



HAL
open science

Indoor localization in wireless sensor networks

Xiaowei Lv

► **To cite this version:**

Xiaowei Lv. Indoor localization in wireless sensor networks. Operations Research [math.OC]. Université de Technologie de Troyes, 2015. English. NNT : 2015TROY0009 . tel-03358916

HAL Id: tel-03358916

<https://theses.hal.science/tel-03358916>

Submitted on 29 Sep 2021

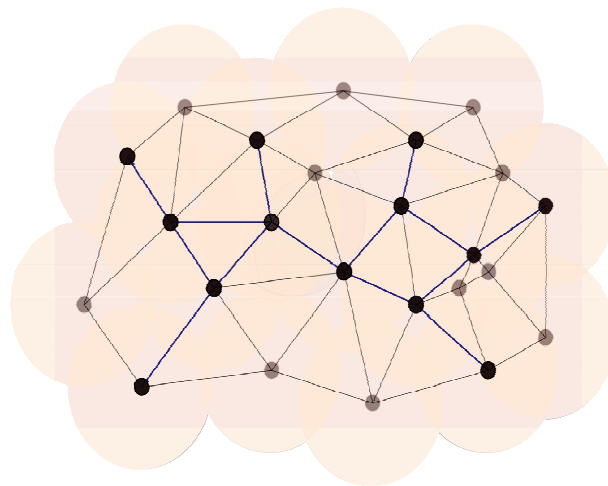
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse
de doctorat
de l'UTT

Xiaowei LV

Indoor Localization in Wireless Sensor Networks



Spécialité :
Optimisation et Sécurité des Systèmes

2015TROY0009

Année 2015

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Xiaowei LV

le 19 mars 2015

Indoor Localization in Wireless Sensor Networks

JURY

M. G. GELLÉ	PROFESSEUR DES UNIVERSITES	Président
M. P. HONEINE	MAITRE DE CONFERENCES - HDR	Examineur
M. O. MICHEL	PROFESSEUR DES UNIVERSITES	Rapporteur
Mme F. MOURAD-CHEHADE	MAITRE DE CONFERENCES	Directrice de thèse
M. C. PHAM	PROFESSEUR DES UNIVERSITES	Rapporteur
M. H. SNOUSSI	PROFESSEUR DES UNIVERSITES	Directeur de thèse

TO PEACE

Acknowledgements

Le temps passe vite. It feels just like a twinkling of an eye all this cherished days of doing a thesis. In this occasion, reviewing the passed works and lifes, the most desired word I want to say is "thanks", to all ones who generously contribute to this elaboration of thesis in all ways and on all sides.

I would like to proceed my sincerest gratitude to my supervisors, M. Hichem SNOUSSI and Mme Farah Mourad-CHEHADE, who led me into this marvelous research field of WSNs, for their wholehearted guidance, unlimited patience, generous supports, kindness and understandings. I greatly appreciate, and truly benefit from their profound knowledge, serious scientific attitudes, as well as the creativity and enthusiasms devoted to research. Thanks to Hichem, especially, for his systematic, veteran and insightful guidance and advices on my research. Thanks to Farah, in particular, for her generous, non-stop and dedicated guidance and helps on all aspects of my work all along these years, who sacrifices so much personal times that her families also own my deep sense of appreciations.

Great thanks to the jury members for their participation of the examination and evaluation of this work, and attendances from afar, particularly thanks to M. Olivier MICHEL of Grenoble INP and M. Congduc PHAM of University of Pau, for their works of reviewers, and M. Paul HONEINE of University of Technology of Troyes and M. Guillaume GELLE of University of Reims Champagne-Ardenne, for their works of examiners.

I felt genuinely the supports and understandings surrounding me at l'UTT. Thanks to M. Antoine GRALL, the director of the pôle ROSAS, M. Pierre BEAUSEROY, the director of the team LM2S and Mme Anne BARROS, the director in charge of the formation of specialty OSS, for their warm receptions delivered to the young candidates. Special thanks go to our dearest secretaries of the pôle ROSAS, Mme Véronique BANSE, Mme Marie-josé ROUSSELET and Mme Bernadette ANDRÉ, as well as the ones of doctoral school, Mme Isabelle LECLERCQ, Mme Pascal DENIS and Mme Therese KAZARIAN for their welcomes, kindnesses and amities. Thanks also to the technicians of department CRI and persons of SCD for their excellent services and helps.

Thanks to the wonderful times spent with my friends, my colleagues throughout all these years. Thanks to all the ones with whom we have ever supported

each other, understood each other and encouraged each other, and as it ever was, living the mutual life. I would like to address my thankfulness to, but not limited to, my "little partners", the senior fellows, tian, lei, jian, huan, wenjin, fanjuan, yue et haiyan..., the fellows, thai, anh, abeer, hui, mengyi, yingjun, yugong, hongchang et jiong..., as well as the adorable junior fellows aichun, tong, sandy, na, fei, heping, xiaoyi, nan, yubing, jianqiang, hajar, nassara, irce, canh and tan... Salute to the youths and sparkling dreams we have and ever had, to the struggles, suffers and enjoys.

I would like to thank also my landlady Mme De GRANRUT and her families, for their warmest reception, all kindness and consideration in all sides of living, making me feel at home. I owe my countless gratitude to my father who motivates me, inspires me, impresses me with the faith, hopes and fears, by his words and deeds, all along with my growth for ever, for all that he has done for me, thanks to my grandma, my aunts and my cousins for their endless love...

In this occasion, I would like to convey my thanks to the project UT-INSA cooperated between these excellent grande écoles in France and China Scholarship Council(CSC), thanks to the persons in charge of the affaires, particularly M. Regis LENGELLE and Mme Ling GONG, thanks to the financiers CSC, thanks all together to the staffs of Chinese Embassy and its affiliated Education Office in France, for all their concerns and cares to the chinese students living in France.

I would like to give all my thanks to all ones meant to encounter, to separate, and to souvenir forever, for the destiny belong to us somewhere over the rainbow...

Abstract

This thesis is dedicated to solve the localization problem in mobile wireless sensor networks. It works mainly with fingerprints features and inertial movements information. The former tackles the RSSIs values between sensors while the latter deals with the objects movement attitude by using accelerometer and gyroscope. The combination of both information is performed in terms of interval analysis, or Kalman filtering. The proposed work introduces three orders mobility model to approximate nodes trajectories using accelerations, combined then to the weighted K nearest neighbors algorithm in a centralized scheme. Then the mobility models are extended up to the inertial information taking into consideration the rotations of the nodes. A clusterized localization method is also proposed in the following in view of the working mechanism of large scale sensor networks. Finally, this thesis proposes a zoning localization method aiming at determining the zones in which the nodes reside. The proposed method addresses the zoning problem by using both the belief functions theory and the interval analysis.

Keywords: Sensor networks; estimation theory; interval analysis (mathematics); Kalman filtering; accelerometers; gyroscopes.

Résumé

Ce manuscrit est dédié à la résolution du problème de localisation dans les réseaux de capteurs sans fil mobiles. Les méthodes développées se basent principalement sur des caractéristiques de fingerprints ainsi que sur des informations de mobilité. Les premières s'attaquent aux valeurs de RSSI entre capteurs tandis que les deuxièmes prennent en considération la mobilité des capteurs mesurée à l'aide d'accéléromètres et de gyroscopes. La combinaison des données collectées est effectuée dans le cadre de l'analyse par intervalles, ou bien du filtrage de Kalman. Les travaux proposés introduisent des modèles de mobilité d'ordres un, deux ou trois, permettant d'approximer au mieux les trajectoires des capteurs à l'aide des accélérations mesurées. Ceux-là sont couplés à l'algorithme des K plus proches voisins, d'abord dans un système centralisé. Ensuite, les modèles de mobilités sont améliorés pour prendre en compte les rotations des nœuds. Une méthode de localisation clusterisée est également proposée dans ce qui suit, s'adaptant au mécanisme fonctionnel des réseaux de capteurs de grande échelle. Enfin, ce manuscrit propose une méthode de zonage visant à déterminer les zones dans lesquelles les capteurs résident. La méthode proposée aborde le problème de zonage en utilisant à la fois la théorie des fonctions de croyance et l'analyse par intervalles.

Mots-clés : Réseaux de capteurs (technologie); estimation, théorie de l'; calcul sur des intervalles; Kalman, filtrage de; accéléromètres; gyroscopes.

Contents

List of Figures	xiii
List of Tables	xv
I Introduction	1
I.1 Research background	2
I.1.1 Description of WSNs	2
I.1.2 Applications of wireless sensor networks	4
I.2 Illustration of the localization problem	7
I.2.1 Definition of the problem	7
I.2.2 Solutions of the localization problem	8
I.3 Content Organization	22
References	24
II Interval analysis	31
II.1 Introduction	32
II.2 Definition of intervals	33
II.3 Calculus of interval analysis	35
II.3.1 Basic tools	35
II.3.2 Basic algorithms	46
II.4 Conclusion	53
References	54
III Centralized localization	57
III.1 Introduction	58
III.2 Problem statement	60
III.2.1 Configuration of the network	60
III.2.2 Localization using fingerprints database	61
III.2.3 Localization using accelerations	62
III.3 Centralized localization algorithm	63
III.3.1 Reformulation over accelerations	64
III.3.2 Reformulation over fingerprints	68
III.3.3 Combination step	70

CONTENTS

III.4 Analysis and simulations	70
III.4.1 Generation of simulated data	71
III.4.2 Illustration of the proposed method	72
III.4.3 Analysis of measurements incertitude	73
III.4.4 Impact of parameters	79
III.4.5 Comparison to a connectivity-based method	84
III.5 Conclusion	84
References	86
IV Clusterized localization	89
IV.1 Introduction	90
IV.2 Configuration of the network	91
IV.3 Clusterized localization method	92
IV.3.1 Estimation using fingerprints	92
IV.3.2 Estimation using the inertial information	94
IV.4 Resolution using the interval analysis	96
IV.5 Resolution using the Kalman filter	98
IV.5.1 Kalman parameters definition	99
IV.5.2 Algorithm using Kalman	100
IV.6 Simulation results	101
IV.6.1 Illustration of the proposed method	103
IV.6.2 Influences of K , N_C , N_A , N_P and weight parameters	104
IV.6.3 Impact of additional noises	108
IV.6.4 Comparison to range-based methods	111
IV.6.5 Comparison to a connectivity-based method	112
IV.6.6 Comparison to fingerprint-based methods	113
IV.7 Conclusion	115
References	117
V Zoning localization	121
V.1 Introduction	122
V.2 Belief functions theory	123
V.2.1 Description of the theory	123
V.2.2 Combination rules	125
V.2.3 Decision rules	125
V.3 Zoning methodology	126
V.3.1 Configuration of the network	126
V.3.2 Fingerprints zoning	127
V.3.3 Accelerometer zoning	128
V.3.4 Final zoning	130
V.4 Simulation results	131
V.4.1 Generation of simulated data	131
V.4.2 Illustration of the proposed method	132
V.4.3 Impacts of parameters	133

V.4.4 Comparison with SVM	138
V.5 Conclusion	138
References	140
VI Conclusion and perspectives	143
VI.1 Conclusion	144
VI.2 Perspectives	145
A Appendix: Résumé en Français	149
A.1 Introduction aux réseaux de capteurs sans fil	150
A.1.1 Présentation générale	150
A.1.2 Illustration du problème de localisation	152
A.2 Théorie des intervalles	153
A.2.1 Définition des intervalles	154
A.2.2 Analyse par intervalles	155
A.3 Méthode de localisation centralisée	159
A.3.1 Problématique	160
A.3.2 L'algorithme	162
A.3.3 Analyse et simulations	164
A.4 Méthode de localisation clusterisée	165
A.4.1 Configuration du réseau	166
A.4.2 L'algorithme clusterisé	166
A.4.3 Analyse et simulations	170
A.5 Extension au zonage	171
A.5.1 Théorie des fonctions de croyance	172
A.5.2 Algorithme de zonage	174
A.5.3 Analyse et simulations	176
A.6 Conclusion et perspectives	177
A.6.1 Conclusion	177
A.6.2 Perspectives	179
References	181
References	189
.	189

CONTENTS

List of Figures

I.1	An example of a wireless sensor network	2
I.2	Three topologies of wireless sensor networks	3
I.3	Different application fields of wireless sensor networks	4
I.4	Sensors equipped with GPS	7
I.5	Network with anchors equipped with GPS in (a) or having fixed locations in (b)	8
I.6	Illustration of fundamental ranging methods	9
I.7	Illustration of RSSI ranging methods	10
I.8	Illustration of the positioning methods	11
I.9	Illustration of the diffusion method	12
I.10	Illustration of APIT method	13
I.11	Illustration of DV-Hop	13
I.12	Illustration of SDP	15
I.13	Illustration of euclidean propagation	16
I.14	Illustration of Stitch	18
I.15	Bounding Box I	20
I.16	Bounding Box II	20
I.17	Received signal strength with increase of distance (figure from [Ilyas and Mahgoub, 2004])	21
I.18	An example of pattern matching	22
II.1	Procedure of calculating $\sqrt{2}$	32
II.2	Problems attached. (a) demonstrates the case of two curves with crossing points, (b) demonstrates the case of a curved plan cut by a curve and (c) demonstrates the case of non-convex surface.	33
II.3	Two examples of interval	33
II.4	An example of 2D interval	34
II.5	Illustration of inclusion functions	39
II.6	Variety of formal inclusion functions resulting in different resulting intervals	41
II.7	A regular subpaving	43
II.8	A tree for regular subpaving	44
II.9	SIVIA	47

LIST OF FIGURES

III.1	A configuration of a centralized network	61
III.2	Interval-based localization	70
III.3	An example of acceleration signals	71
III.4	An example of the node's trajectory	72
III.5	Estimated boxes using the proposed method	73
III.6	Tested trajectories	75
III.7	Acceleration signals of trajectory 4	79
III.8	Estimation errors as functions of K	80
III.9	Estimation errors as functions of N_A	81
III.10	Estimation errors as functions of N_P	81
III.11	Estimation errors with different distributions of reference points	82
III.12	Estimation errors as functions of noise standard deviations σ_γ	82
III.13	Estimation errors as functions of noise standard deviations σ_ξ	83
III.14	Estimation errors as functions of weight parameters	83
IV.1	Composition of the clusterized network	91
IV.2	3D rotations	95
IV.3	Interval-based localization	99
IV.4	First trajectory	102
IV.5	Second trajectory	102
IV.6	Acceleration signals for the first trajectory in the top plots and for the second trajectory in the bottom plots	103
IV.7	Estimated results using the interval analysis and the Kalman filter for the first trajectory	104
IV.8	Estimated results using the interval analysis and the Kalman filter for the second trajectory	105
IV.9	Estimation errors (a) and times (b) as functions of K	106
IV.10	Estimation errors (a) and times (b) as functions of N_C	106
IV.11	Estimation errors (a) and times (b) as functions of N_A	107
IV.12	Estimation errors (a) and times (b) as functions of N_P	107
IV.13	Estimation errors (a) and times (b) as functions of the weight parameter	108
IV.14	Estimation errors as functions of noise standard deviations	109
IV.15	Ratios of estimation errors and computation times of the Kalman-based method over the connectivity-based one, with respect to anchors	113
V.1	Room-level localization	122
V.2	An example of a centralized zone network	127
V.3	An example of zoning using accelerometer information	130
V.4	Acceleration signals	131
V.5	Trajectory of the node	132
V.6	An illustration of altered reference position	135
V.7	Comparison between applications of acceleration incertitude	137

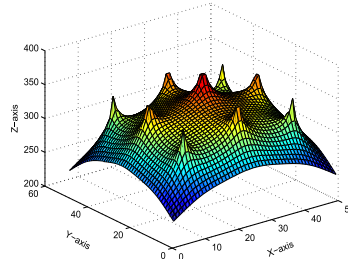
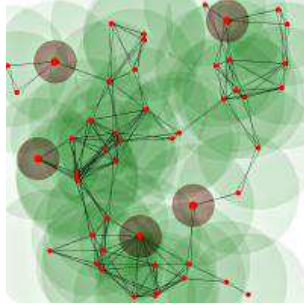
List of Tables

III.1	Simulation results using trajectory 1	73
III.2	Accelerations incertitude - Estimation results using trajectory 1	76
III.3	Accelerations incertitude - Estimation results using trajectory 2	76
III.4	Accelerations incertitude - Estimation results using trajectory 3	76
III.5	Accelerations incertitude - Estimation results using trajectory 4	76
III.6	Accelerations incertitude - Estimation results using trajectory 5	76
III.7	Accelerations incertitude - Estimation results using trajectory 6	76
III.8	Estimation errors using trajectories 1 to 6 with velocities modification	76
III.9	Fingerprints incertitude - Estimation results using trajectory 1	78
III.10	Fingerprints incertitude - Estimation results using trajectory 2	78
III.11	Fingerprints incertitude - Estimation results using trajectory 3	78
III.12	Fingerprints incertitude - Estimation results using trajectory 4	78
III.13	Fingerprints incertitude - Estimation results using trajectory 5	78
III.14	Fingerprints incertitude - Estimation results using trajectory 6	78
III.15	Comparison results	84
IV.1	Comparison of the proposed method to only fingerprinting-based and only acceleration-based methods	105
IV.2	Impact of acceleration noises with $\sigma_\gamma = 0.01m/s^2$, $\sigma_\xi = 1dB$, $N_A = 16$, and $N_P = 100$	110
IV.3	Impact of the orientation noise with $\sigma_\gamma = 0.01m/s^2$, $\sigma_\xi = 1dB$, $N_A = 16$, and $N_P = 100$	110
IV.4	Comparison of the proposed method to range-based methods with $N_A = 16$	112
IV.5	Comparison of the proposed method to range-based methods with $\sigma_\xi = 1dB$	112
IV.6	Comparison to fingerprinting-based methods with $N_A = 16$ and $N_P = 100$ while varying σ_ξ	114
IV.7	Comparison to fingerprinting-based methods with $\sigma_\xi = 1dB$ and $N_P = 100$ while varying N_A	115
IV.8	Comparison to fingerprinting-based methods with $\sigma_\xi = 1dB$ and $N_A = 16$ while varying N_P	115
V.1	Zones hitting rate over 100 seconds trajectory	133
V.2	Zones hitting rate while varying N_Z	134

LIST OF TABLES

V.3	Zones hitting rate while varying N_P	135
V.4	Zones hitting rate while varying N_A	136
V.5	Zones hitting rate while varying σ_ξ	136
V.6	Zones hitting rate while varying σ_γ	137
V.7	Zones hitting rate with δ_γ and δ_{x_γ}	138
V.8	Comparison to the SVM methods	138

Introduction



Contents

- I.1 Research background**
 - I.2 Illustration of the localization problem**
 - I.3 Content Organization**
 - References**
-

Wireless sensor networks have been promoted and prosperously evolved since the last two decades. They were voted as one of the ten breakthrough technologies ten years ago with continuous improvements towards faster, larger and stronger capabilities. Nowadays, the wireless sensor networks are applied for all walks of life, as for instance in monitoring machinery in the factories, tracking the patients in the hospitals, geological disaster reliefs, migration surveillance or space exploration and so on. Grateful for the massive advances in the field of micro-electromechanical systems, embedded micro-processors and wireless communications, these networks are designed with components of low-cost, low-consumption, tiny volume and a variety of functions to meet the requirements of the various applications. However they have meanwhile irresistible defects of limited storage and computational capability, leading to constraints that all the operations in a network should obey. A fundamental issue in wireless sensor networks is the self-localization of the sensors, which offers the geographical locations of networks nodes. This issue influences a priori and eventually the networks ultimate performance since almost all sensors measurements are position-related. There exist amounts of researches on localization algorithms relevant to different scenarios and applications. In this thesis, the objective is to solve the problem of localizing the networks nodes in an indoor environment while achieving higher positioning accuracy with less costs. This chapter begins with a description of wireless sensor networks. Next, it illustrates the localization problem then presents several existing methods.

I. INTRODUCTION

I.1 Research background

All the technologies are created and developed to improve the world's lifestyle. With no exception, Wireless Sensor Networks (WSNs), born in the 1990's, have never gave up to revolutionize the living's life. During the dozen years, WSNs develop generations after generations, from simple and rigid to more flexible and intelligent, along with the rapid developments of micro-electromechanical system, embedded micro-processors and wireless communication techniques [Akyildiz et al., 2002; Zhao and Guibas, 2004]. WSNs provide an intelligent solution for many applications since they are easy to deploy with the uselessness of a fixed infrastructure to connect their nodes. This section introduces first general information about WSNs then it discusses the involved problem about localization.

I.1.1 Description of WSNs

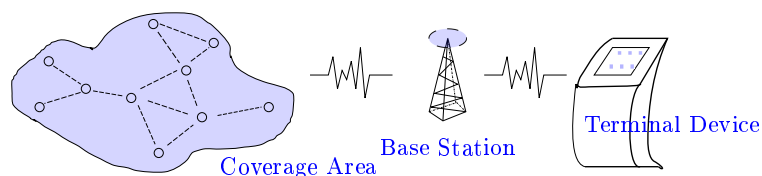


Figure I.1: An example of a wireless sensor network

WSNs are networks composed of a large number of wireless sensor nodes. These sensors have small sizes, low-cost and low-consumption, they are able to sense, process and communicate with other nodes via typically radio frequency [Ilyas and Mahgoub, 2004]. They are also called smart sensors, essentially consisting of sensation, computation, communication and power supply units [Bharathidasan and Ponduru, 2002]. Sensors in WSNs are universally motivated by carrying a power resource like nonrenewable batteries. This leads to their limited energy supplies, implying limited sensing and communication ranges. They have also limited computational capabilities, because of their small volume. The wireless nodes could be largely and densely deployed in the area of interest. See Figure I.1 for an example. They are prone to failure by reasons of hardware breakdown, exhausted batteries, radio interference or environment factors as for the network topology could change frequently. The topology could also change with specific intentions like extending the interest coverage or having additional sensors. However WSNs still have high level of fault tolerance due to the massive number of nodes, cooperation efforts between sensors and the strong self-organizing capabilities, allowing the compensation of the influence of failure of minority nodes and improving the sensing quality [Akyildiz et al., 2002; Stojmenovic, 2005].

