot equilibria: characterizations and computations. *Operations Research*, 31(2):253–276, 1983.
- [89] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685, 1993.
- [90] M Figueiredo and F Gomide. Adaptive neuro-fuzzy modeling. In *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, volume 3, pages 1567–1572. IEEE, 1997.
- [91] Steven Nahmias. Fuzzy variables. *Fuzzy sets and systems*, 1(2):97–110, 1978.

- [92] B Liu. Theory and practice of uncertain programming, 2002. *Incorporated: Springer Publishing Company*, 2.
- [93] Cheng Wang, Wansheng Tang, and Ruiqing Zhao. On the continuity and convexity analysis of the expected value function of a fuzzy mapping. *Journal of Uncertain Systems*, 1(2):148–160, 2007.
- [94] Baoding Liu and Yian-Kui Liu. Expected value of fuzzy variable and fuzzy expected value models. *IEEE transactions on Fuzzy Systems*, 10(4):445–450, 2002.
- [95] Leonid Maksimovich Brekhovskikh, Yu P Lysanov, and Robert T Beyer. Fundamentals of ocean acoustics. *The Journal of the Acoustical Society of America*, 90(6):3382–3383, 1991.

Acronyms

UWSNs	Underwater Sensor Networks	1
MCL	Monte Carlo Localization	ix
WSNs	Wireless Sensor Networks	6
GPS	Global Positioning System	6
IoT	Internet of Things	6
MCB	Monte Carlo localization Boxed	8
AoA	Angle of Arrival	8
ToA	Time of Arrival	8
RSSI	Received Signal Strength Indication	8
WLAN	Wireless Local Area Network	21
RFID	Radio-Frequency Identification	21
GRNN	Generalized Regression Neural Network	21
SVR	Support Vector Regression	21
LR	Linear Regression	22
RSS	Received Signal Strength	22
PDR	Pedestrian Dead Reckoning	22
RNNs	Recurrent Neural Networks	23
RMSE	Root Mean Squared Error	31
EVS	Explained Variance Score	31
MAE	Mean Absolute Error	31
OLTC	Opportunity Localization Topology Control	41
3DUL	three-Dimensional Underwater Localization	41
AUVs	Autonomous Underwater Vehicles	43
HMM	Hidden Markov Model	44
3D	three Dimension	44
UWSN	Underwater Sensor Network	44

UAN	Underwater Acoustic Network	62
MCM	Meandering Current Mobility	62
ALC	Average Localization Coverage	70
AEN	Average Energy per Node	70
ALD	Average Localization Delay	70
ANFIS	Adaptive Neuro-Fuzzy Inference System	72

List of Figures

- 2.1 Sample situation with two anchor nodes and one sensor nodes. Red are filtered particles and green are admissible particles, ©2016 IEEE. 12
- 2.2 Tri-MCL error relative to radio range over varying ring width and V_{Max} , ©2016 IEEE. 16
- 2.3 Tri-MCL error compared to MCL error over varying V_{Max} with a ring width of 0.25 m, ©2016 IEEE. 16
- 2.4 Tri-MCL error compared to MCL error over varying V_{Max} with a ring width of 2 m, ©2016 IEEE. 19

- 3.1 Example of cart regression tree. 24
- 3.2 Proposed Twi-AdaBoost algorithm based on the collaborative exploitation of smart-phone and smart-watch. 26
- 3.3 Interactive training process of Twi-AdaBoost across datasets from both smart-phone and smart-watch. 28
- 3.4 Map of the data collecting environment. 30
- 3.5 Illustration of a valid possible connection between samples of each dataset after pre-processing. 31
- 3.6 RMSE of position x estimation using Twi-AdaBoost and AdaBoost.R2 methods. 33
- 3.7 RMSE of position y estimation using Twi-AdaBoost and AdaBoost.R2 methods. 34
- 3.8 EVS of position x estimation using Twi-AdaBoost and AdaBoost.R2 methods. 34
- 3.9 EVS of position y estimation using Twi-AdaBoost and AdaBoost.R2 methods. 35
- 3.10 Box-and-whiskers plots of the position offsets using AdaBoost.R2 based on the smart-phone dataset. 36
- 3.11 Box-and-whiskers plots of the position offsets using AdaBoost.R2 based on the smart-watch dataset. 36

3.12	Box-and-whiskers plots of the position offsets using AdaBoost.R2 based on the mixed dataset.	37
3.13	Box-and-whiskers plots of the position offsets using Twi-AdaBoost based on the mixed dataset.	37
4.1	Deployment scenario of the proposed EELA scheme.	45
4.2	A scenario depicting the 'two-hop' transmission power calculation.	61
4.3	Localization coverage.	66
4.4	Average energy consumption per Sensor Node (SN).	67
4.5	Average energy consumption per Anchor Node (AN).	68
4.6	Average energy consumption per node.	69
4.7	Average localization error of EELA with different speed.	71
4.8	ANFIS structure of sensor nodes.	73
4.9	Architecture of the proposed Adaptive EELA scheme.	82
4.10	Localization coverage of scene A.	86
4.11	Localization coverage of scene B.1.	87
4.12	Localization coverage of scene B.2.	87
4.13	Average energy consumption per node of scene A.	89
4.14	Average energy consumption per node of scene B.1.	90
4.15	Average energy consumption per node of scene B.2.	90