There exist essentially three styles of topology architectures in WSNs: centralized, distributed and hybrid [Mourad, 2010], each corresponding to the provided services. In the centralized networks, all the sensed data are transmitted to the central fusion station

and computed over there. Normally nodes in that kind of networks are only capable of sending and receiving measurements therefore they do not perform computation or processing. These networks could provide high quality processes but consume extremely large energies, especially for the nodes nearby the central station. The distributed architecture permit nodes to execute specified computations at the node itself. They could exchange the signals among their neighbors within their communication range and carry out necessary computations at appointed nodes. This kind of network is much more robust to failure than the former in the case of failure of the fusion center or interference of measurements during the transmission. It also permits the parallel computing and allows to introduce the individual agents proving much more privacy protection for decentralized data. However, such a topology needs distributed algorithms that might be more complex to develop and sometimes less accurate than the centralized ones. The third topology combines the ideas of centralized and distributed architectures. Practically, information transmission consumes more energy than computation. Under this rule, the whole network is partitioned into several groups, or clusters. Each of them owns a head or cluster-head that plays the role of central fusion station. Providing an optimum balance between energy consumption and processing capacity, the hybrid topology mostly fits the large-scale and complicated organized wireless sensor networks [Bandyopadhyay and Coyle, 2003]. Figure I.2 illustrates the three types of network architecture.

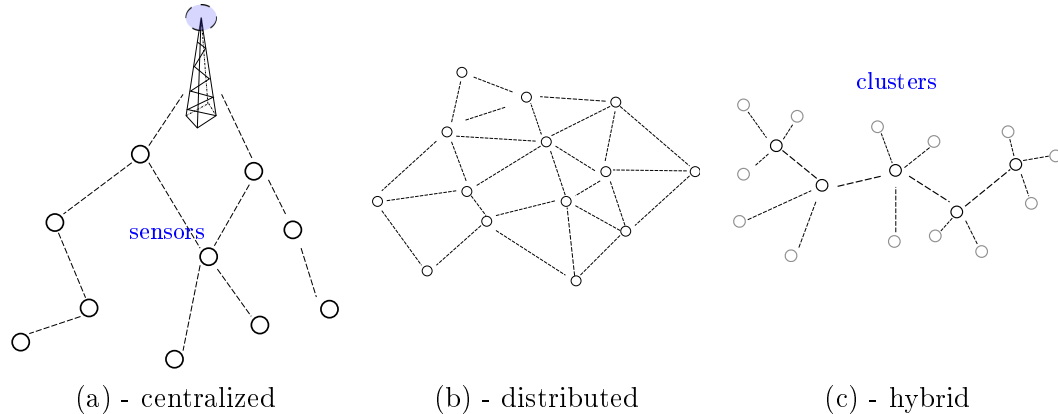


Figure I.2: Three topologies of wireless sensor networks

On the basis of networks architecture, WSNs communication is established through specified transmission medium to maintain the networks connectivity. Radio frequency communication is a basic communication mode. Based on the modulation and demodulation techniques, it achieves low energy consumption with simple hardware but with low data rate during the signal transmission. Moreover the path loss of the transmitted signals between two nodes in the communication range could be seriously up to fourth order exponent of the distance between them [Akyildiz et al., 2002]. The reduction of the path loss influence is always a challenge issue from the views of concrete hardware design and abstract signal transmission [Zhang and Pathirana, 2013]. Zigbee technique

I. INTRODUCTION

[Cheong et al., 2011] based on IEEE 802.15.4 protocol could be an alternative choice in case of deploying a large number of nodes, which is also in low-consumption, low-rate and low-cost but with weak penetrating ability in indoor surroundings. Comparatively, the Ultra-WideBand (UWB) [Chehri et al., 2009], which serves for the measurements of time arrival, is capable of penetrating thick walls and insensitive to channel fading, however in cost of relatively higher consumption. Bluetooth [Erasala and Yen, 2002] is anti-interference but requires additional device and consumes and costs more than the latter. Therefore it is not suitable for a large-scale sensor network where equipping massive nodes with additional devices costs a lot. Comparing with the technique mentioned before, WiFi, based on IEEE 802.11 protocol, is highly reliable but relies on powerful communication station and access nodes [Li et al., 2011]. Other options exist such as that the use of Radio-Frequency IDentification (RFID) [Choi et al., 2011], infrared [Gao and Guo, 2010] or ultrasound [Shahid et al., 2010; Runge et al., 2011]. This kind of approach requires correspondingly additional devices embedded to the sensor. They could achieve rather good precision and normally are limited to some specified scenarios like underwater, jungles or hand-held terminals. Practically, no matter what kind of communication protocol is adopted in a wireless sensor network, the key issue is how well it suits for the exposed environment to serve for the transmission and measurements. As the French proverb says, "*chacun son métier*".

I.1.2 Applications of wireless sensor networks

Research ultimately comes down to practice. The application area of WSNs falls in a wide range involving military, environment, medical, biology, civil life, space exploration and so on, performing a set of complicated information processing tasks categorized possibly into monitoring and tracking. This part cites as follows several classical examples of WSNs applications existing in that diverse domains [Akyildiz et al., 2002; Yick et al., 2008]. Figure I.3 illustrates some applications of WSNs.



Figure I.3: Different application fields of wireless sensor networks

Military applications : The characteristics of rapid deployment, self-organization

and high fault tolerance of WSNs promise to make quick response to the constant changes in the battlefield environments. In the first place, it helps in monitoring and further managing the status of friendly troops, equipments or ammunitions by attaching to them the smart sensors. All the sensed information are gathered and aggregated in the networks, then forwarded to the commanders to make the proper decision of action. Similarly, deploying the sensors in critical areas with enemy troops, vehicles and supplies could keep a close watch of all the movements of opposing forces, recognize the possible attack intension early and response quickly. In case of crucial combat conditions where the nuclear, biological or chemical attack occur, the multifunction sensor could sense polluted areas timely and accurately, detect the concentration levels, warn the friendly forces and prevent the human exposure in such areas. A counter-sniper systems is proposed in [Simon et al., 2004] aiming at the detection and the localization of the snipers. It measures the time of arrival between muzzle blast or shock waves then computes shooter's hiding location. The army of US have carried out a set of tests using WSNs at the desert of California [Hill, 2003]. The DARPA (Defense advanced research projects agency) of US has early launched the SensIT (Sensor information technology) project to construct ubiquitous sensor networks for realizing all kinds of measurements like light, sound, vibration, magnetism, humidity, pressure, temperature, pollution, acceleration, etc.

Environmental applications : The most famous project in recent years concerning the wildlife habitats is the great duck island project [Mainwaring et al., 2002]. This project is conducted by the the team of computer engineers from the university of California Berkeley to improve the research of the biologists at the college of the Atlantic on the nesting behaviors of seabirds. Traditionally, in order to make their studies, the biologists have to travel through the island frequently and in person, which probably have disturbed the seabirds regular nesting behaviors and even induced them to abandon their home. With this project, the networks consisting of 190 wireless sensors at the island take place of man-work at the stage, biologists could check on the birds in their office a few tens kilometers off the island, browsing data from the sensors linked with satellite and sharing their observations with their colleagues. Another similar application is ZebraNet [Juang et al., 2002; Zhang et al., 2004], where sensors are attached to the zebras collars to study their migration as well as their interaction with other species. At the university of Hawaii, the PODS project [Biagioni, 2001; Biagioni and Bridges, 2002] uses WSNs to observe and further determine the critical factors influence on the growth of endangered species of plants. Based on acoustic techniques integrated with TinyOS systems, a case of underwater monitoring is developed in [Vasilescu et al., 2005] to monitor coral reefs and fisheries in a long term. In Sonoma California, WSNs participate into the study of redwood trees [Tolle et al., 2005] by measuring several kinds of physiological signs like air temperature, relative humidity, solar radiation and more at different spacial gradients around the tree. Besides the application about flora and fauna, some other sensors, like

I. INTRODUCTION

seismic sensors, could detect the occurrence of an earthquake and monitor the earthquakes direction and magnitude. Others could monitor the air quality, track water pollution, detect forest fires or toxic leakage. In 2001, Kris Pister and his Berkeley team conducted an imitated experiment in the Mojave desert of southern California based on the UAV (unmanned aerial vehicles). They air-dropped six wireless sensors along a road over the vicinity of the accident site. Once the network is self-organized, it initiates the detection of passing vehicles then reports to the UAV.

Health care applications : Biomedical WSNs are designed to measure a variety of vital signs for a patient in bed for in-time surveillance in [Hande et al., 2006]. The sensors are integrated with the device of electrocardiogram (ECG) for measuring the blood pressure. Additionally, it is equipped with the solar cells panels facilitating self-charging in need. Patients equipped with the sensors, as a consequence, are ideally under 24h non-intermittent supervision even though they move around freely and independently. More over, in case of accidents such as sudden slip or falls, sensor would notify the base station immediately and call for immediate medical attention such as alerting a doctor to make a quick response to the emergency. Another application [Schwiebert et al., 2001] adopts a kind of glucose meter, embedded within patients to continuously observe blood-sugar level trends for diabetes surveillance. Meanwhile the indicator of corresponding measurements is relayed to the wristwatch display to remind the host in case an abnormal concentration of blood-sugar occurs. The biomedical WSNs highly improve the health care quality. They could serve equally for monitoring fire-fighters vital signs, alerting the disabled abnormal event in a unattended environment, ensuring the baby-sitting, tracking the activities of people with alzheimer disease, etc.

Civil applications : The smart sensors networks promise an intelligent lifestyle. That involves the surveillance of residence, transportation, industry and all kinds of critical infrastructures in daily life. Specifically, energy control in offices buildings reduces energy consumption by two quadrillion british thermal units, in the US, saving up to 55 billion dollars per year and reducing 35 million metric tons of carbon emissions [Rabaey et al., 2000]. [Viani et al., 2011] proposes a WSN-based system for early alert of road incident by reason of wildlife road crossing. [Dagher et al., 2014] conceives an ubiquitous navigation system based on WSNs and smart city concept. It provides a local navigation service for the citizens driving a vehicle with aid of the smart street lightning system. Last but not least an industrial sensor network is designed in [Krishnamurthy et al., 2005] for the view of equipment maintenance.

I.2 Illustration of the localization problem

Handling with the issue of surveillance or tracking, WSNs reports eventually the location where the detected event occurs in real-time. This objective requires in fact the prior knowledge of sensors location themselves, which are in most real cases kind of unknown. The problem of determining the nodes locations is referred to the localization problem [Yick et al., 2008].

I.2.1 Definition of the problem

In some applications of WSNs, sensor nodes could be deployed manually beforehand at expected positions. However, on request of the application in diverse domains as mentioned above, sensors nodes are generally randomly scattered into a specified place, such as in the air-drop manner, afterwards they self-organize into an efficient communicable network. Moreover, in some cases, the sensors move in requirements or unexpectedly by uncontrolled outside factors such as the airflow or the vibration nearby. As a consequence, sensors stay generally in the statue of unawareness of their actual positions at the very beginning of organizing networks or during the period of reconstructing networks. Hence, the capability of sensor nodes for self-localization is expected to be a key issue in playing the role of monitoring or tracking.

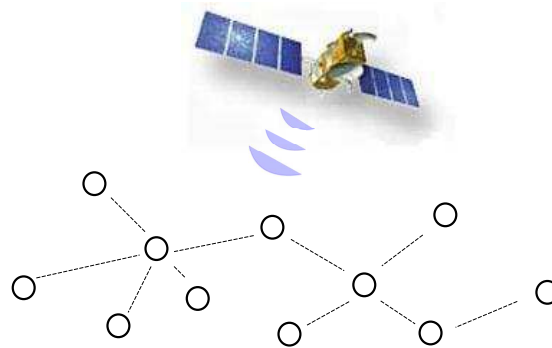


Figure I.4: Sensors equipped with GPS

One intuitive solution to this problem is to equip all the sensors with Global Positioning Systems (GPS) [Hofmann-Wellenhof et al., 2004] to make an absolute positioning as shown in Figure I.4. However, this solution implies two impractical facts, one severely constrained with the place where the WSN is applied and the other with the hardware support. Namely, in case of obstacles blockage between the satellite and the sensors, GPS is incapable to obtain the accurate communication signals between the sender and the receiver. This is the case in indoor environments where GPS signals are completely not reliable or acceptable. In the other hand, a full set of GPS receivers would be needed to equip all sensors, which is highly-costing, extremely energy-consuming during working and also requiring special design. That device is thus bulky for the tiny sensors and too much expensive to furnish in a large-scale network.

I. INTRODUCTION

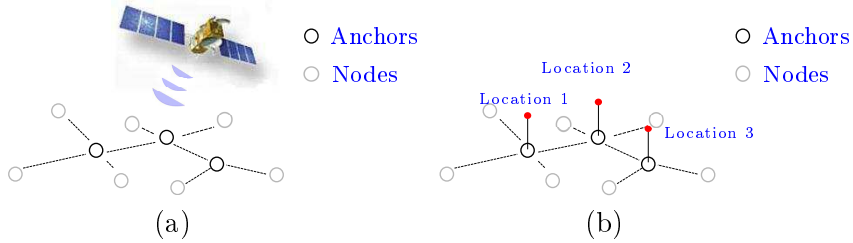


Figure I.5: Network with anchors equipped with GPS in (a) or having fixed locations in (b)

Because of the impracticality of GPS, alternative solutions serving for localization in WSNs have emerged. A majority of such approaches solve this problem with the help of **anchors**, which are sensors having positions known in advance. That could be by means of GPS in case of mobile outdoor networks as shown in Figure I.5-(a), or anchors could have fixed locations as in Figure I.5-(b). The other remaining sensors are called **nodes**, having unknown locations no matter fixed or moving. The key idea of localization here is to make use of the internodes communications, the sensors measurements and/or the information exchanged with the anchors, such as temperature, signal strengths, transmission time or angles, or even 1 or 0 denoting a detection of anchors or not, to determine nodes' locations.

Let us define the localization problem in an indoor scenario where a node j moves in a D -dimensional space included in \mathfrak{R}^D , with $D = 2$ or $D = 3$. Suppose there are N_A anchors and N_X nodes, that is $j \in \{1, \dots, N_X\}$. Each anchor i is denoted coordinately by $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,D})$, $i \in \{1, \dots, N_A\}$. Each node j is denoted similarly by $\mathbf{x}_j(t) = (x_{j,1}(t), \dots, x_{j,D}(t))$, where t denotes the discrete time-step. We assume also that anchors are fixed and that nodes use the exchanged information with anchors to be localized. Let $m_{j,i}(t)$ be the measurement of node j related to anchor i . Then $\mathbf{m}_j(t) = (m_{j,1}(t), m_{j,2}, \dots, m_{j,i}, \dots, m_{j,N_A})^T$ is the measurements vector collected by the node j at time t . We also denote in the following by $\mathbf{m}(t) = \{\mathbf{m}_1(t), \mathbf{m}_2(t), \dots, \mathbf{m}_j(t), \dots, \mathbf{m}_{N_X}(t)\}$ the set of all measurements vectors collected at time t . Then, the objective is to compute $\mathbf{x}_j(t)$ while having the positions \mathbf{a}_i and the corresponding measurements $\mathbf{m}_j(t)$. Different anchor-based techniques have been proposed to meet the diverse requirements of sensors in WSNs, such as those based on distance estimation, connectivity measurements, scenario information collection etc. In the following part, couples of such kinds of existing methods are introduced successively.

I.2.2 Solutions of the localization problem

The measurements $\mathbf{m}(t)$ provide practically kinds of possibilities to locate a wandering node. Some measurements could be transferred to distance information, others reflect the connectivity information. In the following, basic solutions are introduced first to give an intuitive recognition of sensors positions in the network. Then the connectivity based

methods and other range based methods follow. Other types of localization methods such as signal-strengths-based and scenario-based approaches are also included.

I.2.2.1 Basic solutions

Intuitively, according to the geographical geometry, it is possible to infer a node's location when its distances to at least three anchors are available. Meanwhile, the localization process is achieved under the assumption of $\mathbf{m}(t)$ relying on concrete measurements, that could be periods of elapsed time, relative angles, signal strengths, radio hops or other features obtained by all means of measurements between the sensors. In basic localization methods, the measurements are used to compute the relative range (or distance) between nodes and anchors, namely in the ranging phase, which is followed by the combination phase to obtain the nodes' positions. Methods using the distances between sensors in networks are called range based localization methods. At the very beginning, it is recommended to describe the classical ways in terms of different types of measurements techniques serving for range measurements. The combination of range is variant and presented later. There exist couples of fundamental ranging methods for distance estimation, based on Time of Arrival (ToA) [Guvenc and Chong, 2009], Time Difference of Arrival (TDoA) [Priyantha et al., 2000; Savvides et al., 2001], Angle of Arrival (AoA) [Niculescu and Nath, 2003; Priyantha et al., 2001], Received Signal Strength Indicator (RSSI) [Kaemarungsi and Krishnamurthy, 2004; Bahl and Padmanabhan, 2000] or radio hop-count which is illustrated later.

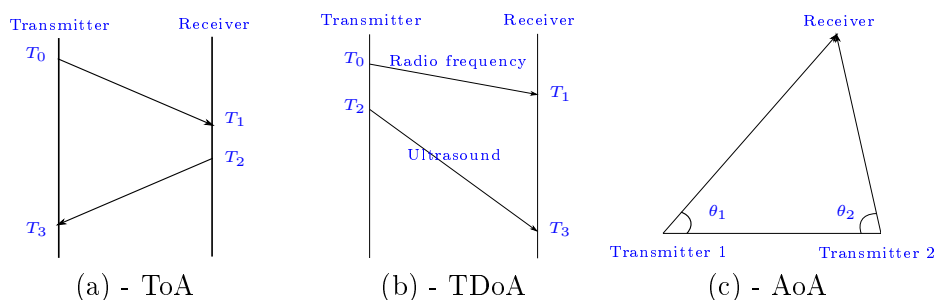


Figure I.6: Illustration of fundamental ranging methods

The ToA method records the traveled time of signals emitted by the anchors and received by the nodes. These signals could be radio frequency, acoustic, infrared, ultrasound and so on [Pal, 2010]. By knowing the propagation velocity of the emitted signal and its travel time, the ToA method consists of computing the distance separating the two sensors. The problem here is that ToA measures the absolute time of arrival at the receiver according to its clock, which needs exact synchronous clocks at both sensors. Inaccuracy in the synchronization process leads to inaccurate distances estimates. Respective distances of each node to three anchors are normally needed to get then the position estimates. When the synchronization of emitter and receiver clocks is not possible, two-ways-travel ToA could be applied, where an anchor sends first a signal to a node, then

I. INTRODUCTION

at reception, the node sends a signal of the same type (ultrasound for instance) back to the anchor. Here the absolute time of arrival is measured by the anchor and then divided by two to be used to get their separating distance. An illustration is given in Figure I.6-(a) and Eq. I.1. In order to overcome the multiple ambient environment factors, multipath, pathloss, interference and the like, the Ultra Wideband (UWB) signals could be used here due to their flexibility and their fine properties, specially in indoor communications, which benefits the final localization precision [Denis et al., 2006].

$$d = 0.5 * \nu * ((T_1 - T_0) + (T_3 - T_2)), \quad (\text{I.1})$$

where T_0, T_1, T_2 and T_3 are known values denoting respectively the emission time of the first signal of the anchor, its reception time by the node, the emission time of the second signal of the node and its reception time by the anchor and ν denotes the velocity of the signals.

The TDoA method expands ToA method by using the time difference of arrival of signals traveling between two sensors. Indeed, it consists of sending two different kinds of signals from an anchor at the same time then measuring at a receiving node the difference of their two times of arrivals as shown in Figure I.6-(b). The distance between the two sensors is computed by using the difference of arrival times and the individual propagation speeds of the signals, as shown in Eq. I.2. This method is impressively accurate under condition of LoS (light of sight) at the expense of equipping the sensors with additional hardware. Nevertheless, adopting the sound signal is not accurate to ranging on account of the varied sound speed in environment with air temperature, humidity and other environmental factors [Stojmenovic, 2005].

$$d = ((T_3 - T_1) - (T_2 - T_0)) / (1/\nu_{ultrasound} - 1/\nu_{radio}), \quad (\text{I.2})$$

where T_0 and T_2 denote the emission times of both signals, T_1 and T_3 their respective reception times and ν_{radio} and $\nu_{ultrasound}$ their respective velocities with known values. The AoA method measures the relative angles of arrival of signals exchanged between two anchors and a node as shown in Figure I.6-(c). That could be obtained through antennas phase interferometry or amplitude response [Rappaport et al., 1996; Koks, 2007]. Using the trigonometry laws of sine and cosines, one can get the position of the node by drawing the triangle constituted of the anchors and the node [Kulaib et al., 2011]. Then the relative distance between node to each anchors are determined.

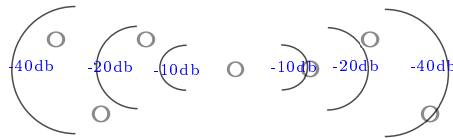


Figure I.7: Illustration of RSSI ranging methods

Other ranging methods based on the Received Signal Strength Indicators (RSSI) exist. These methods benefit of the property of the signal degradation while traveling, that

I.2 Illustration of the localization problem

is signal power decreases proportionally with the increase of the traveled distance separating the emitter and the receiver. Practically, each anchor broadcasts signals in the network with the same initial power. These signals are received by nodes with different RSSI, according to their distances to the anchor. Then the measured RSSI are converted to distances using the pathloss model [Kaemarungsi and Krishnamurthy, 2004]. One classical degradation model is shown in Eq. 1.3 where P_{d_0} is the power measured at 1 meter from the anchor, d is the corresponding distance between the sensors, n is the channel parameter and P_d is the RSSI measured at distance d in *dbm*.

$$P_d = P_{d_0} - 10 * n * \log(d). \quad (1.3)$$

Compared to the first three methods which need extra hardware like ultrasonic or synchronization equipments, RSSI-based methods overcome the burden of additional equipments hence they are easier to implement and less-costing than the others.

Given the anchors positions and their corresponding distances to a node, it is then possible to locate the node by trilateration (with three anchors) or multilateration (with more than three anchors) by intersecting the circles (or spheres in three-dimensions) centered on the anchors and having their distances to the node as radii [Pal, 2010]. Having the angles of anchors signals to a node using the AOA technique, one is also able to localize the node using the triangulation technique. In two-dimensions for example, the node would be located at the vertex of the triangle having two anchors as other vertices with their respective measured angles. These positioning methods are depicted in Figure I.8 in order of trilateration (a), triangulation (b) and multilateration (c).