List of Tables

- 2.1 Table of symbols, ©2016 IEEE. 10

- 3.1 Number of total samples recorded. 29
- 3.2 Comparison results among Twi-AdaBoost and the state-of-the-art. 38

- 4.1 Symbols used in the proposed EELA scheme. 46
- 4.2 Simulation parameters. 62
- 4.3 Average localization delay and error 70
- 4.4 Average performance of EELA with different speed. 70
- 4.5 Different network topologies used to verify Adaptive EELA. 83
- 4.6 Average localization delay (s). 91
- 4.7 Average localization error (m). 92

List of Algorithms

- 1 Construction of Cart regression model 24
- 2 AdaBoost.R2 25
- 3 Multi-Device AdaBoost Algorithm 27

- 4 Topology control of a sensor node (leader) 60
- 5 Topology control of an anchor node (follower) 61

Curriculum Vitae of Yali Yuan

Telematics group, Goettingen University
Goldschmidtstr. 7 37077 Goettingen, Germany

yali.yuan@informatik.uni-goettingen.de
+49-15905364105

Education Background:

Goettingen University, Germany

- Doctoral study in Telematics Group, Institute of Computer Science, Sep. 2015-Sep. 2018.
- Major research direction: Wireless Networks & Machine Learning.

Lanzhou University, China

- Graduate study in Computer Software and Theory, School of Information Science & Engineering, Sep. 2012-Jun. 2015.
- Major research direction: Wireless Networks & Machine Learning.
- Core disciplines: Network Theory and Technology, Advanced Artificial Intelligence, Data Warehouse and Data Mining, Advanced Computer System Structure, Multimedia Technology.

King Abdullah University of Science and Technology (KAUST),

Saudi Arabia

- Intern student in MINE Lab, Division of Computer, Electrical and Mathematics Science & Engineering, May. 2014-Nov. 2014.
- Major research direction: Machine Learning and Data Mining.
- Core disciplines: Pattern Recognition and Machine Learning.

Zhengzhou University of Light Industry, China

- Undergraduate study in Software Engineering, College of Software, Sep. 2008-Jul. 2012.
- Cumulative overall **GPA: 3.85/4.0**.
- Core disciplines: All the courses required in Software Engineering, including C/Java/JSP programming, Data Structure, Database Theory, Software Engineering, Principles of Computer Composition, and so on.

Selected Publications:

- 1) **Yali Yuan**, Chencheng Liang, Megumi Kaneko, Xu Chen, Dieter Hogrefe, Topology Control for Energy Efficient Localization in Mobile Underwater Sensor Networks using Stackelberg Game, IEEE Transactions on Vehicular Technology. (Under Second Review) (Impact factor: 4.432)

- 2) **Yali Yuan**, Liuwei Huo, Zhixiao Wang, Dieter Hogrefe, Secure APIT Localization Scheme against Sybil Attacks in Distributed Wireless Sensor Networks, Journal of IEEE Access. (Published) (Impact factor: 3.557)
- 3) **Yali Yuan**, Liuwei Huo, Georgios Kaklamanos, Dieter Hogrefe, Semi-supervised Tri-Adaboost Algorithm for Network Intrusion Detection, Journal of Computers and Electrical Engineering. (Under Review) (Impact factor: 1.747)
- 4) **Yali Yuan**, Christian Melching, Yachao Yuan, Dieter Hogrefe, Multi-Device Fusion for Enhanced Contextual Awareness of Localization in Indoor Environments, Journal of IEEE Access. (Published) (Impact factor: 3.557)
- 5) **Yali Yuan**, Liuwei Huo, Dieter Hogrefe (2017), Two Layers Multi-class Detection Method for Network Intrusion Detection System, In 22nd IEEE Symposium on Computers and Communications (ISCC 2017), Crete, Greece, Jul. 2017. (Published)
- 6) Ruoxing Li, Yachao Yuan, Wei Zhang, **Yali Yuan**, Unified Vision-based Methodology for Simultaneous Concrete Defect Detection and Geo-Localization, Journal of Computer-Aided Civil and Infrastructure Engineering. (Published) (Impact factor: 5.475)
- 7) **Yali Yuan**, Georgios Kaklamanos, Dieter Hogrefe (2016), A Novel Semi-Supervised Adaboost Technique for Network Anomaly Detection, In 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2016), Malta, Malta, Nov. 2016. (Published)
- 8) Arne Bochem, **Yali Yuan (Co-first Author)**, Dieter Hogrefe (2016), Tri-MCL: Synergistic Localization for Mobile Ad-hoc and Wireless Sensor Networks, In 41st IEEE Conference on Local Computer Networks (LCN 2016), Dubai, UAE, Nov. 2016. (Published)
- 9) **Yali Yuan**, Caihong Li, Yi Yang, Xiangliang Zhang, Lian Li (2014), CAF: Cluster and A-star with Fuzzy for Lifetime Enhancement in Wireless Sensor Networks, Journal of Abstract and Applied Analysis. (Published)
- 10) Caihong Li, **Yali Yuan (corresponding author)**, Lulu Song, Yunjian Tan, Guochen Wang (2013), Mathematical Model Based on BP Neural Network Algorithm for the Deflection Identification of Storage Tank and Calibration of Tank Capacity Chart, Journal of Abstract and Applied Analysis. (Published)
- 11) Hongjun Yuan, **Yali Yuan** (2013), A Survey of Library Knowledge Transfer Efficiency, Journal of information science. (Published) (CSSCI)
- 12) Yunjian Tan, Caihong Li, Rong Zhao, Jinze Du, **Yali Yuan**, Fuqiang Mu, Lian Li (2013), An Optimal Energy-Efficient Path Selection Algorithm Based on Ant Colony-Genetic Optimization Scheme, 2013 2nd International Symposium on Electrical & Electronics Engineering (EESYM 2013). (Published)