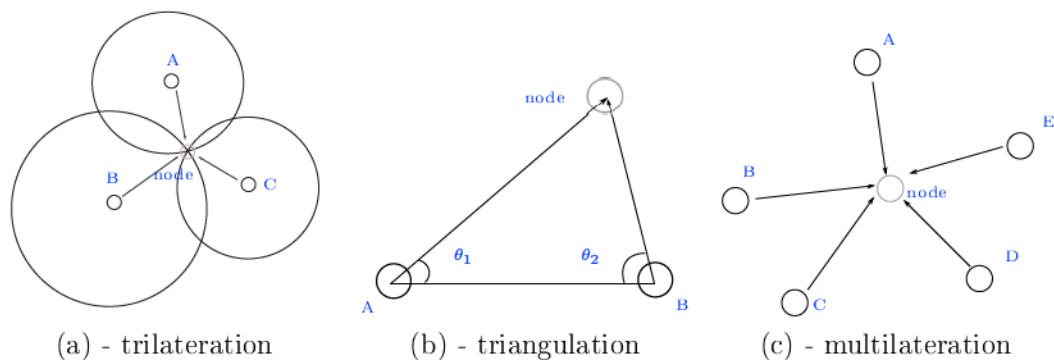


Figure I.8: Illustration of the positioning methods

I.2.2.2 Connectivity-based solutions

Other existing localization methods go towards the range free approach. They are also called connectivity-based methods since here the measurements $\mathbf{m}_j(t)$ are attributes of connection rather than concrete values about the signals. Indeed, the elements of $\mathbf{m}_j(t)$ denoted by $m_{j,i}(t)$ are supposed to be positive if anchor i and node j are capable to sense each other within a range r at time t . Under such definition, the connectivity-based

I. INTRODUCTION

methods come into being. These methods are more robust against signal perturbations, than the basic methods, since they use comparison results of signals information instead of their exact values. In the following, several connectivity-based methods are discussed.

- Unlike the basic solutions, connectivity-based methods do not need the ranging phase where distances to anchors or angles are computed. The contribution of [Bulusu et al., 2000] is such a method. It only takes into account the anchors that are in the communication range of each node. It then estimates a node's position by the centroid of the geographical coordinates of the connected anchors around it as shown in Figure I.9-(a). This technique leads to a blinding simplicity of algorithm no matter in terms of computation processing or hardware design. However, it results in a coarse estimation due to blind average operation, particularly when the density of anchors in networks is low, or in case of outliers where the node falls outside the convex of all its truly connected anchors as shown in Figure I.9-(b).

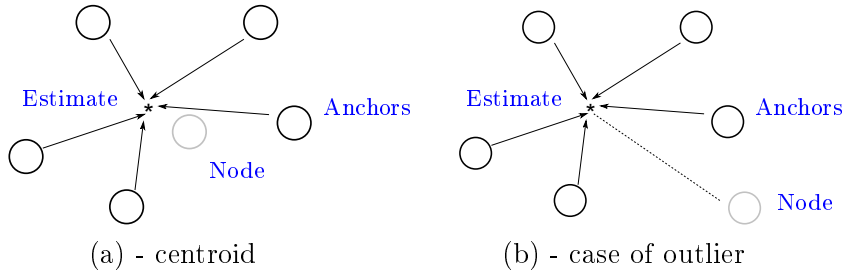


Figure I.9: Illustration of the diffusion method

In this context, works of [Bulusu et al., 2001, 2002] manage to improve the performance by an adaptive placement of anchors beforehand. Additionally, [Savvides et al., 2002] recommends to place several anchors around the edges of the full sensor networks in considering the network topological structure.

- Similarly to the described method, the gradient introduced in [Bachrach et al., 2004] employs equally the connectivity with anchors around a node. Furthermore, it constructs gradient information and estimates the distances between the node and at least three nonlinear anchors roughly by utilizing the minimum hop-count value $h_{j,i}$ from node j to anchor i and maximum sense range r associated with one single hop. Consequently, node j is at most distance of gradient value $h_{j,i} * r$ from anchor i . Then all of the gradient values could be combined to estimate the nodes position by multilateration algorithm.
- Another localization method that prevents the use of ranging processing is introduced in contribution [He et al., 2003]. As a range free locating method, moreover, it performs equally well even though when nodes and anchors are irregularly deployed over surveillance area due to its area-based idea cooperating also with

I.2 Illustration of the localization problem

anchors location information Figure I.10. The theoretical clue of this idea lies in that a node detects all of its connected anchors and then tests if it resides in any triangle composed by any arbitrary three detected anchors. This test, denominated the Point-In-Triangulation (PIT) test, consists of considering that a node is outside a triangle of three anchors if moving the node in any direction would increase/decrease its distances simultaneously to the three anchors. It is inside the triangle otherwise. By exhausting all the possibilities, the node defines the triangles where it resides and then by aggregating these triangles, it narrows down the possible triangular area where it is located. Its position is then estimated by the center of gravity of the obtained area. This method, called APIT, is illustrated in Figure I.10. This method has shown good estimation results, however the PIT test is not applicable when the node could not be moved in a controllable manner and here an approximated version of PIT is needed.

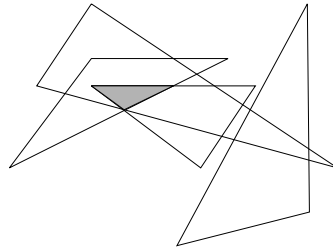


Figure I.10: Illustration of APIT method

- The connectivity idea could also be realized by implementing hop-count throughout the sensors topological network. [Niculescu and Nath, 2001] proposes a distributed style, similarly with the contributions presented above, however using hop-by-hop and Distance Vector (DV-Hop) routing as well as Landmarks of anchors to infer nodes location, namely the DV-Hop propagation algorithm divided into the following actions as shown in Figure I.11 .
 - The objective node builds a table $X_i, Y_i, H_{j,i}$ denoting the coordinates of the i -th anchor (or landmark as denoted in [Niculescu and Nath, 2001]) and the path length obtained by counting the number of hops connecting the node to its neighboring anchor L_j .

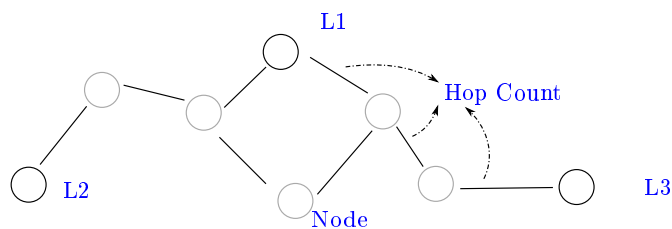


Figure I.11: Illustration of DV-Hop

I. INTRODUCTION

- Each anchor or landmark L_i estimates its own distance indicator per hop with help of the real distances as well as hop counts between its neighboring landmarks, which then flood the correction over the whole network.

$$C_i = \frac{\sum_{k \neq i} \sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2}}{\sum_{k \neq i} H_{i,k}}. \quad (\text{I.4})$$

- Node estimates separately its distances to at least three anchors by multiplying the path lengths with a distance indicator computed by the closest anchor.

$$\hat{d}_{j,i} = H_{j,i} * C_{closest}. \quad (\text{I.5})$$

- Once the distances to at least three anchors/Landmarks are available, node could estimate its position by trilateration or multilateral method as introduced in the basic solution part.

The DV-Hop method takes advantage of simplicity of hop measurements therefore it is very easy to carry out in a connectable networks. It performs well in isotropic networks where the property of graph remains the same in all direction, consequently however it brings constraints in application when the property of graph changes.

- The distance inferred by hop-count may also be obtained by another way. The contribution of [Nagpal et al., 2003] proposes a similar positioning method called Amorphous with DV-hop but with a different way of estimating the average distance per hop d_{hop} employing the local neighborhood n_{local} and communication range r , as shown in Eq. I.6,

$$d_{hop} = r \cdot (1 + e^{-n_{local}} - \int_{-1}^1 e^{-\frac{n_{local}}{\pi} (\arccost-t-\sqrt{1-t^2})} dt). \quad (\text{I.6})$$

Then the distance $d_{j,i}$ separating the node j from anchor i is approximated to be $d_{hop} * h_{j,i}$ where $h_{j,i}$ denoting the hop-counts from node j to anchor i . This contribution makes also theoretical analysis that a minimum average neighborhood size of 15 is required for achieving good accuracy. Meanwhile, the resolution of the coordinate system is also limited due to the strict determination from local communication.

- Contribution [Doherty et al., 2001] solves the localization problem by a distinct exploration. For instance, if two sensors are able to sense each other, it is concluded that their separation are definitively less than the communication range r . It represents then this geometric constraints, existing between two sensors in the networks, as linear matrix inequalities(LMIs). In this case, the radio constraints expression between sensors positions \mathbf{a} and \mathbf{b} at left is represented as equivalent LMIs [Boyd et al., 1994] at right in Eq. I.7,

$$\|\mathbf{a} - \mathbf{b}\|_2 \leq r \Rightarrow \begin{bmatrix} I_2 r & \mathbf{a} - \mathbf{b} \\ (\mathbf{a} - \mathbf{b})^T & r \end{bmatrix} \geq 0. \quad (\text{I.7})$$

where I_2 is the two-dimensional identity matrix. In a convex topology, connectivity of the network can be represented as a set of convex position constraints. The set of individual LMIs can be then combined/stacked in diagonal blocks to form one single semi-definite program (SDP) for the entire network. Note that the intersection of the convex sets is also a convex set. Hence, the localization problem is transferred to solve the convex SDP problem as Eq. I.8,

$$\begin{aligned}
 & \text{Minimize } \mathbf{c}^T \mathbf{x}, \\
 & \text{subject to } F(x) = F_0 + x_1 F_1 + \dots + x_n F_n < 0, \\
 & A\mathbf{x} < b, \\
 & F_i = F_i^T,
 \end{aligned} \tag{I.8}$$

where $\mathbf{x} = [x_1 \ y_1 \ \dots \ x_m \ y_m \ x_{m+1} \ y_{m+1} \ \dots \ x_n \ y_n]^T$, each couple of terms denoting the position (x, y) of one sensor. The principal of such evolution could be illustrated by graphs as shown in Figure I.12-(a). For node N , it possesses the radio constraints as area within the circle A and circle B . After combining/intersecting the individual geographic constraints sets A with B , the feasible area decreases to be smaller and more precise area as the overlapped shaded area. Then the rest of question remains to solve this optimized semi-definite program. The SDP problem is possible to be simplified to a rectangular bounding box which represents the possible set of this problem as shown in Figure I.12-(b).

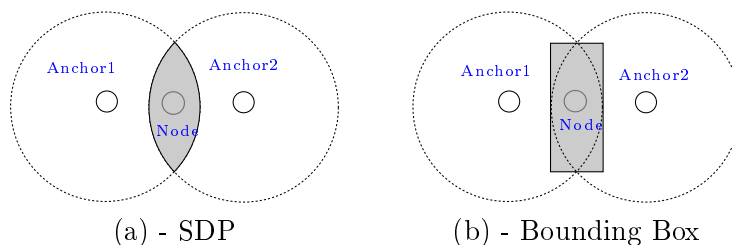


Figure I.12: Illustration of SDP

- Instead of semi-definite program, [Shang et al., 2003] solves this positioning problem by mathematical technique multidimensional scaling (MDS) which includes three steps.
 1. For all the sensor nodes, compute the shortest path length between all of possibilities of two pairs of nodes over the network and form a global path matrix.
 2. Construct a relative map by applying MDS technique to the path matrix obtained in the last step.
 3. Transform the relative map to absolute position with aid of known locations of at least three anchors participating in the global map.

I. INTRODUCTION

This method utilizes the full connectivity information all over the network, flexible when additional information is introduced. However it is executed in centralized scheme when gathering path map information thus leads to technical limitations in most applications. [Shang and Ruml, 2004; Shang et al., 2004] improves the original method by a MDS-P algorithm which builds local maps and then patches them into global map in a decentralized scheme. The latter acquires more accuracy than the former even when few anchors exist.

I.2.2.3 Propagation to range-based method

Considerably, the range-based methods are fundamentally based on the possibility of connectivity between sensors. Intuitively developed from this point of view, several methods mentioned above could be propagated to range-based methods even achieving better performance. Comparatively, couples of other original range-based methods are introduced in the following.

- As for the DV-Hop method [Niculescu and Nath, 2001], this contribution also proposes that the distance between neighboring nodes is able to be propagated by the Received signal strength instead of the distance vector hops. In that case, distance vector is in fact accumulated with signals traveling. It obtains higher accuracy comparatively with DV-Hop but it is inevitably kind of sensitive to measurements errors.

A third scheme based on the true Euclidean distance propagation from anchor/landmark L is equally developed in [Niculescu and Nath, 2001], in which the position is eventually computed through quadrilateral relation existing in a quad. As shown in Figure I.13, node A resides in the neighborhood of nodes B and C where B and C are neighbors also. Additionally, distance estimates from landmark L of B and C are also available through distance measurements. In that case, all of the outer edges of ABL are already known as well as one diagonal BC , consequently the distance AL is possible to be estimated by applying Pythagoras' generalized theorem in triangles ACB , BCL , and ACL of the quadrangular ABL . This method is considered to give the closest estimates compared to GPS method.

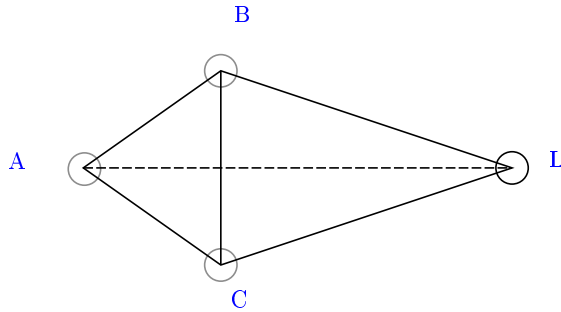


Figure I.13: Illustration of euclidean propagation

I.2 Illustration of the localization problem

The ranging measurements directly between neighboring nodes technically leads to more accurate estimates. Nevertheless, some estimates are acquired with accumulation of ranging errors as mentioned above. [Savvides et al., 2002] proposes using a cooperative multilateration for the purpose of relieving the error accumulation. It performs when the density of anchors over the networks is higher, strictly when a nodes hears at least three anchors or even more. The initial rough estimates are also improved by least square estimation at the end.

Furthermore, [Rabaey and Langendoen, 2002] proposes an in-house DV-Terrain method. A first hop step is taken over the network to overcome the sparse anchors embarrass. Then the algorithm gives the initial estimates by the way similar to DV-hop, followed by an iterative refinement algorithm which yields more accurate nodes positions through solving the least square triangulation equation using its own and neighborhoods estimates.

- A ranging-oriented semi-definite program (SDP) method is alternatively discussed in [Biswas and Ye, 2004], where the measured distance between neighboring nodes is employed to formulate further the semi definite program problem with tighter convex constraints. As a consequence, the localization problem is rescheduled as Eq. I.9,

$$\begin{aligned}
 \|\mathbf{x}_i - \mathbf{x}_j\|_2 &= \hat{d}_{i,j}^2, \|\mathbf{a}_k - \mathbf{x}_j\|_2 = \hat{d}_{k,j}^2, \forall (i, j), (k, j) \in N_e, \\
 \|\mathbf{x}_i - \mathbf{x}_j\|_2 &\geq \underline{r}_{i,j}^2, \|\mathbf{a}_k - \mathbf{x}_j\|_2 \geq \underline{r}_{k,j}^2, \forall (i, j), (k, j) \in N_l, \\
 \|\mathbf{x}_i - \mathbf{x}_j\|_2 &\leq \bar{r}_{i,j}^2, \|\mathbf{a}_k - \mathbf{x}_j\|_2 \leq \bar{r}_{k,j}^2, \forall (i, j), (k, j) \in N_u.
 \end{aligned} \tag{I.9}$$

where $\mathbf{a}_k \in \mathbb{R}^2, k = 1, \dots, m$, denotes anchors with known positions, $\mathbf{x}_j \in \mathbb{R}^2, j = 1, \dots, n$, denotes nodes with unknown positions. $\hat{d}_{i,j}$ is the measured Euclidean distance between nodes \mathbf{x}_i and \mathbf{x}_j whereas $\hat{d}_{k,j}$ is that between anchor \mathbf{a}_k and \mathbf{x}_j . $\underline{r}_{i,j}$ and $\bar{r}_{i,j}$ denotes separately the distance lower and upper bound between nodes \mathbf{x}_i and \mathbf{x}_j whereas $\underline{r}_{k,j}$ and $\bar{r}_{k,j}$ denotes that between anchor \mathbf{a}_k and node \mathbf{x}_j . N_e, N_l and N_u stands respectively for the set of indices which lead to the nodes distance to nodes, or anchors, satisfy the preceding equations or inequations. The localization problem remains to find a such \mathbf{x}_j satisfying the conditions above. Contribution [Liang et al., 2004] further improves the SDP-ranging results by a gradient search procedure.

- The MDS approach is similarly extended to incorporate distance measurements into location decision as in [Ji and Zha, 2004]. In that contribution, the whole network is required to be divided into several small parts in advance, where the adjacent groups may share common sensors. Then the pairwise distance is collected by RSSI measurements between all pairs of sensors in that part. Whereafter the MDS algorithm is used to estimate the relative location between sensors using the estimated distance. The local relative maps built in each part are stitched together to form a global relative map by utilizing the common sensors existing

I. INTRODUCTION

between adjacent local maps. The position alignment technique is employed at the last step to map the relative coordinates to physical coordinates.

- Similar to the idea of stitching small areas together, there exists contribution [Meertens and Fitzpatrick, 2004] which employs the same framework but differs in solutions. In this contribution, local map derived from distance information is constructed in each area at the beginning. Then align and reconcile a local map with its nearby maps which have some nodes in common but differ on the positions by minimizing the discrepancies between them throughout the transformation of translation, rotation and optional reflection.

$$discrepancy(L, M) \doteq \sum_{p \in C} |L_p - M_p|^2, \quad (\text{I.10})$$

Eq. I.10 shows the computation of discrepancy between map L_p and M_p , where C is the set of common nodes appearing in both maps. Other relative stitching based method could also be referenced in [Čapkun et al., 2002; Moore et al., 2004]. In the former, a local coordinate system is constructed with aid of range of internodes and known anchors positions. Then the direction of the local coordinate systems is adjusted to obtain the same direction for all nodes. Finally, according to the nodes residing along the same direction, the unknown nodes position could be estimated simply as a sum of two vectors like $\vec{i}l = \vec{i}k + \vec{k}l$ as illustrated in Figure I.14. In the

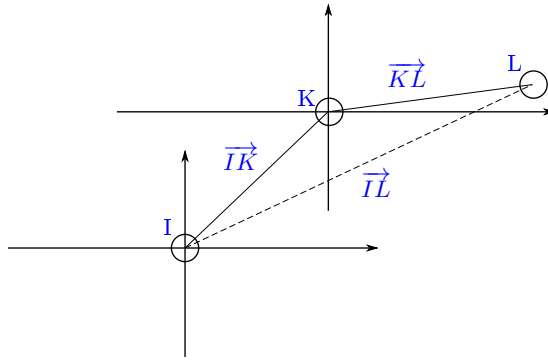


Figure I.14: Illustration of Stitch

latter contribution, the node and its set of neighbors are considered as a cluster. Nodes first localize themselves in the coordinate system of their own cluster under the constraints of properties of the quadrilateral. In this step, they further restrict the quadrilateral to be robust by identifying the triangles satisfying the inequality as Eq. I.11 which bounds the probability of worst-case of a flip error for each triangle.

$$b * \sin^2\theta > d_{min}, \quad (\text{I.11})$$

In the inequality above, b is the shortest side length, θ is the smallest angle and d_{min} is the set threshold. A numerical optimization such as spring relaxation

I.2 Illustration of the localization problem

or Newton-Raphson could be optionally applied whereafter in order to refine the estimates with the full set of distance constraints in terms of reducing and redistributes any accumulated error. Finally, clusters are aligned with help of common nodes existing in two neighboring clusters as well as the transformation operation like rotation, translation and reflection.

- Besides the range-propagated methods cited above, there exist also some other localization approaches based on ranging measurements. Simulated Annealing (SA) [Kannan et al., 2005], utilizing the measured distance of internodes, solves the localization problem by a stochastic optimization approach which is in detail a generalization of Monte Carlo method in combinatorial optimization. Applying this idea to localization issues, the location estimation problem is formulated into an approximated optimization framework as minimizing the cost function as Eq. I.12,

$$CostFunction = \sum_{i=1}^N \sum_{j \in N_i} (\hat{d}_{ij} - \tilde{d}_{ij})^2, \quad (I.12)$$

where N denotes the nodes number, N_i denotes the set of neighbors of nodes i , \tilde{d}_{ij} is the measured distance between nodes i and nodes j estimated in simulation by blurring the true distance by Gaussian noise, \hat{d}_{ij} is the estimated distance obtained by computing Euclidean distance of the estimated coordinates of node i and node j . This cost function reflects practically the quantitative measure of the goodness of the estimated coordinates. The minimization problem of this function is solved by using Simulated Annealing techniques to make out the meaningful estimates. The core of simulated annealing relies in selecting empirically the annealing parameter (temperature T , cooling rate α and perturbation distance Δd) which perturbs a node in a random direction then evaluating the change of cost function by $\Delta CF = CF_{new} - CF_{old}$. According to the value of ΔCF , optional decisions could be made. Another framework utilizing Monte Carlo idea to localize mobile nodes is introduced in a connectivity based way in [Hu and Evans, 2004]. Moreover, the Simulated Annealing localization method is further improved in [Kannan et al., 2006] by a phase of optimization performing on the nodes that are likely to have flip ambiguity problem.

- A bounding box thinking is proposed in [Savvides et al., 2002]. In this contribution, the possible location for a node is constrained in a region with aid of measured distances between node and anchors. Then the initial estimate is decided as the center of the bounding box. The bounding box positioning principle is shown in Figure I.15. As a consequence, the coordinate of node C along x axe, X_C , is assumed to be located in the interval of $[X_A - a, X_B + b + c]$, similarly applied on the y coordinates. Then it only needs to take the center of interval values as nodes position. [Savvides et al., 2002] also refines the initial estimated positions using specifically Kalman Filter instead of the standard least squares methods. This method employs collaborative multihop information combined with the known anchors location to estimate accurately nodes positions.

I. INTRODUCTION

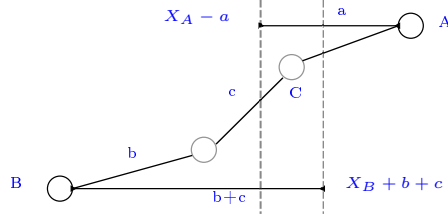


Figure I.15: Bounding Box I

An optional framework is described in [Savvides et al., 2001; Simic and Sastry, 2002]. The principle of this positioning scheme relies in the fact that node resides within the intersection of all its anchors squared bounding boxes. The bounding box of anchor A is centered at the anchors position with height and width $2d$, where d is possibly to be the measured nodes distance or some other representative range value. The final estimate is then taken as the center of the final intersected bounding box as shown in Figure I.16. The bounding box is a rather computation-

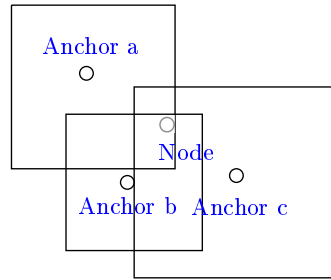


Figure I.16: Bounding Box II

ally saving and simple method. It works best among massive localization method when the sensor network is extremely limited by capacity of computation.

I.2.2.4 Scenario based method

Regarding to the variety of the localization methods discussed above, either connectivity-based or range-based, they all adopt a thinking that infers a node's position straightforward from the measurements $\mathbf{m}_j(t)$, either by hop-count, distance value or other connectivity indicator. In most cases, measurements $\mathbf{m}_j(t)$ are possibly influenced and interrupted by various surrounding factors such as additive noise, obstacles, temperature, pressure, etc. Even the signal itself has some influences like multi-path, channel fading, system clock synchronization, etc.

Cite the case of RSSI in real environment which is usually serving for estimating the distance or other characteristics owing to its simplicity for implements. RSSI is supposed to be a function of the distance separating the internodes. However, an exact model of RSSI as a function of the distance is more complex and uncontrollable in reality thus it is practically not easy to obtain. That is to say, the pathloss model is supposed to

I.2 Illustration of the localization problem

be capable of describing the trend of the relation between distance and signal strength, but not able to reflect an exact degradation in a real environment, due to the factors of additive noise, obstacles, multi-path, channel fading, etc. An example of RSS values with various distances in real case [Ilyas and Mahgoub, 2004] is shown in Figure I.17. The plot shows a slight tremulous curve of variation of RSS not strictly following the logarithmic distribution.

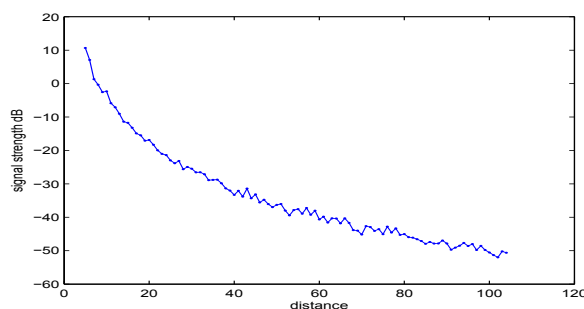


Figure I.17: Received signal strength with increase of distance (figure from [Ilyas and Mahgoub, 2004])

Consequently, there emerged kinds of improvements on the RSSI-based methods. Such works perform improvements by adjusting transmission power, constructing localization error models, generating connectivity measurements or an empirical method in contribution [Jiang et al., 2013]. The combination of the distances constraints could be solved using different approaches, such as Sequential Monte-Carlo algorithm [Hu and Evans, 2004; Baggio and Langendoen, 2008], interval analysis [Mourad et al., 2008, 2009], variational filter [Teng et al., 2010], etc.

Another type of improvement for RSSI-based method takes the scenario characteristics into consideration. It is based on the idea of pattern matching as shown in Figure I.18, which tries to compare the measured signal strength in real time with the training ones recorded in the database in advance and determine the likelihood between them, defined as fingerprinting method [Bahl and Padmanabhan, 2000; Lorincz and Welsh, 2005]. A further improvement method is executed by a full collection of scenario information, which indicates gathering the surrounding significant characteristics, including but not limited to colors, edges, humidity, temperatures, magnetic field [Tapus and Siegart, 2006; Robles et al., 2010; Haverinen and Kemppainen, 2009], as well as RSSI, to model the environment as far as possible. The fingerprinting methods typically consist of two phases, offline phase and online phase. Once a comparison between the data of the two phases is made, the similarity of the measured data with the training positions could be determined by several different methods, such as the nearest neighbor and the K-nearest neighbor algorithms [Bahl and Padmanabhan, 2000; Quan et al., 2010; Chuenurajit et al., 2013], probability-based algorithms [Ilyas and Mahgoub, 2004; Castro et al., 2001], fuzzy logic such as in [Suroso et al., 2011; Liu et al., 2012; Rozyyev et al., 2012] or learning algorithms such as neural networks [Gogolak et al., 2011] or machine

I. INTRODUCTION

learning [Farjow et al., 2011; Mahfouz et al., 2013b,a] that are further introduced in Chapter IV.

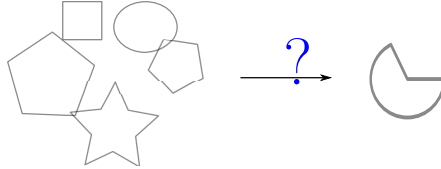


Figure I.18: An example of pattern matching

The fingerprinting methods are very prospective, allowing configurations of complex scenarios with high flexibility in real applications. This method circumambulates the uncertainty of direct conversion of the signal strength to the distance as mentioned before. Instead, it directly employs the values of signal strength which could be considered as modeling of scenario reflecting much more real environment. Therefore, it is also considered as range-free method. Note that, although utilizing connectivity information is more robust than estimating distances using RSSIs, the distribution of anchors and nodes would highly affect the accuracy of the positioning algorithms. Unlike connectivity-based methods, the fingerprinting methods perform well even with low-density anchors in the network.

In order to improve the estimation accuracy, it is practicable to combine different methods, like inertial navigation measurements to improve the precision of localization. In the application of indoor localization, it is possible to use the additional lightweight devices for example accelerometer or gyroscope for localizing and tracking the object moving in indoor environment [Hsu and Yu, 2009] along with the fingerprinting methods as discussed in the following chapters.

I.3 Content Organization

This thesis concentrates on the localization problem settled in an indoor environment. It explores couples of methods around the RSS based localization method, for instance the fingerprinting as mentioned above. It also manages to combine other useful measurements that would help localize an unknown node in the case of movements, for instance the inertial information derived from accelerometer and gyroscope to correct and further refine the measurement precision. With regarding to the solution, it equally introduces the interval analysis technique and Kalman filter with which all the measurements are dealt in respective way to infer an approximated position estimation.

In what follows, at the very beginning, Chapter II introduces a new trend mathematical technique, the interval analysis. This theory is able to include all solutions of sets of inequalities, nonlinear equations even non convex global optimization problem. It is proposed since 1960's and developed until nowadays owning its own journal and regular international conferences. It pursues the guaranteed results depending on rather simple computations which are feasible and applicable in real processing platform. The core

of this technique relies on the concept of an interval $[\mathbf{x}] = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$, in which lower bound and upper bound constrain the possible solution into a box.

Chapter III presents a centralized localization method using radio-fingerprints and acceleration measurements. In this method, all measurements are collected and processed at the central fusion station. The computations lead to two estimates, the first offered by a pattern matching approach based on RSSI and the second offered by the mobility-based 2-nd order integration equation. The solution is given using the interval analysis technique. This framework takes all the measurements uncertainties into consideration leading to an interval box of location estimates.

In Chapter IV, the localization algorithm is extended to the clusterized scheme. Indeed, clusterized algorithms are more feasible and flexible in a large scale sensor network offering much more opportunity in a variety of applications. In this case, the whole surveillance area is divided into several small areas on request where a calculator is settled separately serving as individual central station. The localization is then executed at the relative individual stations at each small area. The computation at these small areas is similar to that of the centralized algorithm mentioned in Chapter III. The solution is given using either interval analysis or Kalman filter. Moreover, the attitude of node is possibly to change such as in the case of pitch, yaw and roll directions. This chapter makes an additional extension to the primary movement model by considering the rotation of nodes, equipped with gyroscopes in addition to their accelerometers.

In view of current trend and application requests nowadays, Chapter V proposes a localization method based on scenario of zoning applying on the area addressing localization. Instead of locating a node at an exact point, it addresses the indicator of room or part of the area where the node actually resides. This method is nominated as *zonage* in french or zoning as normally speaking. In this part, several zoning attempts are illustrated and compared with some fashionable classification methods since categorizing a node to a corresponding zone is possibly considered as classification initiatives. This part employs also interval technique with mobility measurements to help deciding the probability of an area where a node drops in as well as testing with real data.

Chapter VI finally makes a conclusion and describes the perspectives. This thesis discusses a hybrid localization method making use of measured RSSI and mobility information. Interval technique and Kalman filter are both introduced to solve the localization problem from different views. This work is not an ending but many perspective facts remain to be considered and improved. Especially in real cases, the online measurements are empirically blurring due to the surrounding factors. It always remains a great challenge for the researchers in this domain to estimate correctly the real locations with minimal errors.

REFERENCES

References

- I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002. [2](#), [3](#), [4](#)
- J. Bachrach, R. Nagpal, M. Salib, and H. Shrobe. Experimental results for and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks. *Telecommunication Systems*, 26(2-4):213–233, 2004. [12](#)
- A. Baggio and K. Langendoen. Monte carlo localization for mobile wireless sensor networks. *Ad Hoc Networks*, 6(5):718–733, 2008. [21](#)
- P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. IEEE, 2000. [9](#), [21](#)
- S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1713–1723. IEEE, 2003. [3](#)
- A. Bharathidasan and V. A. S. Ponduru. Sensor networks: An overview. *Department of Computer Science. University of California*, 2002. [2](#)
- E. Biagioni. Pods: Interpreting spatial and temporal environmental information. In *In Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents, and Virtual Reality, Volume I of the Proceedings of HCI International 2001, the 9th International Conference on Human-Computer Interaction*. Citeseer, 2001. [5](#)
- E. S. Biagioni and K. Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16(3):315–324, 2002. [5](#)
- P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004. [17](#)
- S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994. [14](#)
- N. Bulusu, V. Bychkovskiy, D. Estrin, and J. Heidemann. Scalable, ad hoc deployable, rf-based localization. In *Proceedings of the Grace Hopper Conference on Celebration of Women in Computing*, volume 31, 2002. [12](#)
- N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, 7(5):28–34, 2000. [12](#)
- N. Bulusu, J. Heidemann, and D. Estrin. Adaptive beacon placement. In *Distributed Computing Systems, 2001. 21st International Conference on.*, pages 489–498. IEEE, 2001. [12](#)
- S. Čapkun, M. Hamdi, and J.-P. Hubaux. Gps-free positioning in mobile ad hoc networks. *Cluster Computing*, 5(2):157–167, 2002. [18](#)

- P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A probabilistic room location service for wireless networked environments. In *UbiComp 2001: Ubiquitous Computing*, pages 18–34. Springer, 2001. 21
- A. Chehri, P. Fortier, and P. M. Tardif. Uwb-based sensor networks for localization in mining environments. *Ad Hoc Networks*, 7(5):987–1000, 2009. 4
- P. Cheong, K.-F. Chang, Y.-H. Lai, S.-K. Ho, I.-K. Sou, and K.-W. Tam. A zigbee-based wireless sensor network node for ultraviolet detection of flame. *Industrial Electronics, IEEE Transactions on*, 58(11):5271–5277, 2011. 4
- B.-S. Choi, J.-W. Lee, J.-J. Lee, and K.-T. Park. A hierarchical algorithm for indoor mobile robot localization using rfid sensor fusion. *Industrial Electronics, IEEE Transactions on*, 58(6):2226–2235, 2011. 4
- T. Chuenurajit, S. Phimmasean, and P. Cherntanomwong. Robustness of 3d indoor localization based on fingerprint technique in wireless sensor networks. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, pages 1–6. IEEE, 2013. 21
- R. Dagher, N. Mitton, I. Amadou, et al. Towards wsn-aided navigation for vehicles in smart cities: An application case study. In *1st International IEEE Percom Workshop on Pervasive Systems for Smart Cities (PerCity 2014)*, 2014. 6
- B. Denis, J.-B. Pierrot, and C. Abou-Rjeily. Joint distributed synchronization and positioning in uwb ad hoc networks using toa. *Microwave Theory and Techniques, IEEE Transactions on*, 54(4):1896–1911, 2006. 10
- L. Doherty, L. El Ghaoui, et al. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663. IEEE, 2001. 14
- N. Erasala and D. C. Yen. Bluetooth technology: a strategic analysis of its role in global 3g wireless communication era. *Computer Standards & Interfaces*, 24(3):193–206, 2002. 4
- W. Farjow, A. Chehri, M. Hussein, and X. Fernando. Support vector machines for indoor sensor localization. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 779–783. IEEE, 2011. 22
- B. Gao and S. Guo. Development of an infrared sensor-based wireless intelligent fish-like underwater microrobot. In *Information and Automation (ICIA), 2010 IEEE International Conference on*, pages 1314–1318. IEEE, 2010. 4
- L. Gogolak, S. Pletl, and D. Kukolj. Indoor fingerprint localization in wsn environment based on neural network. In *Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on*, pages 293–296. IEEE, 2011. 21
- I. Guvenc and C.-C. Chong. A survey on toa based wireless localization and nlos mitigation techniques. *Communications Surveys & Tutorials, IEEE*, 11(3):107–124, 2009. 9
- A. Hande, T. Polk, W. Walker, and D. Bhatia. Self-powered wireless sensor networks for remote patient monitoring in hospitals. *Sensors*, 6(9):1102–1117, 2006. 6

REFERENCES

- J. Haverinen and A. Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10):1028–1035, 2009. [21](#)
- T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95. ACM, 2003. [12](#)
- J. L. Hill. *System architecture for wireless sensor networks*. PhD thesis, University of California, Berkeley, 2003. [5](#)
- B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer, September 2004. ISBN 3211835342. [7](#)
- C.-H. Hsu and C.-H. Yu. An accelerometer based approach for indoor localization. In *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on*, pages 223–227. IEEE, 2009. [22](#)
- L. Hu and D. 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](#), [21](#)
- M. Ilyas and I. Mahgoub. *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press, 2004. [xiii](#), [2](#), [21](#)
- X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2652–2661. IEEE, 2004. [17](#)
- J.-A. Jiang, X.-Y. Zheng, Y.-F. Chen, C.-H. Wang, P.-T. Chen, C.-L. Chuang, and C.-P. Chen. A distributed rss-based localization using a dynamic circle expanding mechanism. *Sensors Journal, IEEE*, 13(10):3754–3766, 2013. [21](#)
- P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebnet. In *ACM Sigplan Notices*, volume 37, pages 96–107. ACM, 2002. [5](#)
- K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1012–1022. IEEE, 2004. [9](#), [11](#)
- A. A. Kannan, G. Mao, and B. Vucetic. Simulated annealing based localization in wireless sensor network. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 2–pp. IEEE, 2005. [19](#)
- A. A. Kannan, G. Mao, and B. Vucetic. Simulated annealing based wireless sensor network localization with flip ambiguity mitigation. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 2, pages 1022–1026. IEEE, 2006. [19](#)
- D. Koks. Numerical calculations for passive geolocation scenarios. Technical report, DTIC Document, 2007. [10](#)

-
- L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 64–75. ACM, 2005. 6
- A. Kulaib, R. Shubair, M. Al-Qutayri, and J. W. Ng. An overview of localization techniques for wireless sensor networks. In *Innovations in Information Technology (IIT), 2011 International Conference on*, pages 167–172. IEEE, 2011. 10
- L. Li, H. Xiaoguang, C. Ke, and H. Ketai. The applications of wifi-based wireless sensor network in internet of things and smart grid. In *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, pages 789–793. IEEE, 2011. 4
- T.-C. Liang, T.-C. Wang, and Y. Ye. A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization. *Sanford University, formal report*, 5, 2004. 17
- D. Liu, D. Tang, Z. Xu, Y. Hu, and C. Huang. The research of the fuzzy cluster algorithm for indoor location based on rssi. In *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pages 5–7. IEEE, 2012. 21
- K. Lorincz and M. Welsh. Motetrack: A robust, decentralized approach to rf-based location tracking. In *Location-and Context-Awareness*, pages 63–82. Springer, 2005. 21
- S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi. Decentralized localization using fingerprinting and kernel methods in wireless sensor networks. In *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*, pages 1–5, Sept 2013a. 22
- S. Mahfouz, F. Mourad-Chehade, P. Honeine, H. Snoussi, and J. Farah. Kernel-based localization using fingerprinting in wireless sensor networks. In *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on*, pages 744–748. IEEE, 2013b. 22
- A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002. 5
- L. Meertens and S. Fitzpatrick. The distributed construction of a global coordinate system in a network of static computational nodes from inter-node distances. *Kestrel Institute TR KES. U*, 4, 2004. 18
- D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61. ACM, 2004. 18
- F. Mourad. *Auto-localisation et suivi de cibles dans les réseaux de capteurs mobiles*. PhD thesis, Université de Technologie de Troyes, Décembre 2010. 2
- F. Mourad, H. Snoussi, F. Abdallah, and C. Richard. Guaranteed boxed localization in manets by interval analysis and constraints propagation techniques. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5. IEEE, 2008. 21

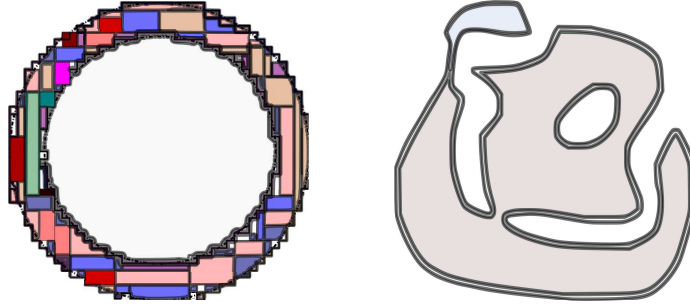
REFERENCES

- F. Mourad, H. Snoussi, F. Abdallah, and C. Richard. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *Signal Processing, IEEE Transactions on*, 57(8): 3226–3239, 2009. [21](#)
- R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Information Processing in Sensor Networks*, pages 333–348. Springer, 2003. [14](#)
- D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2926–2931. IEEE, 2001. [13](#), [16](#)
- D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1734–1743. IEEE, 2003. [9](#)
- A. Pal. Localization algorithms in wireless sensor networks: Current approaches and future challenges. *Network Protocols and Algorithms*, 2(1):45–73, 2010. [9](#), [11](#)
- N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000. [9](#)
- N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 1–14. ACM, 2001. [9](#)
- M. Quan, E. Navarro, and B. Peuker. Wi-fi localization using rssi fingerprinting. 2010. [21](#)
- C. S. J. Rabaey and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX technical annual conference*, 2002. [17](#)
- J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr, D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 33(7):42–48, 2000. [6](#)
- T. S. Rappaport, J. Reed, and B. D. Woerner. Position location using wireless communications on highways of the future. *Communications Magazine, IEEE*, 34(10):33–41, 1996. [10](#)
- J. J. Robles, M. Deicke, and R. Lehnert. 3d fingerprint-based localization for wireless sensor networks. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, pages 77–85. IEEE, 2010. [21](#)
- A. Rozyyev, H. Hasbullah, and F. Subhan. Combined k-nearest neighbors and fuzzy logic indoor localization technique for wireless sensor network. *Research Journal of Information Technology*, 4(4):155–165, 2012. [21](#)
- A. Runge, M. Baunach, and R. Kolla. Precise self-calibration of ultrasound based indoor localization systems. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–8. IEEE, 2011. [4](#)
- A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001. [9](#), [20](#)

-
- A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 112–121. ACM, 2002. [12](#), [17](#), [19](#)
- L. Schwiebert, S. K. Gupta, and J. Weinmann. Research challenges in wireless networks of biomedical sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 151–165. ACM, 2001. [6](#)
- B. Shahid, A. A. Kannan, N. H. Lovell, and S. Redmond. Ultrasound user-identification for wireless sensor networks. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 5756–5759. IEEE, 2010. [4](#)
- Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz. Localization from connectivity in sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 15(11):961–974, 2004. [16](#)
- Y. Shang and W. Ruml. Improved mds-based localization. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2640–2651. IEEE, 2004. [16](#)
- Y. Shang, W. Ruml, Y. Zhang, and M. P. Fromherz. Localization from mere connectivity. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 201–212. ACM, 2003. [15](#)
- S. N. Simic and S. Sastry. Distributed localization in wireless ad hoc networks. Technical report, Technical Report UCB/ERL, 2002. [20](#)
- G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 1–12. ACM, 2004. [5](#)
- I. Stojmenovic. *Handbook of sensor networks: algorithms and architectures*, volume 49. John Wiley & Sons, 2005. [2](#), [10](#)
- D. J. Suroso, P. Cherntanomwong, P. Sooraksa, and J.-i. Takada. Fingerprint-based technique for indoor localization in wireless sensor networks using fuzzy c-means clustering algorithm. In *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*, pages 1–5. IEEE, 2011. [21](#)
- A. Tapus and R. Siegwart. A cognitive modeling of space using fingerprints of places for mobile robot navigation. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1188–1193. IEEE, 2006. [21](#)
- J. Teng, H. Snoussi, and C. Richard. Decentralized variational filtering for target tracking in binary sensor networks. *Mobile Computing, IEEE Transactions on*, 9(10):1465–1477, 2010. [21](#)
- G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, et al. A macroscope in the redwoods. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 51–63. ACM, 2005. [5](#)

REFERENCES

- I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 154–165. ACM, 2005. [5](#)
- F. Viani, P. Rocca, L. Lizzi, M. Rocca, G. Benedetti, and A. Massa. Wsn-based early alert system for preventing wildlife-vehicle collisions in alps regions. In *Antennas and Propagation in Wireless Communications (APWC), 2011 IEEE-APS Topical Conference on*, pages 106–109. IEEE, 2011. [6](#)
- J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008. [4](#), [7](#)
- H. Zhang and P. N. Pathirana. Uplink power control via adaptive hidden-markov-model-based pathloss estimation. *Mobile Computing, IEEE Transactions on*, 12(4):657–665, 2013. [3](#)
- P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 227–238. ACM, 2004. [5](#)
- F. Zhao and L. J. Guibas. *Wireless sensor networks: an information processing approach*. Morgan Kaufmann, 2004. [2](#)



Contents

- II.1 Introduction
 - II.2 Definition of intervals
 - II.3 Calculus of interval analysis
 - II.4 Conclusion
 - References
-

In a majority of engineering applications, the nonlinear problem is frequently encountered and addressed in reality however looking forward to a robust solution. Such kinds of mathematical problems are normally represented in terms of solving a set of non-linear equations, non-linear inequalities, or a global minimization for a non-convex cost function. Interval analysis provides a framework for solving such problems in a guaranteed way. The basic idea lies in enclosing all the possible scalar or vector solutions of the problem, leading to intervals or pavings/boxes, ensuring guaranteed results unlike some classical numerical approaches, even with strong nonlinearities and discontinuities. Instead of performing an inner approximation of the solution by computing single estimates, interval analysis consists of performing an outer approximation leading to paving boxes. The solution boxes ideally and consequently include the exact solutions of the problem, and also all computation and measurement incertitude. Not to mention that it is extremely simple and easy to implement in on-board equipments such as distributed wireless sensor networks. This chapter gives a general view of interval analysis, including the basic operations, functions and functional solvers.