Conference Activities:

*13-17 Nov. 2016 Joined in the 19th ACM International Conference on Modeling,

- Analysis and Simulation of Wireless and Mobile Systems (MSWiM) and made the presentation in Malta.
- *3-6 Jul. 2017 Joined in the 22nd IEEE Symposium on Computers and Communications (ISCC) and made the presentation in Crete, Greece.

Academic Skills:

- * Familiar with Python/Java/C/C++ and their design patterns.
- * Familiar with Matlab and Latex Programming.
- * Familiar with NS2 and NS3 network simulators.

Teaching Experiences:

- * Teaching assistant in the course of “Advanced Machine Learning”.
- * Advisor in “Seminar” and “Pro-seminar”.
- * Advisor in five master projects and theses, one bachelor project and thesis.

Awards and Honors:

- * Sep. 2008-Jul. 2009 Awarded National Motivational Scholarship.
- * Sep. 2008-Jul. 2009 Awarded First-Class Scholarship, ‘Miyoshi’ Student and Excellent Student Cadres.
- * Sep. 2009-Jul. 2010 Awarded Second-Class Prize of ACM Contest.
- * Sep. 2009-Jul. 2010 Awarded National Scholarship.
- * Sep. 2009-Jul. 2010 Awarded First-Class Scholarship, ‘Miyoshi’ Student and Excellent League Member.
- * Sep. 2009-Jul. 2010 Awarded First-Class Prize of National Mathematical Modeling Contest.
- * Sep. 2010-Jul. 2011 Awarded National Motivational Scholarship.
- * Sep. 2010-Jul. 2011 Awarded Second-Class Scholarship, ‘Miyoshi’ Student, and Excellent Student Cadres.
- * Sep. 2010-Jul. 2011 Awarded Certificate of Software Designer.
- * Sep. 2011-Jul. 2012 Awarded Excellent Graduates in Henan Province.
- * Sep. 2012-Jul. 2013 Awarded First-Class Scholarship.

Work Experiences:

- * Oct. 2010-Feb. 2012 Software programming in emergency center research & development department, Henan, China.
- * Jul. 2011-Aug. 2011 Teaching assistant in software training project in Zhengzhou University of Light Industry, Henan, China.
- * Mar. 2012-Aug. 2012 Engineer assistant for Java software development in Argumentum Software Development (SH) Co., Ltd, Shanghai, China.

Project Experiences:

***Oct. 2012-Aug. 2013 Design and Implement Architecture of Wireless Sensor Networks for Debris Flow Disaster Monitoring**

Description:

This project intends to cope with the deficiencies of the current early warning technology of debris flow disaster monitoring and combines the network as well as communication technology, such as various sensing devices, Internet of Things, 3S, and 3G. It employs the advanced technology, such as cloud storage, data fusion, and autonomous computer, implementing mass data storage as well as efficient intelligent analysis, so as to make early warning about debris flow disaster accurately, as well as track its tendency timely.

(Major participant)

Responsibility:

- Study the routing protocol of the nodes communication.
- Study the experiment design as well as theory of the displacement calculation scheme.
- Design and implement the software platform (involved technology: ExtJS4, Servlet, Struts, Spring, Hibernate, Spring, SVN).
- Other assistance works.

Achievement:

- Read many literatures as well as did lots of simulated experiment during the study of the routing protocol, obtaining two efficient routing protocol schemes.
- Designed a simple experiment during the study of the displacement calculation scheme and employed filtering method to analyze the practical collected data to find the best displacement scheme.
- Designed and implemented the software platform which was mainly employed to storage and handle the collected data, as well as obtained the debris flow inversion shows and learned the usage of the ExtJS4.
- Had a deeper understand of the wireless sensor networks mechanism through the practical network deployment as well as learned many algorithms, such as A-star method, fuzzy inference method, BP neural network, and clustering method.

***Aug. 2013-May. 2014 Design and Implement Intelligent Voice System of Community Banks**

Description:

This project provides a platform for the banks to serve people with better, more convenient services. **(Major participant)**

Responsibility:

- Write the related documents.
- Design and implement semantic recognition algorithms.

Achievement:

- Learned many matching algorithms.