II. INTERVAL ANALYSIS

II.1 Introduction

For various reasons, interval analysis have been with limited impact in engineering applications until early 90's last century. Since then, it has turned back to the acting stage definitively. Interestingly, issuing from 1950s and 1960s, the notion of interval was once motivated by the procedure of calculating $\sqrt{2}$ [Sunaga 1958; 2009]. It is thus recommended to firstly glance at the demonstration of procedure of square rooting of numeral 2 as in Figure II.1.

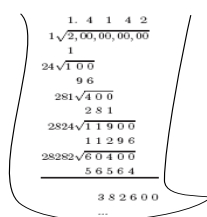


Figure II.1: Procedure of calculating $\sqrt{2}$

As inferred in that contribution [Sunaga 1958; 2009], firstly it is calculated up to numeral 1, as 1 is the maximum real number whose square value is inferior than object 2. Then it is proceeded to sign at the follow decimal places as 1.4, 1.41, 1.414 etc. The author questioned the meanings of this sequence of numerals and proposed that numeral 1 does not simply denote a rational number, moreover it implies that, in the consequence steps, one value from 0 to 9 is definitively picked up as the following decimal place under the decimal numeral system. Namely, the obtained numeral 1 implies in fact an interval counting initially from 1 and up to next integer numeral 2, concluded as the interval [1, 2] containing all numbers from 1 to 2. By such analogy, renewed 1.4 denotes interval [1.4, 1.5], 1.41 denotes [1.41, 1.42], and 1.414 denotes [1.414, 1.415] etc. This procedure of calculating $\sqrt{2}$ gives a hint that the inference of $\sqrt{2}$ is acquired practically under the assumption of series of intervals.

Thereout, as concluded, the concept of an interval is considered more fundamental and illustratable than that of a real number. Furtherer, such kind of interval ensemble is possibly to be derived simply from only one rational number rather than two rational numbers, such as the numeral 1.414 which actually signifies an interval, for instance [1.4135, 1.4145] by rounding off. Therefore, what do all of this inference mean? To describe a physical quantity, the representation of an interval is more suitable than a concrete real number. This transition also allows a certain breadth for the physical quantities description. The interval concept is also considered as an applicable tool which is capable to link the pure mathematics with reality, as well as pure analysis with applied analysis.

In consequence, interval analysis arose and promoted at historic moments [Sunaga 1958; Moore 1966; Kearfott et Kreinovich 1996; Jaulin *et al.* 2001]. This analysis is an approach of mathematical computation aiming at operating sets, instead of numerical numbers. It is attached to solve the unsolvable problems by many normal ways, such

as solving a set of nonlinear equations [Neumaier 1990], a set of nonlinear inequalities [Moore 1992; Jaulin et Walter 1993; Walter et Jaulin 1994] as well as global optimization problems for non-convex cost functions [Hansen et Walster 2003] as shown in Figure II.2. It provides then guaranteed conclusions enclosing all the solutions of the addressed problem, even works with strong nonlinearities or discontinuous sets. All of the guaranteed operations rely on the robust properties of interval or paving(boxes) which will be introduced in the following sections.

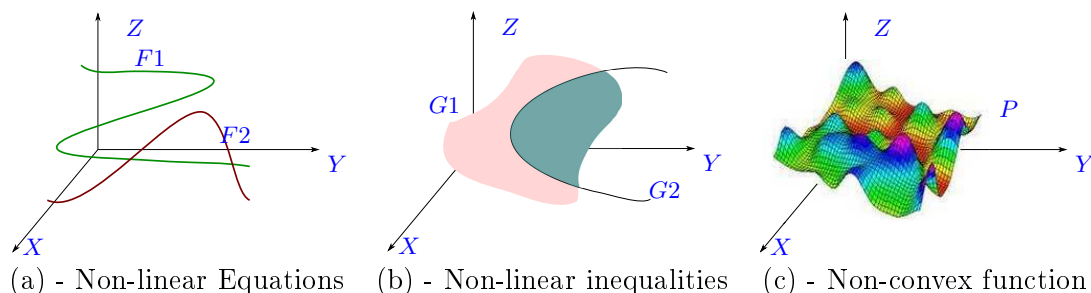


Figure II.2: Problems attached. (a) demonstrates the case of two curves with crossing points, (b) demonstrates the case of a curved plan cut by a curve and (c) demonstrates the case of non-convex surface.

II.2 Definition of intervals

The set of all variables ξ satisfying the condition of $\alpha \leq \xi \leq \beta$ is called an interval. It is denoted by $[\alpha, \beta]$. An example is shown in Figure II.1 below.

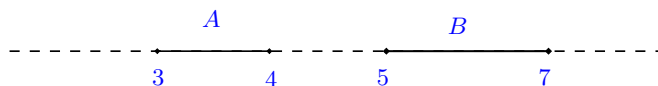


Figure II.3: Two examples of interval

Interval A is denoted by $[3, 4]$, also interval B is denoted by $[5, 7]$. Generally, in a real space \mathbb{R} , a real interval $[x]$ is a connected subset included in \mathbb{R} , defined as in Eq. II.1 below,

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}, \quad (\text{II.1})$$

where \underline{x} and \bar{x} denotes respectively the lower bound $\text{inf}[x]$ and the upper bound $\text{sup}[x]$ of interval $[x]$. Both of the bounds could be finite or infinite. An empty set \emptyset is considered also as an interval if indicator \emptyset represents the absence of solution for certain problem. Note that, the ensemble of all the real interval is denoted by \mathbb{IR} . There are couples of terms very useful in applied cases such as *width* and *center*. The width of an interval $[x] \neq \emptyset$, one of the distinctions of $[x]$, is defined as in Eq. II.2,

$$w([x]) = \bar{x} - \underline{x}. \quad (\text{II.2})$$

II. INTERVAL ANALYSIS

$w([x])$ owns equally the attribute of denoting the size of interval $[x]$ which is determined by the uncertainty of the quantity x . The interval is degraded as null if its width is equal to zero, consequently denoted and identified by real number x . Another distinction of $[x]$ is its center defined as in Eq. II.3,

$$c([x]) = \frac{x + \bar{x}}{2}. \quad (\text{II.3})$$

Based on these notations, a very useful formula is introduced to present the interval $[x]$ in another modality. In fact, any interval $[x]$ is able to be transformed as the expression of Eq. II.4,

$$[x] = c([x]) + [-\frac{1}{2} \cdot w([x]), \frac{1}{2} \cdot w([x])] = c([x]) + \frac{1}{2} \cdot w([x]) \cdot [-1, 1]. \quad (\text{II.4})$$

This equation is quite handy and applicable in need of using interval analysis which will be demonstrated in the following.

The concept of interval in one dimensional space as in Figure II.1 is further required to be extended into multidimensional case. In a two dimensional case, a planar box $[\mathbf{x}]$ is constructed by aligning two intervals $[x_1]$ at axis x_1 and $[x_2]$ at axis x_2 paralleled as in Figure II.4. This could be equalized as $[\mathbf{x}] = [x_1] \times [x_2]$. Generatively, an interval vector

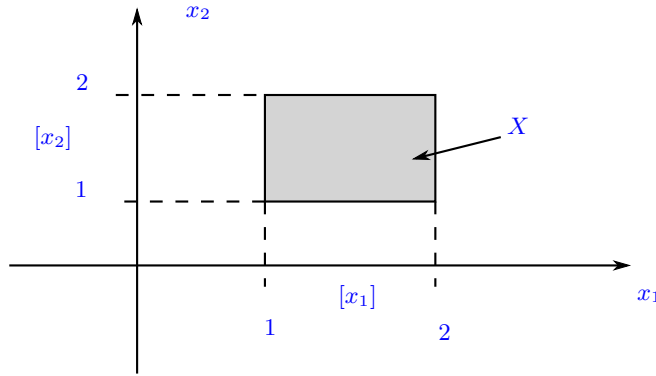


Figure II.4: An example of 2D interval

$[\mathbf{x}]$ denoting a subset of \mathbb{R}^n can be derived by the cartesian product of n intervals $[x_i]$ which is defined as closed. The n -dimensional interval or vector $[\mathbf{x}]$, also called paving or box, is written as in Eq. II.5 below,

$$[\mathbf{x}] = [x_1] \times [x_2] \times \dots \times [x_n], \quad i = 1, \dots, n, \quad (\text{II.5})$$

where $[x_i] = [x_i, \bar{x}_i]$ denotes the i -th interval component projected onto the i th axis of vector $[\mathbf{x}]$. Note that an empty set of \mathbb{R}^n is denoted as $\emptyset_1 \times \emptyset_2 \times \dots \times \emptyset_n$ if and only if all of the interval components are empty. This assumption guarantees the unique notation

of a given box. Extensively, the lower and upper bounds of a box $[\mathbf{x}]$ are both bound vectors composed of all the components' lower or upper bounds as in Eq. II.6,

$$\begin{aligned}\underline{\mathbf{x}} &= (x_1, x_2, \dots, x_n)^T, \\ \overline{\mathbf{x}} &= (\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)^T.\end{aligned}\tag{II.6}$$

Similarly, the width of $[\mathbf{x}]$ is defined as the maximum width existing in the set of all the components, as in Eq. II.7,

$$w([\mathbf{x}]) = \max_{1 \leq i \leq n} w([x_i]),\tag{II.7}$$

whereas the center of a box is taken as the vector center stretched along each projected axis as in Eq. II.8,

$$c([\mathbf{x}]) = (c([x_1]), c([x_2]), \dots, c([x_n]))^T.\tag{II.8}$$

II.3 Calculus of interval analysis

Once the concept of interval is defined, it is feasible to apply the usual operation rules of sets to the interval framework, since interval analysis theory is founded on the set theory and its computation is in fact a special case of computation on sets. Based on this attribute, kinds of useful algorithms are developed and defined as consequence. In this section, basic operations are firstly put into discussion to get a prior knowledge, then they are followed by the inclusion function and subpaving which are deemed to be used in interval analysis algorithm SIVIA and the determination of contractors.

II.3.1 Basic tools

On the basis of the discussion of preceding sections, the real interval virtually possesses double natures, represented as real number as well as set, which benefit eventually in applying the normal set and number operations on interval space. As a consequence, it is possible to define the concrete operation rules in interval framework [Moore et Moore 1979; Kearfott et Kreinovich 1996; Jaulin *et al.* 2001]. On account of this nature, basic interval operations are divided into two ways, one based on set/ensemble operations and another based on arithmetical operations.

II.3.1.1 Set/ensemble operations

Similar with the set theory, the basic set operations are equally suitable to an interval $[x]$ of \mathbb{R} . For two real intervals $[x]$ and $[y]$ of \mathbb{R} , if the condition of $\underline{x} \geq \underline{y}$ and $\overline{x} \leq \overline{y}$ is satisfied, it is naturally concluded the inclusion relationship of $[x]$ and $[y]$ as in Eq. II.9,

$$[x] \subseteq [y], \text{ if } \forall x \in [x], x \in [y].\tag{II.9}$$

The intersection of $[x]$ and $[y]$ is defined as in Eq. II.10 which is also an interval,

$$[x] \cap [y] = \begin{cases} \emptyset, & \text{if } \overline{x} < \underline{y} \text{ or } \overline{y} < \underline{x}, \\ [\max\{\underline{x}, \underline{y}\}, \min\{\overline{x}, \overline{y}\}], & \text{otherwise.} \end{cases}\tag{II.10}$$

II. INTERVAL ANALYSIS

The interval union of $[x]$ and $[y]$ is nevertheless an exception, which is not the case of their union as $[x] \cup [y] = \{z \in [x] \text{ or } z \in [y]\}$. Indeed, $[x] \cup [y]$ is not necessarily a connected interval. As consequence, a new interval union of $[x]$ and $[y]$ is defined as $[x] \sqcup [y]$ and expressed as in Eq. II.11,

$$[x] \sqcup [y] = [\min\{\underline{x}, \underline{y}\}, \max\{\bar{x}, \bar{y}\}]. \quad (\text{II.11})$$

The relationship of $[x]$ deprived of $[y]$ is defined as $[x] \setminus [y] = [\{x \in [x] \mid x \notin [y]\}]$. Alternatively it is expressed also as in Eq. II.12,

$$[x] \setminus [y] = \begin{cases} \emptyset, & \text{if } [x] \subseteq [y], \\ [\underline{x}, \underline{y}], & \text{if } \underline{x} \leq \underline{y} \leq \bar{x} \leq \bar{y}, \\ [\bar{y}, \bar{x}], & \text{if } \bar{y} \leq \bar{x} \leq \underline{y} < \underline{x}, \\ [x], & \text{else} \end{cases} \quad (\text{II.12})$$

Example 1 *The ensemble operations of intervals are illustrated as follows:*

$$\begin{aligned} [1, 6] \cap [4, 9] &= [4, 6], \\ [-1, 2] \sqcup [4, 5] &= [-1, 5], \\ [1, 4] \setminus [3, 5] &= [1, 3]. \end{aligned}$$

II.3.1.2 Arithmetic operations

The real interval is actually composed of real numbers. The double natures of real number and set also permit to apply rules of arithmetics on intervals. There exist four fundamental operators such as addition (+), subtraction (-), multiplication (*) and division (/) possibly extended to intervals. For generalizing, let symbol \diamond denote any binary operator among (+, -, *, /) bridging $[x]$ and $[y]$ as in Eq. II.13,

$$[x] \diamond [y] = [\{x \diamond y \in \mathbb{R} \mid x \in [x], y \in [y]\}], \text{ with } \diamond \in \{+, -, *, /\}. \quad (\text{II.13})$$

More specifically, the associated operation is performed using the real numbers, namely the lower bounds and upper bounds of individual intervals, defined as in Eq. II.14,

$$[x] \diamond [y] = [\min\{\underline{x} \diamond \underline{y}, \underline{x} \diamond \bar{y}, \bar{x} \diamond \underline{y}, \bar{x} \diamond \bar{y}\}, \max\{\underline{x} \diamond \underline{y}, \underline{x} \diamond \bar{y}, \bar{x} \diamond \underline{y}, \bar{x} \diamond \bar{y}\}]. \quad (\text{II.14})$$

Note that this definition is true for division in only cases where $[y]$ does not include 0. Otherwise, one or the two bounds of the resulting interval would be infinite. Eq. II.14 is possibly simplified in further for the addition and subtraction operations as in Eq. II.15,

$$\begin{aligned} [x] + [y] &= [\underline{x} + \underline{y}, \bar{x} + \bar{y}], \\ [x] - [y] &= [\underline{x} - \bar{y}, \bar{x} - \underline{y}]. \end{aligned} \quad (\text{II.15})$$

Additionally, the operations of negation (-) and square (²) are also applicable to interval computations as in Eq. II.16,

$$\begin{aligned} -[x] &= [\{-x \mid x \in [x]\}] = [-\bar{x}, -\underline{x}], \\ [x]^k &= \{x^k \mid x \in [x]\}, \quad k \in \{1, 2, \dots\}. \end{aligned} \quad (\text{II.16})$$

As for the operations of multiplication and division, under the condition of both bounds are absolutely positive, the relative interval computations is possible to be included as in Eq. II.17,

$$\begin{aligned} [x] * [y] &= [\underline{x} * \underline{y}, \bar{x} * \bar{y}], \\ [x]/[y] &= [\underline{x}/\bar{y}, \bar{x}/\underline{y}], \\ [x]^2 &= [\underline{x}^2, \bar{x}^2]. \end{aligned} \tag{II.17}$$

Note that, in Eq. II.16, $[x]^2 \neq [x] * [x]$. For instance, $[-1, 2]^2$ results in the interval $[0, 4]$ rather than $[-1, 2] * [-1, 2] = [-2, 4]$ that should be distinguished in computations applied on intervals.

Example 2 *The arithmetics of intervals are illustrated as follows:*

$$\begin{aligned} [1, 2] + [4, 5] &= [5, 7], \\ [-1, 5] - [1, 3] &= [-4, 4], \\ -[-5, 6] &= [-6, 5], \\ [-1, 3] * [-2, 5] &= [-6, 15], \\ [-2, 4]/[-1, 2] &=]-\infty, +\infty[, \text{ since } 0 \text{ is included in } [-1, 2], \\ [1, 2]^2 &= [1, 4]. \end{aligned}$$

II.3.1.3 Operations extension to interval vectors

Since an interval vector $[\mathbf{x}] \in \mathbb{IR}^n$ is derived from the cartesian product of n one-dimensional intervals $[x_i]$, $i \in 1, 2, \dots, n$ as in Eq. II.5, the basic operations of one-dimensional interval $[x]$ is equally applicable to each dimension modifying the corresponding multidimensional interval $[\mathbf{x}]$. Accordingly, the basic notions on sets and arithmetics for ordinary intervals are extended into interval vectors as follows.

In the first place, a vectorial set/ensemble, denoting a paving or a box, $[\mathbf{x}] \in \mathbb{IR}^n$ is considered to be included in another vectorial set/ensemble $[\mathbf{y}] \in \mathbb{IR}^n$ if all of the components in $[\mathbf{x}]$ are included in associated components of $[\mathbf{y}]$, as in Eq. II.18,

$$[\mathbf{x}] \subset [\mathbf{y}], \text{ if } \forall i, [x_i] \subset [y_i], 1 \leq i \leq n. \tag{II.18}$$

Intersection of $[\mathbf{x}]$ and $[\mathbf{y}]$ of \mathbb{IR}^n is also a vector of \mathbb{IR}^n defined as in Eq. II.19,

$$[\mathbf{x}] \cap [\mathbf{y}] = \begin{cases} \emptyset^n, & \text{if } \exists i, 1 \leq i \leq n, [x_i] \cap [y_i] = \emptyset, \\ ([x_1] \cap [y_1]) \times ([x_2] \cap [y_2]) \times \dots \times ([x_n] \cap [y_n]), & \text{otherwise.} \end{cases} \tag{II.19}$$

For example, the intersection of $[-1, 2] \times [2, 3]$ and $[1, 5] \times [6, 9]$ leads to $[1, 2] \times \emptyset$. This result could be considered correct but inconsistent as the resulted interval $[1, 2]$ is supposed to be the projection onto the first axis. It is also possible to consider that the intersection interval vector is \emptyset^n since the second component is empty.

II. INTERVAL ANALYSIS

The union of $[\mathbf{x}]$ and $[\mathbf{y}]$ of $\mathbb{I}\mathbb{R}^n$ is the cartesian product of all unions \sqcup of all components of intervals $[\mathbf{x}]$ and $[\mathbf{y}]$, denoted as in Eq. II.20,

$$[\mathbf{x}] \sqcup [\mathbf{y}] = ([x_1] \sqcup [y_1]) \times ([x_2] \sqcup [y_2]) \times \dots \times ([x_n] \sqcup [y_n]). \quad (\text{II.20})$$

The vector $[\mathbf{x}]$ deprived of $[\mathbf{y}]$ is also a vector denoted by $[\mathbf{z}] = [\mathbf{x}] \setminus [\mathbf{y}]$, whose components $[z_i]$, $1 \leq i \leq n$, are further illustrated in condition as shown in Eq. II.21,

$$[z_i] = \begin{cases} [x_i] \setminus [y_i], & \text{if } \forall j \neq i, 1 \leq j \leq n, [x_j] \subseteq [y_j], \\ [x_i], & \text{otherwise.} \end{cases} \quad (\text{II.21})$$

Example 3 *The set operations applied on interval vectors are illustrated as follows:*

$$\begin{aligned} [1, 3] \times [2, 5] \cap [2, 4] \times [-1, 4] &= [2, 3] \times [2, 4], \\ ([-1, 1] \times [3, 5] \times [2, 4]) \sqcup ([1, 2] \times [4, 7] \times [0, 1]) &= [-1, 2] \times [3, 7] \times [0, 4], \\ ([1, 5] \times [1, 4]) \setminus ([2, 5] \times [0, 6]) &= [1, 2] \times [1, 4]. \end{aligned}$$

Another extension to the interval vectors is the application of classical arithmetic operation equally due to the double natures as mentioned above. The notions of arithmetical operations (+, -, *, /, ^) are all considered in terms of a symbol of binary operator \diamond for the interval vector computations as in Eq. II.22.

$$[\mathbf{x}] \diamond [\mathbf{y}] = ([x_1] \diamond [y_1]) \times ([x_2] \diamond [y_2]) \times \dots \times ([x_n] \diamond [y_n]). \quad (\text{II.22})$$

Additionally, some classical algebraic properties on arithmetics such as commutativity and associativity applied on addition and multiplication are equally suitable for the interval vectors, such as in Eq. II.23,

$$\begin{aligned} [\mathbf{x}] + [\mathbf{y}] &= [\mathbf{y}] + [\mathbf{x}], \\ [\mathbf{x}] * [\mathbf{y}] &= [\mathbf{y}] * [\mathbf{x}], \\ \alpha \cdot [\mathbf{x}] &= (\alpha \cdot [x_1]) \times (\alpha \cdot [x_2]) \times \dots \times (\alpha \cdot [x_n]), \\ [\mathbf{x}]^T * [\mathbf{y}] &= [x_1] * [y_1] + [x_2] * [y_2] + \dots + [x_n] * [y_n], \end{aligned} \quad (\text{II.23})$$

where $\alpha \in \mathbb{R}$, T denotes the transposition operation. The distributive law of ordinary arithmetics fails to apply. However it is dropped into a sub-distributive law as in Eq. II.24,

$$[x] * ([y] + [z]) \subseteq [x] * [y] + [x] * [z]. \quad (\text{II.24})$$

Moreover, the properties of inclusion of sets operations is combined with the intervals operators with attribute shown as in Eq. II.25,

$$[x_1] \diamond [y_1] \subseteq [x_2] \diamond [y_2], \text{ if } [x_1] \subseteq [x_2] \text{ and } [y_1] \subseteq [y_2]. \quad (\text{II.25})$$

All these properties provide thus the feasibility and the theoretical foundation in applying interval analysis on solving the approximation problem.

II.3.1.4 Inclusion functions

In this section, the interval analysis is further explored leading to the interval function oriented analysis. Consider such a case often encountered in signal processing system, with an existing variable \mathbf{x} which is a vector of \mathbb{R}^n . It is required that variable $\mathbf{x} \in \mathbb{R}^n$ is called for being mapped into another defined space \mathbb{R}^m , leading to a mapping function $f(\mathbf{x})$ associated to the argument \mathbf{x} . This procedure is defined as in Eq. II.26.

$$\mathbf{f} : \mathbf{x} \in \mathbb{R}^n \mapsto \mathbf{f}(\mathbf{x}) \in \mathbb{R}^m. \quad (\text{II.26})$$

If the argument \mathbf{x} is a vector included in an interval vector $[\mathbf{x}] \in \mathbb{IR}^n$, then its function $\mathbf{f}(\mathbf{x})$ turns to be an interval function with argument $[\mathbf{x}]$ defined as in Eq. II.27,

$$\mathbf{f}([\mathbf{x}]) = \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\}. \quad (\text{II.27})$$

Here it comes to the following question: towards the function $\mathbf{f}([\mathbf{x}])$ with interval vector argument $[\mathbf{x}]$, is there any general method to define exactly the coverage area of such a function? The problem here is that $\mathbf{f}([\mathbf{x}])$ is not necessarily an interval. To resolve this query, the concept of inclusion function $[\mathbf{f}](\circ)$ corresponding to original function $\mathbf{f}(\circ)$ is introduced here and defined as in Eq. II.28 satisfying condition shown also in Eq. II.28.

$$\begin{aligned} [\mathbf{f}]([\mathbf{x}]) &= \{\{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\}\}, \text{ satisfying} \\ \forall [\mathbf{x}] \subset \mathbb{IR}^n, \mathbf{f}([\mathbf{x}]) &\subset [\mathbf{f}]([\mathbf{x}]), \end{aligned} \quad (\text{II.28})$$

where $[\mathbf{f}([\mathbf{x}])] \in \mathbb{IR}^m$. A physical signification of inclusion functions with $n = m = 2$ is illustrated explicitly in Figure II.5. The inclusion function $[\mathbf{f}^*]([\mathbf{x}])$ is the minimal

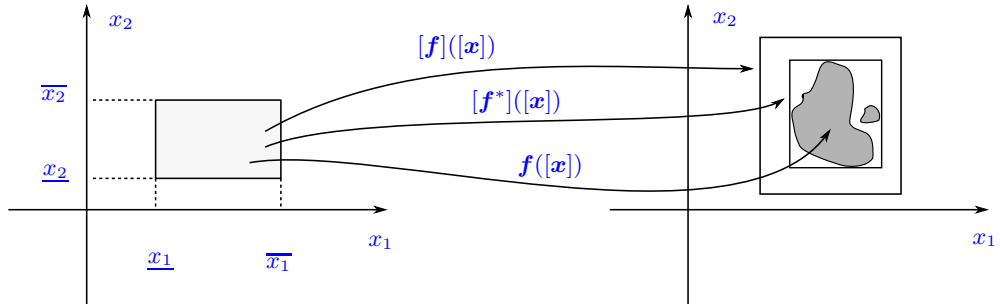


Figure II.5: Illustration of inclusion functions

function if, for any input $[\mathbf{x}]$, $[\mathbf{f}]([\mathbf{x}])$ is the smallest or the most compact box containing $\mathbf{f}([\mathbf{x}])$, uniquely denoted as $[\mathbf{f}^*]([\mathbf{x}])$. According to the illustration above, it is obvious that there exist many possibilities of inclusion function $[\mathbf{f}]([\mathbf{x}])$ since it is possible to be an arbitrary interval paving/box that is able to enclose the original sets of solutions. Concerning this, the selection and construction of inclusion functions will influence the purity of resulted enclosing function. Consider a special case of inclusion function

II. INTERVAL ANALYSIS

$[f]([x]) \in \mathbb{IR}$ with input argument $x \in \mathbb{R}$, the most fitted inclusion function could be simply abstracted by Eq. II.29 as below,

$$[f^*]([x]) = [\min\{f(\underline{x}), f(\bar{x})\}, \max\{f(\underline{x}), f(\bar{x})\}]. \quad (\text{II.29})$$

Additionally, an inclusion function $[f]$ is considered to be inclusion monotonic function if it satisfies the condition as shown in Eq. II.30,

$$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow [f]([\mathbf{x}]) \subset [f]([\mathbf{y}]). \quad (\text{II.30})$$

This definition is true in only cases where the function f owns the monotonicity property, either increasing or decreasing with increasing argument \mathbf{x} . An example could be given by the extensive exponential function expressed in terms of intervals as in Eq. II.31,

$$[\exp^*]([x]) = [\exp(\underline{x}), \exp(\bar{x})]. \quad (\text{II.31})$$

as well as the logarithmic function as in Eq. II.32,

$$[\log^*]([x]) = [\log(\underline{x}), \log(\bar{x})]. \quad (\text{II.32})$$

It is also nevertheless possible to define a restricted non-monotonic function as a renewed monotonic one. For example, the function \sin is restricted as monotonic with input $[x]$ limited in interval $[\frac{-\pi}{2}, \frac{\pi}{2}]$ as shown in Eq. II.33,

$$[\sin^*]([x]) = [\sin(\underline{x}), \sin(\bar{x})]. \quad (\text{II.33})$$

The inclusion vector function \mathbf{f} , defined at the very beginning of this section, mapped from \mathbb{R}^n to \mathbb{R}^m is alternatively expressed by the cartesian product of m inclusion functions $[f_i] \in \mathbb{IR}, i \in \{1, 2, \dots, m\}$ mapped from sets of inputs of \mathbb{IR}^n . Thus an inclusion function \mathbf{f} is given actually as in Eq. II.34,

$$[\mathbf{f}]([\mathbf{x}]) = [f_1]([\mathbf{x}]) \times [f_2]([\mathbf{x}]) \times \dots \times [f_m]([\mathbf{x}]). \quad (\text{II.34})$$

This definition of $[\mathbf{f}]$ hints that the inclusion function $[\mathbf{f}]$ is convergent (thin, minimal, inclusion monotonic), if all its coordinates functions $[f_i]([\mathbf{x}]), i \in \{1, 2, \dots, m\}$ are convergent (thin, minimal, inclusion monotonic) respectively. Moreover, comprehensively, this nature permits and facilitates the interval analysis in multidimensional functional background in many real cases.

Example 4 *This example shows four different formal expressions of the same continuous function $f(x)$,*

$$\begin{aligned} f_1(x) &= x * (x + 1), \\ f_2(x) &= x * x + x, \\ f_3(x) &= x^2 + x, \\ f_4(x) &= (x + \frac{1}{2})^2 - \frac{1}{4}. \end{aligned}$$

The evaluation of their separate inclusion functions with input $[x] = [-1, 1]$ is given in the following and also illustrated in Figure II.6.

$$\begin{aligned} [f_1](x) &= [x] * ([x] + 1) = [-2, 2], \\ [f_2](x) &= [x] * [x] + [x] = [-2, 2], \\ [f_3](x) &= [x]^2 + [x] = [-1, 2], \\ [f_4](x) &= ([x] + \frac{1}{2})^2 - \frac{1}{4} = [-\frac{1}{4}, 2]. \end{aligned}$$

According to the plot, $[f_4]$ is the minimal inclusion function, leading to the smallest interval. However, all other intervals includes the resulting interval of $[f_4]$.

The information reveals from the various results that the inclusion function could be a variety of frames. On the other hand, accuracy of interval computation much relies on the formal expression of function f . Analytically, interval $[x]$ participates only once in the interval equation f_4 , the corresponding inclusion result thus encloses tightly the real solution as $[-\frac{1}{4}, 2]$, illustrated as interval $[f_4]$ in Figure II.6.

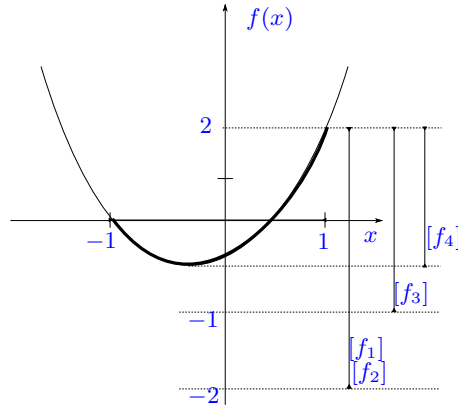


Figure II.6: Variety of formal inclusion functions resulting in different resulting intervals

II.3.1.5 Inclusion tests

The inclusion tests are introduced to figure out whether all elements of a given box $[x]$ satisfy a given property corresponding to the set, or none of them does. Generally, a test leads to a boolean result taken from any value of $\mathbb{B} = \{true, false\}$ or equivalently $\mathbb{B} = \{0, 1\}$. Let \mathbb{IB} be the set of all subsets of \mathbb{B} , that is

$$\mathbb{IB} = \{\emptyset, 0, 1, [0, 1]\}, \tag{II.35}$$

where \emptyset means empty, 0 means false, 1 means true and $[0, 1]$ means indeterminate. Since a test is defined from domain \mathbb{R}^n to \mathbb{B} as a function t , an inclusion test is performed from

II. INTERVAL ANALYSIS

domain \mathbb{IR}^n to \mathbb{IB} as a function $[t]$. For any $[\mathbf{x}] \in \mathbb{IR}^n$, $[t]$ is determined by Eq. II.36.

$$\begin{aligned} ([t]([\mathbf{x}]) = 1) &\Rightarrow (\forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 1), \\ ([t]([\mathbf{x}]) = 0) &\Rightarrow (\forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 0). \end{aligned} \quad (\text{II.36})$$

Example 5 Let the test t for $[\mathbf{x}] = [x_1] \times [x_2]: \mathbb{R}^2 \rightarrow \{0, 1\}$ given by

$$t(\mathbf{x}) = \begin{cases} 1, & \text{if } x_1 - x_2 \geq 0, \\ 0, & \text{if } x_1 - x_2 < 0. \end{cases} \quad (\text{II.37})$$

An inclusion test associated to t is given by

$$[t]([\mathbf{x}]) = \begin{cases} 1, & \text{if } \underline{x}_1 - \overline{x}_2 \geq 0, \\ 0, & \text{if } \overline{x}_1 - \underline{x}_2 < 0, \\ [0, 1], & \text{otherwise.} \end{cases} \quad (\text{II.38})$$

Now let \mathbb{A} be a set of \mathbb{R}^n . The test $t_{\mathbb{A}}(\mathbf{x})$ denotes whether $\mathbf{x} \in \mathbb{A}$ or not. An inclusion test $[t_{\mathbb{A}}]$ associated to $t_{\mathbb{A}}$ is programmed in Eq. II.39.

$$\begin{aligned} ([t_{\mathbb{A}}]([\mathbf{x}]) = 1) &\Leftrightarrow (\forall \mathbf{x} \in [\mathbf{x}], t_{\mathbb{A}}(\mathbf{x}) = 1) \Leftrightarrow ([\mathbf{x}] \subset \mathbb{A}), \\ ([t_{\mathbb{A}}]([\mathbf{x}]) = 0) &\Leftrightarrow (\forall \mathbf{x} \in [\mathbf{x}], t_{\mathbb{A}}(\mathbf{x}) = 0) \Leftrightarrow ([\mathbf{x}] \cap \mathbb{A} = \emptyset), \\ ([t_{\mathbb{A}}]([\mathbf{x}]) = [0, 1]) &\Leftrightarrow [\mathbf{x}] \cap \mathbb{A} \neq \emptyset \text{ but } [\mathbf{x}] \not\subset \mathbb{A}. \end{aligned} \quad (\text{II.39})$$

If there exist two inclusion tests $[t_{\mathbb{A}}]$ and $[t'_{\mathbb{A}}] \in \mathbb{IB}$ satisfying,

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, [t_{\mathbb{A}}]([\mathbf{x}]) \subset [t'_{\mathbb{A}}]([\mathbf{x}]), \quad (\text{II.40})$$

the inclusion test $[t_{\mathbb{A}}]$ is said to be more accurate than $[t'_{\mathbb{A}}]$. Couples of properties on set/ensemble operation are equally able to be applied on inclusion test $[t_{\square}]$, such as union, intersection or complementation. If, $\forall \mathbf{x} \in \mathbb{R}^n, [t_{\mathbb{A}}](\mathbf{x}) \neq [0, 1]$, then test $[t_{\mathbb{A}}]$ is supposed to be thin. For two such thin inclusion tests $[t_{\mathbb{A}}]([\mathbf{x}])$ and $[t_{\mathbb{B}}]([\mathbf{x}])$, the inclusion tests for their integrated sets $\mathbb{A} \cap \mathbb{B}$, $\mathbb{A} \cup \mathbb{B}$ and $\neg \mathbb{A} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \notin \mathbb{A}\}$ are defined consecutively in Eq. II.41.

$$\begin{aligned} [t_{\mathbb{A} \cap \mathbb{B}}]([\mathbf{x}]) &= ([t_{\mathbb{A}}] \cap [t_{\mathbb{B}}])([\mathbf{x}]) = [t_{\mathbb{A}}]([\mathbf{x}]) \cap [t_{\mathbb{B}}]([\mathbf{x}]), \\ [t_{\mathbb{A} \cup \mathbb{B}}]([\mathbf{x}]) &= ([t_{\mathbb{A}}] \cup [t_{\mathbb{B}}])([\mathbf{x}]) = [t_{\mathbb{A}}]([\mathbf{x}]) \cup [t_{\mathbb{B}}]([\mathbf{x}]), \\ [t_{\neg(\mathbb{A})}]([\mathbf{x}]) &= \neg [t_{\mathbb{A}}]([\mathbf{x}]) = 1 - [t_{\mathbb{A}}]([\mathbf{x}]). \end{aligned} \quad (\text{II.41})$$

II.3.1.6 Subpavings

A paving or a box is denoted by an interval vector $[\mathbf{x}]$ of \mathbb{IR}^n . From the view of extreme approximation in optimal analysis, any form of $[\mathbf{x}]$ is possible to be redefined as a union of sufficient small pavings $[\mathbf{x}]_k$ of \mathbb{IR}^n . A subpaving [Jaulin 1994; Kieffer 1999; Sam-Haroud et Faltings 1996] is considered as a set of non-overlapping pavings/boxes of \mathbb{IR}^n ,

$$\text{subpaving} = \bigcup_k [\mathbf{x}]_k, \text{ with } [\mathbf{x}]_k \cap [\mathbf{x}]_i = \emptyset, k \neq i. \quad (\text{II.42})$$

When a succession of finite operations of bisection and selection is carried out on an interval vector $[\mathbf{x}]$, smaller and more specific boxes are deemed to be generated. The set of such attentive boxes is considered to be a regular subpaving of $[\mathbf{x}]$ illustrated in gray in Figure II.7. To formulate the concept of subpaving precisely and illustratively,

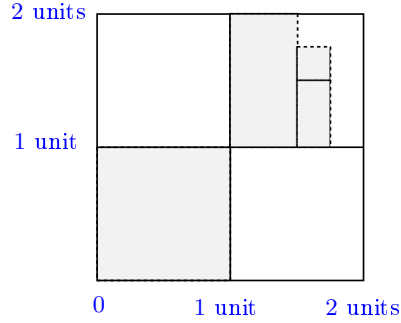


Figure II.7: A regular subpaving

consider first an interval vector $[\mathbf{x}]$ of \mathbb{IR}^n ,

$$\begin{aligned} [\mathbf{x}] &= [x_1] \times [x_2] \times \dots \times [x_n] \\ &= [\underline{x}_1, \overline{x}_1] \times [\underline{x}_2, \overline{x}_2] \times \dots \times [\underline{x}_n, \overline{x}_n], \end{aligned} \quad (\text{II.43})$$

Then select the j -th interval $[x_j]$ who owns the maximum width among all interval components through Eq. II.44,

$$j = \min\{i \mid w([x_i]) = w([\mathbf{x}])\}. \quad (\text{II.44})$$

Then, the first batch of subpaving is obtained by calculating as in Eq. II.45, where L and R respectively denote the left side and the right side of $[\mathbf{x}]$. This is called the bisection process. By making a fusion of both sides $L[\mathbf{x}]$ and $R[\mathbf{x}]$, one obtains the parent paving $[\mathbf{x}]$.

$$\begin{aligned} L[\mathbf{x}] &= [\underline{x}_1, \overline{x}_1] \times \dots \times \left[\underline{x}_j, \frac{x_j + \overline{x}_j}{2} \right] \times \dots \times [\underline{x}_n, \overline{x}_n], \\ R[\mathbf{x}] &= [\underline{x}_1, \overline{x}_1] \times \dots \times \left[\frac{x_j + \overline{x}_j}{2}, \overline{x}_j \right] \times \dots \times [\underline{x}_n, \overline{x}_n]. \\ [\mathbf{x}] &= L[\mathbf{x}] \sqcup R[\mathbf{x}]. \end{aligned} \quad (\text{II.45})$$

Consecutively conduct and repeat the phases of bisection as above, the generations of paving of $[\mathbf{x}]$ are then produced in notions of $LL[\mathbf{x}]$, $RL[\mathbf{x}]$, $RR[\mathbf{x}]$, ..., etc. Refer to the box in Figure II.7 supposing that 1 unit equals to 1, a subpaving of $[\mathbf{x}] = [0, 2] \times [0, 2]$

II. INTERVAL ANALYSIS

is generated as follows,

$$\begin{aligned}
 L[\mathbf{x}] &= [0, 1] \times [0, 2], \\
 LL[\mathbf{x}] &= [0, 1] \times [0, 1], \\
 LRR[\mathbf{x}] &= [1, \frac{3}{2}] \times [1, 2], \\
 LLRRR[\mathbf{x}] &= [\frac{3}{2}, \frac{7}{4}] \times [1, \frac{3}{2}], \\
 LLRRRR[\mathbf{x}] &= [\frac{3}{2}, \frac{7}{4}] \times [\frac{3}{2}, \frac{7}{4}].
 \end{aligned}$$

Alternatively, the bisection procedure could be represented by a binary tree structure in an computer environment since the partitioned parent and the descendant groups look extremely similar with a tree. A binary tree is composed of a finite set of nodes. This nodes set could be empty, could be a single node namely the root of the tree, or may additionally contain two binary subtrees downside whose intersection is supposed to be empty. According to this assumption, an example of a tree structure is illustrated in Figure II.8 which is capable to give more detailed bisection rules. In the illustration of

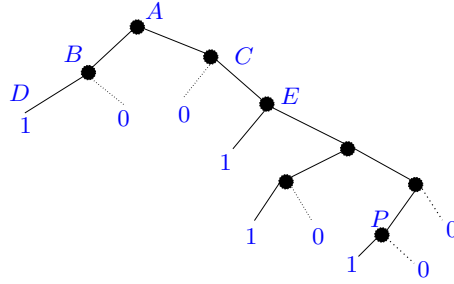


Figure II.8: A tree for regular subpaving

subpaving, suppose one unit equals to numeral 4. Node A , the root of the tree, stands for the initial box $\mathbf{x} = [0, 8] \times [0, 8]$ partitioned as follows. B and C are the left and right generations of A , corresponding respectively to the sub-boxes $[0, 4] \times [0, 8]$ and $[4, 8] \times [0, 8]$. The sub-box B is also partitioned leading to the sub-box $[0, 4] \times [0, 4]$ or notation D . D has no following subtree thus it is considered as a leaf in the tree. Putting 1 next to D means that D is selected in the subpaving. In the part of right generation, node C further produces next generation at right side, or subtree, denoting part of right-up corner $[4, 8] \times [4, 8]$ or notation E . Generation E owns the next generation of double sides, denoted respectively by $[4, 6] \times [4, 8]$ (selected with 1 next to it) and $[6, 8] \times [4, 8]$, further bisected leading to additional sub-boxes. Subdivision of the tree continues until all sub-boxes are selected, as shown in Figure II.8. The number of the necessary steps of bisections from the root node to form a box is defined as depth of the box. For example, box E goes to depth of three. A tree is considered to be minimal if it has no sibling leaves which could be obtain by cutting the leaves of any non-minimal tree representing a subpaving of initial box \mathbf{x} .

The basic operations on regular subpavings is likewise willing to be presented here with aid of binary tree since the subpaving represents and illustrates alternatively the available box $[\mathbf{x}]$. There exist four basic operations paving-oriented enumerated as reuniting the sibling subpavings, intersecting subpavings, taking the union of subpavings and testing whether a box is included in a subpaving.

The sibling subpavings as mentioned above are actually situated at the same level with the parents. It is volunteered to be reunited to form a new family/tree. The reunite algorithm is trivial to implement and is described as follows. The set of all the regular subpavings of box $[\mathbf{x}]$ is denote by $RSP([\mathbf{x}])$.

Algorithm 1 *The reunite algorithm: existing initial box $[\mathbf{x}]$ and its two regular subpavings $\mathbb{X} \in RSP(L[\mathbf{x}])$ as well as $\mathbb{Y} \in RSP(R[\mathbf{x}])$ of the first generation of parent $[\mathbf{x}]$, the computation of reunited subpaving $\mathbb{Z} = (\mathbb{X}, \mathbb{Y}) \in RSP([\mathbf{x}])$ is as followed.*

- if $\mathbb{X} = L[\mathbf{x}]$ and $\mathbb{Y} = R[\mathbf{x}]$, then the reunited subpaving $\mathbb{Z} = [\mathbf{x}]$;
- else if $\mathbb{X} = \emptyset$ and $\mathbb{Y} = \emptyset$, then $\mathbb{Z} = \emptyset$;
- else, $L\mathbb{Z} = \mathbb{X}$ and $R\mathbb{Z} = \mathbb{Y}$.

If intersecting two regular subpavings $\mathbb{X} \in RSP([\mathbf{x}])$ and $\mathbb{Y} \in RSP([\mathbf{x}])$, the intersected result leads to the shared nodes in both trees, or boxes. This procedure is illustrated as follows.

Algorithm 2 *The intersection algorithm:*

- if $\mathbb{X} = \emptyset$ or $\mathbb{Y} = \emptyset$, then $\mathbb{Z} = \emptyset$;
- else if $\mathbb{X} = [\mathbf{x}]$, then $\mathbb{Z} = \mathbb{Y}$;
- else if $\mathbb{Y} = [\mathbf{x}]$, then $\mathbb{Z} = \mathbb{X}$;
- else, $\mathbb{Z} = (INTER(L\mathbb{X}, L\mathbb{Y}, L[\mathbf{x}]), INTER(R\mathbb{X}, R\mathbb{Y}, R[\mathbf{x}]))$.

If uniting two arbitrary subpavings $\mathbb{X} \in RSP([\mathbf{x}])$ and $\mathbb{Y} \in RSP([\mathbf{x}])$, a new expanded subpaving $\mathbb{Z} = \mathbb{X} \cup \mathbb{Y} \in RSP[\mathbf{x}]$ is obtained simply by putting all nodes together.

Algorithm 3 *The union algorithm:*

- if $\mathbb{X} = \emptyset$ or $\mathbb{Y} = [\mathbf{x}]$, then $\mathbb{Z} = \mathbb{Y}$;
- else if $\mathbb{Y} = \emptyset$ or $\mathbb{X} = [\mathbf{x}]$, then $\mathbb{Z} = \mathbb{X}$;
- else, $\mathbb{Z} = (UNION(L\mathbb{X}, L\mathbb{Y}, L[\mathbf{x}]), UNION(R\mathbb{X}, R\mathbb{Y}, R[\mathbf{x}]))$.

To make a judgement whether a box $[\mathbf{z}]$ is included in a subpaving $\mathbb{X} \in RSP[\mathbf{x}]$ or not, tests are firstly applied to the root of tree $[\mathbf{x}]$, then to its left and right subtrees in sequence and denoted by 1 and 0 if the inside judgement result is true or false.

II. INTERVAL ANALYSIS

Algorithm 4 *The inside algorithm:*

- if $[z] = \emptyset$ or if (\mathbb{X} is a box $[\mathbf{x}]$ and $[z] \subset [\mathbf{x}]$), then $t := 1$;
- else if $\mathbb{X} = \emptyset$, then $t := 0$;
- else, $t := (\text{INSIDE}([z] \cap L[\mathbf{x}], L\mathbb{X}) \cup \text{INSIDE}([z] \cap R[\mathbf{x}], R\mathbb{X}))$.

II.3.2 Basic algorithms

Under the basic knowledge on the operations and properties of interval ensemble, vector, inclusion function and regular subpavings, it is possible to go further into introducing two basic algorithms of interval analysis, which are able to be largely applied in the real engineering problems. In this section, the inversion algorithm SIVIA and some compact contractors are going to be discussed aiming to solve the non-linear equations problem.

II.3.2.1 Set inversion SIVIA

In many cases, the problem often encountered is solving the inversion function given a range of the function values. For example, let \mathbf{f} be a function with argument \mathbf{x} respectively in real spaces of \mathbb{R}^m and \mathbb{R}^n . The inversion function of function \mathbf{f} is obtained as shown in Eq. II.46,

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \iff \mathbf{x} = \mathbf{f}^{-1}(\mathbf{y}). \quad (\text{II.46})$$

Similarly, given a possibly non-linear function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and a subpaving $\mathbb{Y} \subset \mathbb{R}^m$ obtained from \mathbf{f} . A set inversion operation is defined as computing a subpaving in \mathbb{R}^n denoted as in Eq. II.47,

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}). \quad (\text{II.47})$$

In order to approximate the solution of subpaving \mathbb{X} , for any function \mathbf{f} admitting a convergent inclusion function $[\mathbf{f}](\cdot)$, two regular subpavings denoted by $\underline{\mathbb{X}}$ and $\overline{\mathbb{X}}$ which constrain the inner and outer bounding approximations for \mathbb{X} as in Eq. II.48,

$$\underline{\mathbb{X}} \subseteq \mathbb{X} \subseteq \overline{\mathbb{X}}, \quad (\text{II.48})$$

are to be worked out. An algorithm called SIVIA (set inverter via interval analysis) is invented to solve this problem [Jaulin *et al.* 2001]. To acquire a guaranteed subpaving of $\overline{\mathbb{X}}$, SIVIA algorithm demands a very large database denoted as search box $[\mathbf{x}]_0$. The procedure of SIVIA algorithm to infer a solution of $\underline{\mathbb{X}}$ and $\overline{\mathbb{X}}$ is illustrated in a simplified example as Figure II.9 where four cases possibly encountered are explicated as follows.

- If the intersection of $[\mathbf{f}](\mathbf{x})$ and \mathbb{Y} is not empty and the former is not entirely included in the latter, then $[\mathbf{x}]$ is considered to be containing a partial solution set. This box is firstly assumed to be under-determined. If the width $w([\mathbf{x}])$ is greater than a threshold parameter ϵ , then it is decided to be bisected and tested recursively once updated. Figure II.9-a

- If the intersection of $[f]([\mathbf{x}])$ and \mathbb{Y} is an empty set, then $[\mathbf{x}]$ actually does not belong to \mathbb{X} . As consequence, this value could be cut off from the solution tree. Figure II.9-b
- If the $[f]([\mathbf{x}])$ is entirely included in \mathbb{Y} , then $[\mathbf{x}]$ is considered as belonging to the solution subpaving \mathbb{X} and stored in two bounding subpaving of $\underline{\mathbb{X}}$ and $\overline{\mathbb{X}}$. Figure II.9-c
- If $[\mathbf{x}]$ is determined as ambiguity box but with a small enough value of width $w([\mathbf{x}])$ sentenced by ϵ , then it is acceptable and stored in the outer approximation $\overline{\mathbb{X}}$ of \mathbb{X} . Figure II.9-d

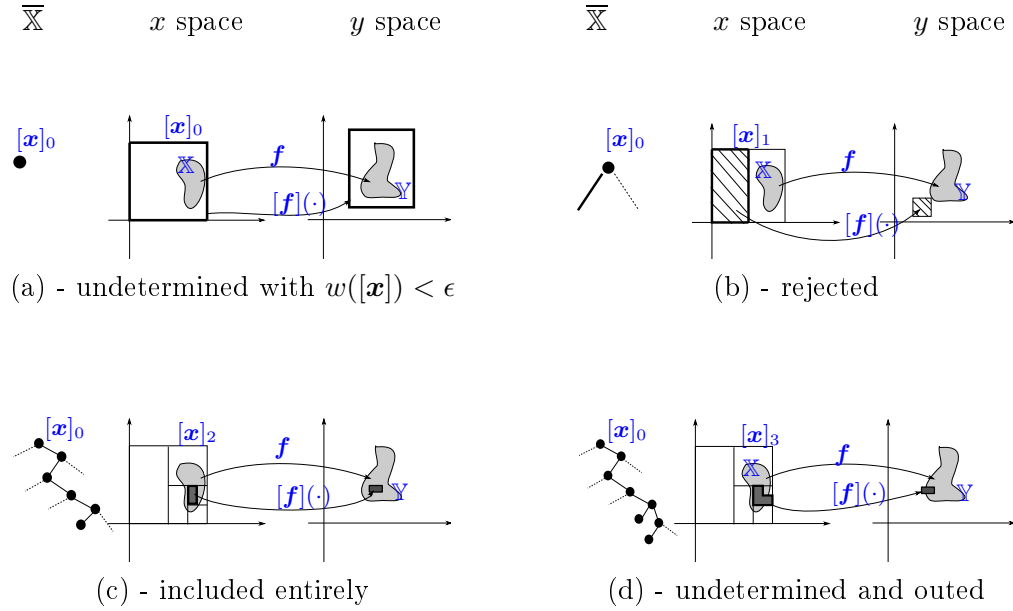


Figure II.9: SIVIA

Algorithm 5 SIVIA algorithm based on an inclusion function:

- if $[f]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$, return; Figure II.9-b
- if $[f]([\mathbf{x}]) \subset \mathbb{Y}$, then $\{\underline{\mathbb{X}} = \underline{\mathbb{X}} \cup [\mathbf{x}], \overline{\mathbb{X}} = \overline{\mathbb{X}} \cup [\mathbf{x}], \text{return};\}$; Figure II.9-c
- if $w([\mathbf{x}]) < \epsilon$, then $\{\overline{\mathbb{X}} = \overline{\mathbb{X}} \cup [\mathbf{x}], \text{return};\}$; Figure II.9-d
- SIVIA ($\mathbf{f}, \mathbb{Y}, L[\mathbf{x}], \epsilon, \underline{\mathbb{X}}, \overline{\mathbb{X}}$), SIVIA ($\mathbf{f}, \mathbb{Y}, R[\mathbf{x}], \epsilon, \underline{\mathbb{X}}, \overline{\mathbb{X}}$). Figure II.9-a

The recursive algorithm is summarized as the Algorithm.5 SIVIA above. Note that, the inner and outer subpavings $\underline{\mathbb{X}}$ and $\overline{\mathbb{X}}$ are initialized as emptysets.

II. INTERVAL ANALYSIS

II.3.2.2 Contractors

The SIVIA algorithm pursues stitched subpavings playing the role of inner and outer approximations for the non-linear problem. Alternatively, as regard to another thinking, consider firstly a function \mathbf{f} of \mathbb{R}^m whose coordinates vectors are composed of functions $f_j, j \in 1, 2, \dots, m$, which are in form as in Eq. II.49,

$$f_j(x_1, x_2, \dots, x_n) = 0, \quad j \in \{1, 2, \dots, m\} \quad (\text{II.49})$$

where the arguments of function f_j , namely vector \mathbf{x} , is expressed as $\mathbf{x} = (x_1, x_2, \dots, x_n)$ whose prior domain is also determined as $[\mathbf{x}] = [x_1] \times [x_2] \times \dots \times [x_n]$. According to this, the function \mathbf{f} could be written in vector form as $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, which corresponds to a constraint satisfaction problem(CSP) \mathcal{H} formulated as in Eq. II.50,

$$\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} \in [\mathbf{x}]). \quad (\text{II.50})$$

Thus this problem is only finding the solution set of \mathbb{R}^n as in Eq. II.51,

$$\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) = \mathbf{0}\}. \quad (\text{II.51})$$

Normally, such CSP problems are likely to involve equality and inequality constraints operated both in discrete domains, like the values taken from x_i belongs to finite sets, as well as in continuous domains, like subsets of \mathbb{R} or intervals [Clowes 1971; Cleary 1986].

In the framework of sets, a possible solution of CSP problem could be defined as an outer approximation of the real \mathbb{S} . The interval based methods do not only provide a guaranteed enclosure $[\mathbf{x}]$ of \mathbb{S} in time and space, especially, it offers an opportunity of contracting the solution of \mathcal{H} into smaller domain $[\mathbf{x}']$, which means,

$$\mathbb{S} \subset [\mathbf{x}'] \subset [\mathbf{x}]. \quad (\text{II.52})$$

If $[\mathbf{x}]$ is able to be contracted into a smallest box containing \mathbb{S} , that is considered as an optimal contraction of \mathcal{H} . Accordingly the contraction operator is nominated as contractor denoted by \mathcal{C} .

Aiming to solve some specific classes of CSP problems, there exist several basic contractors which are efficient even for a much larger class of CSPs. In the following section below, the basic contractors, \mathcal{C}_{GE} , \mathcal{C}_{Gs} , \mathcal{C}_K and $\mathcal{C}_{\downarrow\uparrow}$, used to get efficient intervalized/interval finite solvers are presented to deal with various optimization problems and resolution of equations and inequalities.

Consider the CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} \in [\mathbf{x}])$, a finite subsolver could be obtained by computing the values of some variables x_i while some other variables $x_j, j \neq i$, are known. The formal definition of a subsolver is determined in virtue of the definition of a subvector of a given vector \mathbf{x} . The vector $\mathbf{u} = (u_1, u_2, \dots, u_{n_u})$ is a subvector of vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ if there exists a subset $\mathcal{I} = \{i_1, i_2, \dots, i_{n_u}\}$ belonging to set $\{1, 2, \dots, n\}$ such as $\mathbf{u} = (x_{i_1}, x_{i_2}, \dots, x_{i_{n_u}})$ equally possibly written by $\mathbf{u} = \mathbf{x}_{\mathcal{I}}$. Meanwhile, if there is a subset $\mathcal{J} = \{j_1, j_2, \dots, j_{n_\phi}\}$ of the same vector \mathbf{x} carrying

subvectors $\mathbf{v} = \mathbf{x}_{\mathcal{J}}$ satisfying $\mathcal{I} \cap \mathcal{J} = \emptyset$, then a finite subsolver of \mathcal{H} is defined as a finite set-values algorithm $\phi: \mathbf{u} \mapsto \phi(\mathbf{u})$ making Eq. II.53 hold true.

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \Rightarrow \mathbf{v} \in \phi(\mathbf{u}). \quad (\text{II.53})$$

The components of subvector \mathbf{u} are the inputs of ϕ where those of \mathbf{v} are the corresponding outputs. More often, function $\phi(\mathbf{u})$ is a singleton, translating Eq. II.53 to Eq. II.54,

$$\mathbf{v} = \phi(\mathbf{u}). \quad (\text{II.54})$$

Notion of subsolver ϕ is equally applied on interval counterpart. An inclusion function $[\phi]$ of ϕ transformed from \mathbb{R}^{n_u} to \mathbb{R}^{n_ϕ} satisfies for all pavings/boxes \mathbf{u} ,

$$\phi([\mathbf{u}]) \subset [\phi]([\mathbf{u}]), \text{ where } \phi([\mathbf{u}]) = \cup_{\mathbf{u} \in [\mathbf{u}]} \phi(\mathbf{u}). \quad (\text{II.55})$$

As a consequence, if $[\phi]$ is an inclusion function of ϕ , a contraction of \mathcal{H} can be derived by updating each domain $[x_j], j \in \mathcal{J}$ by the domain $[x_j] \cap [\phi_j]([\mathbf{u}])$, such as in Eq. II.56.

$$[x_j] = [x_j] \cap [\phi_j]([\mathbf{u}]). \quad (\text{II.56})$$

One classical Gauss elimination contractor is proposed to solve the CSPs of square linear systems of interval equations. For a specific CSP problem shown as in Eq. II.57,

$$\mathcal{H}: \left(\begin{array}{l} \mathbf{A} \in [\mathbf{A}], \mathbf{b} \in [\mathbf{b}], \mathbf{p} \in [\mathbf{p}] \\ \mathbf{A}\mathbf{p} - \mathbf{b} = (\mathbf{0}) \end{array} \right), \quad (\text{II.57})$$

with

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n_p} \\ \vdots & & & \vdots \\ a_{n_p 1} & a_{n_p 2} & \cdots & a_{n_p n_p} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_{n_p} \end{pmatrix}. \quad (\text{II.58})$$

The Gauss elimination procedure is considered as a finite subsolver permitting to compute \mathbf{p} when input vector $\mathbf{u} = (a_{11}, a_{12}, \dots, a_{n_p n_p}, b_1, b_2, \dots, b_{n_p})$ is known inferred from the set of variables of \mathcal{H} $\mathbf{x} = (a_{11}, a_{12}, \dots, a_{n_p n_p}, p_1, p_2, \dots, p_{n_p}, b_1, b_2, \dots, b_{n_p})$. The operator of Gauss elimination is defined as in Eq. II.59,

$$\mathcal{C}_{GE}([\mathbf{A}], [\mathbf{b}], [\mathbf{p}]) \mapsto ([\mathbf{A}], [\mathbf{b}], [\mathbf{p}] \cap [\phi]([\mathbf{A}], [\mathbf{p}], [\mathbf{b}])), \quad (\text{II.59})$$

where $[\phi](\cdot)$ is given by Algorithm 6. *GE* denotes Gauss elimination contractor for CSP problem \mathcal{H} .

Algorithm 6 *Gauss elimination contractor:*

1. for $i = 1$ to $n_p - 1$
2. if $0 \in [a_{ii}]$
3. $([p_1], \dots, [p_{n_p}]) = \mathbb{R}^{n_p}$; return;

II. INTERVAL ANALYSIS

4. for $j = i + 1$ to n_p
5. $[\alpha_j] = [a_{ji}]/[a_{ii}]; [b_j] = [b_j] - [\alpha_j] * [b_i];$
6. for $k = i + 1$ to n_p
7. $[a_{jk}] = [a_{jk}] - [\alpha_j] * [a_{ik}];$
8. for $i = n_p$ down to 1
9. $[p_i] = ([b_i] - \sum_{j=i+1}^n [a_{ij}] * [p_j])/[a_{ii}].$

A Gauss-Seidel based contractor is also proposed to solve the same CSP problem,

$$\mathcal{H} : \left(\begin{array}{l} \mathbf{A} \in [\mathbf{A}], \mathbf{b} \in [\mathbf{b}], \mathbf{p} \in [\mathbf{p}] \\ \mathbf{A}\mathbf{p} - \mathbf{b} = \mathbf{0} \end{array} \right), \quad (\text{II.60})$$

where the matrix \mathbf{A} is assumed to be square matrix. \mathbf{A} is then deemed to be decomposed as the sum of a diagonal matrix and its corresponding matrix making all the components on diagonal equal to zero as shown in Eq. II.61,

$$\mathbf{A} = \text{diag}(\mathbf{A}) + \text{extdiag}(\mathbf{A}). \quad (\text{II.61})$$

Consequently, $\mathbf{A}\mathbf{p} - \mathbf{b} = \mathbf{0}$ is equivalent to

$$\text{diag}(\mathbf{A})\mathbf{p} + \text{extdiag}(\mathbf{A})\mathbf{p} - \mathbf{b} = \mathbf{0}. \quad (\text{II.62})$$

Assuming that there exists the inverse matrix $(\text{diag}(\mathbf{A}))^{-1}$ corresponding to $\text{diag}(\mathbf{A})$, Eq. II.62 is rewritten as Eq. II.63,

$$\mathbf{p} = (\text{diag}(\mathbf{A}))^{-1}(\mathbf{b} - \text{extdiag}(\mathbf{A})\mathbf{p}). \quad (\text{II.63})$$

Then formulate a fixed-point subsolver for \mathcal{H} as

$$\psi \left(\begin{array}{l} \mathbf{A} \\ \mathbf{b} \\ \mathbf{p} \end{array} \right) = \left(\begin{array}{l} \mathbf{A} \\ \mathbf{b} \\ (\text{diag}(\mathbf{A}))^{-1}(\mathbf{b} - \text{extdiag}(\mathbf{A})\mathbf{p}) \end{array} \right). \quad (\text{II.64})$$

Applying the inclusion function on ψ then leads to

$$\psi \left(\begin{array}{l} [\mathbf{A}] \\ [\mathbf{b}] \\ [\mathbf{p}] \end{array} \right) = \left(\begin{array}{l} [\mathbf{A}] \\ [\mathbf{b}] \\ (\text{diag}([\mathbf{A}]))^{-1}([\mathbf{b}] - \text{extdiag}([\mathbf{A}])\mathbf{p}) \end{array} \right). \quad (\text{II.65})$$

The Gauss-Seidel contractor is generated accordingly as

$$C_{GS} : [\psi] \left(\begin{array}{l} [\mathbf{A}] \\ [\mathbf{b}] \\ [\mathbf{p}] \end{array} \right) \mapsto \left(\begin{array}{l} [\mathbf{A}] \\ [\mathbf{b}] \\ [\mathbf{p}] \cap (\text{diag}([\mathbf{A}]))^{-1}([\mathbf{b}] - \text{extdiag}([\mathbf{A}])\mathbf{p}) \end{array} \right). \quad (\text{II.66})$$

Another basic contractor is proposed based on Krawczyk [Neumaier 1990]. Considering the CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = 0, \mathbf{x} \in [\mathbf{x}])$, the Krawczyk contractor \mathcal{C}_K utilizes the fixed-point subsolver in the same manner as \mathcal{C}_{GS} . If satisfying $n_f = n$ and a differentiable \mathbf{f} , for any matrix \mathbf{M} ,

$$\mathbf{f}(x) = 0 \Leftrightarrow \mathbf{x} - \mathbf{M}\mathbf{f}(\mathbf{x}) = \mathbf{x}. \quad (\text{II.67})$$

The subsolver $\psi(\mathbf{x})$ is then defined as in Eq. II.68,

$$\psi(\mathbf{x}) = \mathbf{x} - \mathbf{M}\mathbf{f}(\mathbf{x}). \quad (\text{II.68})$$

Applying the centered inclusion function on ψ leads to Eq. II.69,

$$[\psi]([\mathbf{x}]) = \psi(\mathbf{x}_0) + [\mathbf{J}_\psi]([\mathbf{x}]) * ([\mathbf{x}] - \mathbf{x}_0), \quad (\text{II.69})$$

where $[\mathbf{J}_\psi]$ is an inclusion function for the Jacobian matrix of ψ . \mathbf{x}_0 is the center point $mid([\mathbf{x}])$ of $[\mathbf{x}]$. Then a Krawczyk contractor reducing $[\mathbf{x}]$ is obtained by Eq. II.70,

$$\mathcal{C}_K : [\mathbf{x}] \mapsto [\mathbf{x}] \cap (\psi(\mathbf{x}_0) + [\mathbf{J}_\psi]([\mathbf{x}]) * ([\mathbf{x}] - \mathbf{x}_0)). \quad (\text{II.70})$$

Substituting Eq. II.68 into Eq. II.70 leads to

$$\mathcal{C}_K : [\mathbf{x}] \mapsto [\mathbf{x}] \cap (\mathbf{x}_0 - \mathbf{M}\mathbf{f}(\mathbf{x}_0) + (\mathbf{I} - \mathbf{M}[\mathbf{J}_f]([\mathbf{x}])) * ([\mathbf{x}] - \mathbf{x}_0)). \quad (\text{II.71})$$

where notion \mathbf{I} is the identity matrix, $[\mathbf{J}_f]$ is an inclusion function for the Jacobian matrix of function \mathbf{f} . The matrix \mathbf{M} is often defined as the inverse function $\mathbf{J}_f^{-1}(\mathbf{x}_0)$ computed at value \mathbf{x}_0 . The relative algorithm is illustrated as Algorithm.7.

Algorithm 7 *Krawczyk contractor:*

1. $\mathbf{x}_0 = mid([\mathbf{x}]);$
2. $\mathbf{M} = \mathbf{J}_f^{-1}(\mathbf{x}_0);$
3. $[\mathbf{J}_\psi] = \mathbf{I} - \mathbf{M}[\mathbf{J}_f]([\mathbf{x}]);$
4. $[\mathbf{r}] = \mathbf{x}_0 - \mathbf{M}\mathbf{f}(\mathbf{x}_0) + [\mathbf{J}_\psi] * ([\mathbf{x}] - \mathbf{x}_0);$
5. $[\mathbf{x}] = [\mathbf{x}] \cap [\mathbf{r}].$

The fourth contractor, forward-backward contractor $\mathcal{C}_{\downarrow\uparrow}$, is a problem solver on basis of the principal of constraint propagation [Waltz 1972; Cleary 1986; Davis 1987; Benhamou *et al.* 1999], enabling to reduce the domain of the CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = 0, \mathbf{x} \in [\mathbf{x}])$ by iterating all the constraints consecutively with assumption of any $f_i(x_1, x_2, \dots, x_n) = 0$. The dimension of \mathbf{f} is not any long necessarily equaling to n in this case. Each iteration on constraint give a different finite subsolver. Consequently the contractor $\mathcal{C}_{\downarrow\uparrow}$ performs the intersection of all various subsolvers to reduce paving/box $[\mathbf{x}]$.

However, for each constraint in form of $f_i(x_1, x_2, \dots, x_n) = 0$, f_i is possible and deemed to be decomposed into a sequence of operations, involving the basic operators $(+, -, *, /)$

II. INTERVAL ANALYSIS

or functions (sin, cos, log, ...). Alternatively speaking, a constraint is decomposed into the primitive constraints [Waltz 1972] which are in fact single constraints involving just one single operator included in (+, -, *, /), or single function (sin, cos, log, ...). The contraction procedure is actually performed at each of the primitive constraints until the contractor degrades to be inefficient.

Consider an example of constraint equation $f(\mathbf{x}) = 0$ where $f(\mathbf{x}) = x_1 \cdot \exp(x_2) + \sin(x_3)$. The objected domains of variables x_1, x_2 and x_3 are denoted as $[x_1], [x_2]$ and $[x_3]$. To contract such domains by the forward-backward method, it needs firstly to write down an imaginary finite sequence of elementary operations in interval counterpart to apply, for example as in Eq. II.72,

$$\begin{aligned}
 [a_1] &= \exp([x_2]); \\
 [a_2] &= [x_1] * [a_1]; \\
 [a_3] &= \sin([x_3]); \\
 [y] &= [a_2] + [a_3].
 \end{aligned}
 \tag{II.72}$$

The forward-backward algorithm is conducted and concluded as in Algorithm.8.

Algorithm 8 *Forward-backward contractor:*

1. $[a_1] = \exp([x_2]);$
2. $[a_2] = [x_1] * [a_1];$
3. $[a_3] = \sin([x_3]);$
4. $[y] = [a_2] + [a_3];$
5. $[y] = [y] \cap [0];$
6. $[a_2] = ([y] - [a_3]) \cap [a_2];$
7. $[a_3] = ([y] - [a_2]) \cap [x_3]$
8. $[x_3] = \sin^{-1}([a_3]) \cap [x_3];$
9. $[a_1] = ([a_2]/[x_1]) \cap [a_1];$
10. $[x_1] = ([a_2]/[a_1]) \cap [x_1]$
11. $[x_2] = \log([a_1]) \cap [x_2].$

Besides these basic contractors, there exist also some extended and collaborative contractors, as well as that for sets operations. Contractor is a very useful notion of interval analysis since it is capable to downsize the search paving/box by a variety of thinking nevertheless without any loss of all the possibilities. It also serves basically for an advanced function called solvers in concrete analysis, which is not included in this paper. For more information about contractors, please refer to [Moore 1966; Moore et Moore 1979; Moore et al. 2009; Kearfott et Kreinovich 1996; Jaulin et al. 2001].

II.4 Conclusion

Interval analysis technique is a guaranteed numerical approximation approach on sets operations applicable on the unsolvable non-linear problem encountered many often in the area of engineering. The main theoretical foundation relies on a rather simple conception of enclosing real numbers in intervals, or real vectors in boxes ensuring guaranteed results and numerical processions. This chapter briefly introduces the concept of interval analysis. Based on the basic operations such as on ensembles and arithmetics, inclusion function is founded and employed to help producing the outer approximation of problem solutions. Equally, subpaving, an inverse evaluation of function on basis of subpaving planning of box $[\mathbf{x}]$, is then introduced in name of SIVIA leading to a union of possible sets covering the real set of inner and outer approximations which are generated from \emptyset . Supplementally, with the aid of contractors based on inclusion functions, the initial possibly large search box $[\mathbf{x}]$ in solving constraint problems is able to be reduced up to satisfy specification, however without any loss of possible sets of solution of the optimal problem.

The next chapter will illustrate a combined centralized localization method with aid of interval analysis showing how the interval analysis technique is employed to help localize a mobile sensor in a wireless sensor network. Meanwhile, the corresponding reliability is also analyzed.

REFERENCES

References

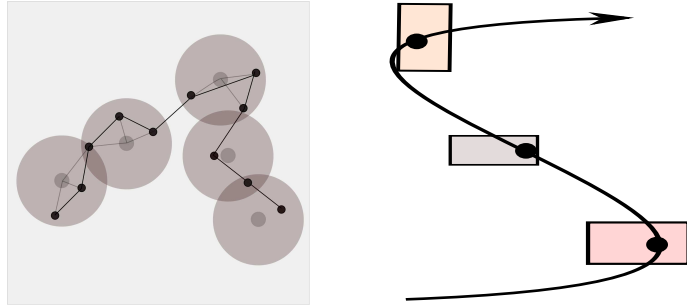
- [Benhamou *et al.*, 1999] F. Benhamou, F. Goualard, L. Granvilliers et J.-F. Puget. Revising hull and box consistency. In *Int. Conf. on Logic Programming*. Citeseer, 1999. 51
- [Cleary, 1986] J. G. Cleary. Logical arithmetic. 1986. 48, 51
- [Clowes, 1971] M. B. Clowes. On seeing things. *Artificial intelligence*, 2 (1): 79–116, 1971. 48
- [Davis, 1987] E. Davis. Constraint propagation with interval labels. *Artificial intelligence*, 32 (3): 281–331, 1987. 51
- [Hansen et Walster, 2003] E. Hansen et G. W. Walster. *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press, 2003. 33
- [Jaulin, 1994] L. Jaulin. Solution globale et garantie de problèmes ensemblistes; application à l'estimation non linéaire et à la commande robuste. orsay, france, 1994. 42
- [Jaulin *et al.*, 2001] L. Jaulin, M. Kieffer, O. Didrit et E. Walter. *Applied interval analysis*. Springer, 2001. 32, 35, 46, 52
- [Jaulin et Walter, 1993] L. Jaulin et E. Walter. Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Mathematics and Computers in Simulation*, 35 (2): 123–137, 1993. 33
- [Kearfott et Kreinovich, 1996] R. B. Kearfott et V. Kreinovich. *Applications of interval computations*, volume 3. Kluwer Academic Dordrecht, 1996. 32, 35, 52
- [Kieffer, 1999] M. Kieffer. *Estimation ensembliste par analyse par intervalles Application à la localisation d'un véhicule*. Thèse de doctorat, Université Paris Sud-Paris XI, 1999. 42
- [Moore, 1992] R. Moore. Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, 34 (2): 113–119, 1992. 33
- [Moore, 1966] R. E. Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966. 32, 52
- [Moore *et al.*, 2009] R. E. Moore, R. B. Kearfott et M. J. Cloud. *Introduction to interval analysis*. Siam, 2009. 52
- [Moore et Moore, 1979] R. E. Moore et R. Moore. *Methods and applications of interval analysis*, volume 2. SIAM, 1979. 35, 52
- [Neumaier, 1990] A. Neumaier. *Interval methods for systems of equations*, volume 37. Cambridge university press, 1990. 33, 51
- [Sam-Haroud et Faltings, 1996] D. Sam-Haroud et B. Faltings. Consistency techniques for continuous constraints. *Constraints*, 1 (1-2): 85–118, 1996. 42
- [Sunaga, 1958] T. Sunaga. Theory of an interval algebra and its application to numerical analysis. *Research Association of Applied Geometry (RAAG) Memoirs*, 2: 29–46, 1958. 32
- [Sunaga, 2009] T. Sunaga. Theory of an interval algebra and its application to numerical analysis. *Japan Journal of Industrial and Applied Mathematics*, 26 (2): 125–143, 2009. 32

REFERENCES

- [Walter et Jaulin, 1994] E. Walter et L. Jaulin. Guaranteed characterization of stability domains via set inversion. *Automatic Control, IEEE Transactions on*, 39 (4): 886–889, 1994. [33](#)
- [Waltz, 1972] D. L. Waltz. Generating semantic descriptions from drawings of scenes with shadows. Rapport interne, DTIC Document, 1972. [51](#), [52](#)

REFERENCES

III Centralized localization



Contents

III.1	Introduction
III.2	Problem statement
III.3	Centralized localization algorithm
III.4	Analysis and simulations
III.5	Conclusion
	References

The capability of sensors for self-localization is one of the basic and crucial requirements in ad-hoc wireless sensor networks. People prefer the robust, flexible and reliable localization algorithms to ensure a guaranteed estimation of sensors positions. Interval analysis technique, presented in the preceding chapter, is capable to solve this localization problem in the interval framework by introducing the uncertainties of relevant measurements in wireless sensor networks. Measurements used in the method introduced in this chapter include both fingerprints and accelerometer information as nodes move freely. The former, in terms of RSSI measurements, serves for two phases of measurements. It is firstly used to configure a priori a global power map over the surveillance area, then compute the first estimate by the way of pattern matching once the second measurements are acquired in the real-time phase. A second estimate relies on the latter, namely acceleration values measured from accelerometer device, as well as the last renewed estimate. Both instant estimates are then integrated using interval analysis, yielding solutions in a form of boxes including the real positions of the nodes.

III. CENTRALIZED LOCALIZATION

III.1 Introduction

Recent developments in electronics miniaturization such as micro-electromechanical systems, embedded micro-processors, as well as in wireless communication techniques enable the emerging and boosting of the Wireless Sensor Networks (WSNs) technology [Akyildiz *et al.* 2002; Zhao *et Guibas* 2004; Conner *et al.* 2004]. WSNs consist of ad-hoc networks composed of a large number of smart and tiny devices, called also sensors, having sensing, computation and communication capabilities [Akyildiz *et al.* 2002]. Sensors are autonomously driven by their light embedded power sources, i.e. on-board batteries. Such power sources are generally non-renewable, hence all communication protocols and processing algorithms should be low-energy consuming to extend the lifespan of the whole network. WSNs have been widely applied for surveillance and tracking in many domains surrounding peoples, like agriculture, industry, medicine and military fields, making meaningful and great achievements [Shorey *et al.* 2006; Czubak *et Wojtanowski* 2009]. In this context, position awareness of the sensors is crucial, since all sensed data are tightly related to the geographical locations where measurements are made.

One solution to this problem is to equip all sensors with Global Positioning Systems (GPS) [Hofmann-Wellenhof *et al.* 1993]. It is quite efficient with regard to localization. Nevertheless, this kind of heavily-armed large scale ad-hoc WSNs means meanwhile the energy-consuming, high cost and bulky burden for tiny sensors. Moreover, GPS receivers are constrained in an indoor or large covered background, hence they turn to be impractical in these situations, GPS signals not being reliable any way.

Massive alternative solutions have been proposed since the passed last decades. Many classical solutions are already roughly described in Chapter I. They use two types of sensors: *anchors*, having known positions, and *nodes*, having unknown positions. Such various localization approaches are able to be categorized into distance estimation, connectivity measurements or scenario information collection based methods [Mao *et al.* 2007], or in term of processing scheme, into centralized or decentralized methods [Stojmenovic 2005; Pal 2010], or in term of adopted signals, into acoustic based, optic assisted and Received Signal Strength Indicator (RSSI) [Schweinzer *et Syafrudin* 2010; Sifuentes *et al.* 2011; Kaemarungsi *et Krishnamurthy* 2004; Bahl *et Padmanabhan* 2000], etc. Considering the simplicity and flexibility of implementation aligned with the reasonable positioning accuracy, RSSI based techniques are taken into account more often. To improve the positioning accuracy of RSSI-based methods, instead of converting straightforward RSSI to distances, several novel ideas based on connectivity make a comparison between the received signal strength and a power threshold leading to distances constraints. Combinations of such constraints have been made diversely, such as using Sequential Monte-Carlo algorithm [Hu *et Evans* 2004; Baggio *et Langendoen* 2006], interval analysis [Mourad *et al.* 2008; 2009], variational filter [Teng *et al.* 2010], etc. Another novel idea is based on collecting scenario information [Tapus *et Siegwart* 2006; Robles *et al.* 2010] which also make comparison between signal strengths but in a different whole-area-search and pattern matching way. This approach is commonly known as fingerprinting method, using the information collected, which indicates gath-

ering the surrounding significant characteristics, like colors, edges or RSSI, to model the actual environment of features. With such considerations, fingerprinting methods are considered very prospective due to their high flexibility to various surroundings.

In case of mobile sensors in a WSN, fingerprinting methods are also adaptive to localize a mobile node since its dependence on actual ambient matters no matter what time instant it is. In fact, the mobility of mobile sensors introduces the difficulty to positioning since the movements of sensors are often unregulated and unpredictable, occurring to any direction or any displacement, especially when the sensors are not equipped with precise instrument for reading the relevant movement data. However, one could improve the localization accuracy by having some mobility information, such as certain attributes of movement, possibly constrained by sensor itself or environmental factors, like the maximum velocity that a mobile sensor could possibly have (patients walking speed), or an angle when a robot could turn around within a time period. Researches on the mobility models are made in [Camp *et al.* 2002] where entity mobility models and group mobility models are proposed offering choices for researchers to localize a mobile sensor. Some efforts regarding the mobile sensors are proposed with exploiting both the connectivity and mobility constraints, such as solutions by Monte-Carlo [Hu *et al.* 2004; Baggio *et al.* 2006], interval analysis [Mourad *et al.* 2008; 2009] and Hidden Markov [Fraile *et al.* 1998; Arthi *et al.* 2010]. The ranging information is also used with mobility to localize the nodes throughout mobility adaptive RSSI filter by [Chandra-Sekaran *et al.* 2009].

Sensors could also be equipped with additional devices for movements measurements. For instance, by having magnetometer, a varying magnetic fingerprinting similar to RSSI is constructed and solved alternatively by Monte-Carlo in [Haverinen *et al.* 2009]. Moreover, [Choi *et al.* 2011] localizes mobile robots by utilizing radio-frequency identification (RFID) tags [Ni *et al.* 2004] and ultrasonic sensors processing in a hierarchical scheme. If the installed device is an inertial measurement unit, it yields directly an estimation for the displacement of the mobile sensor, such as in [Hsu *et al.* 2009] which equips a robot with an accelerometer. Such a method offers accurate estimation if perfect measurements are presented. Otherwise, the device accelerometer is commonly considered to be especially susceptible to the interruption which leads probably to a rather rough localization in normal cases.

This chapter introduces a novel localization technique based on both fingerprints and accelerometer information, which compensates estimation error. The proposed method is an anchor-based method executed in a centralized scheme undertaking the fingerprints collection. It performs firstly a configuration phase, preceding the localization of the nodes. In the configuration phase, RSSI fingerprinting information are collected to model the network, leading to a power map. In the localization phase, a first position estimate is computed using the constructed power map. A second estimate is meanwhile computed using instantly measured accelerometer information. Both instant estimates are then combined using interval analysis [Jaulin *et al.* 2001; Mourad *et al.* 2009], leading to boxes including the actual positions of the nodes. Beside the basic idea, two alternative mobility models are equally proposed to apply on the second estimation,

III. CENTRALIZED LOCALIZATION

either economizing the time-consuming or refining the estimation accuracy. Concerning the first estimate, several different computations of weight factors are equally shown and discussed. Furthermore, the interval analysis technique is applied in various optional ways which will be illustrated in the end of the chapter.

III.2 Problem statement

This chapter proposes a centralized algorithm for sensors localization. It thus considers the centralized topology of WSNs, shown in Chapter I Figure I.2, having a single base-station. Sensors in this topology are uniquely capable of sending and receiving measurements, lacking of the capabilities of computation and processing. As a consequence, all of the sensed data are required to be transmitted to the central fusion station where all the computations are conducted. Such network could collect high quality measurements to provide high accuracy processing result. Nevertheless, the cost lies in that sensors digest much more energies, especially for the sensors nearby the central station, due to the fact that energy consumed in transferring one bit is higher than the energy needed to process one bit. Based on such a network, a centralized localization method is designed using fingerprinting technique and accelerometer measurements. In the following, the network components are first presented. The localization problem using either fingerprints or sensors' accelerations is then described. The proposed algorithm combining all the measurements is provided in Section III.3.

III.2.1 Configuration of the network

The proposed method is an anchor-based localization technique performing fingerprinting collection. Consider a surveillance area of \mathbb{R}^D , with $D = 2$ or $D = 3$ for a two-dimensional or a three-dimensional area. The network has two types of sensors: *anchors* and *nodes*, the former with known fixed positions denoted by $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,D})$, $i \in \{1, \dots, N_A\}$, and the latter having unknown positions, denoted by $\mathbf{x}_j(t) = (x_{j,1}(t), \dots, x_{j,D}(t))$, $j \in \{1, \dots, N_X\}$, N_A and N_X being the number of anchors and nodes in the network. The localization problem consists then on finding $\mathbf{x}_j(t)$, $j \in \{1, \dots, N_X\}$, using measurements and information related to the \mathbf{a}_i , $i \in \{1, \dots, N_A\}$. Moreover, being based on radio-fingerprints, the proposed method performs RSSI collection to define a power or RSSI map over the network. To do this, N_P uniformly deployed *reference positions* are generated over the whole network, denoted by $\mathbf{p}_k = (p_{k,1}, \dots, p_{k,D})$, $k \in \{1, \dots, N_P\}$. An example of such a configuration is shown in Figure III.1.

Once the basic network is built up, a collection of power information is performed at all these reference positions. To this end, anchors, denoted as circles in the plot, broadcast signals in the network with the same initial power ξ_0 . It is noted that their signal powers are inversely proportional to transmission distances, namely they decrease with the increase of the distance separating the sensors. According to this attribute and by putting test nodes at the reference positions, one is able to measure the Received Signal

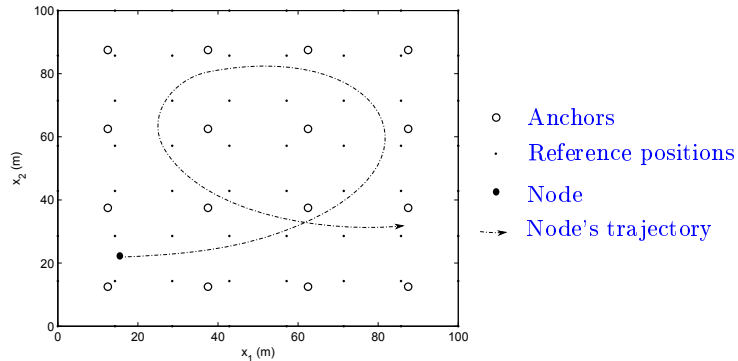


Figure III.1: A configuration of a centralized network

Strength Indicators (RSSI) of anchors signals received at these exact positions denoted as small points in the plot, constructing the reference RSSIs. Let ξ_{P_k} , $k \in \{1, \dots, N_P\}$ be the reference RSSI vectors, given by

$$\xi_{P_k} = (\xi_{P_k,1}, \dots, \xi_{P_k,N_A}), \quad k \in \{1, \dots, N_P\}, \quad (\text{III.1})$$

where $\xi_{P_k,i}$ is the strength of the signal emitted by the anchor i and received at the reference position \mathbf{p}_k . The configuration phase leads then to a fingerprints database, also called power map [Robles *et al.* 2010]. This database consists of N_P couples, each one denotes a couple of a reference position and its associated RSSI vector $(\mathbf{p}_k, \xi_{P_k})$, $k \in \{1, \dots, N_P\}$. The fingerprints database is stored at the central fusion station, also called computation center, to be used with the real-time measurements in order to perform the localization computation. This phase is thus performed offline, before the start of the localization process.

III.2.2 Localization using fingerprints database

During the online positioning procedure, nodes travel freely through the network and collect signals of all anchors. At every time-step t , each node j measures the strengths of its received signals, sent from all anchors. The real-time RSSI vectors is generated and denoted as follows,

$$\xi_{X_j}(t) = (\xi_{X_j,1}(t), \dots, \xi_{X_j,i}(t), \dots, \xi_{X_j,N_A}(t)), \quad j \in \{1, \dots, N_X\}, \quad (\text{III.2})$$

where notation $\xi_{X_j,i}(t)$ stands for the signal strength at reception of the signal emitted by the anchor i and received by the node j at time t . Then, in order to localize a node j , one looks for a pattern matching with the fingerprints database. To this end, the proposed method is based on the K -nearest neighbors algorithm [Navarro *et al.* 2010]. It thus computes for each node j the Euclidean distances $\|\xi_{X_j}(t), \xi_{P_k}\|$ between its instantaneously obtained vector $\xi_{X_j}(t)$ and all reference RSSI vectors ξ_{P_k} , $k \in \{1, \dots, N_P\}$. According to that, the K smallest distances are selected. Let $I_j(t)$ denote

III. CENTRALIZED LOCALIZATION

the set of indices of the reference positions yielding these K smallest power distances. Since signals RSSI are tightly related to geographic distances between sensors, then $I_j(t)$ could be considered as the set of indices of the K nearest reference neighbors to the node j at time t . Accordingly, the fingerprints based position estimate of node j is computed as in Eq. III.3,

$$\hat{\mathbf{x}}_{f,j}(t) = \sum_{k \in I_j(t)} \omega_{j,k}(t) \mathbf{p}_k, \quad (\text{III.3})$$

where $\omega_{j,k}(t)$ is the weight factor associated to the selected reference position k . The weights $\omega_{j,k}(t)$, $k \in I_j(t)$, are chosen in the way of inversely proportional to the Euclidean RSSI distances. Indeed, the smaller a distance between RSSI vectors is, the nearer the node is to the corresponding reference position, thus the greater the weight should be. Also, the weights should be normalized as $\sum_{k \in I_j(t)} \omega_{j,k}(t) = 1$. One could choose an inverse computation of weights as in Eq. III.4,

$$\omega_{j,k}(t) = \frac{\|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_k}\|^{-\alpha}}{\sum_{u \in I_j(t)} \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_u}\|^{-\alpha}}. \quad (\text{III.4})$$

where α is a fixed parameter of weight factor which could be chosen empirically. Another formula expressing the inverse relation existing between distances and RSSIs could also be used for the weights. Eq. III.5 below shows the computation of such weight factors, called exponential weights.

$$\omega_{j,k}(t) = \frac{\exp(-\alpha \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_k}\|)}{\sum_{u \in I_j(t)} \exp(-\alpha \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_u}\|)}. \quad (\text{III.5})$$

The exponential weights in fact amplify the difference among all computed distances. The adaptivity of both weights computations will be discussed later. Note that the computation in this section is performed using exact measured data, and thus it does not take the measurement incertitude into consideration. In the Section III.3, the computation including measurements incertitude, as well as combination to accelerometer measurements are presented.

III.2.3 Localization using accelerations

Each node j is assumed to be equipped with an accelerometer device [Hsu et Yu 2009], yielding, at each time-step t , its acceleration values over all the coordinates, denoted by $\gamma_{j,1}(t), \dots, \gamma_{j,D}(t)$. Based only on this information, node j computes another estimate of its position as in Eq. III.6 using a second-order mobility model,

$$\hat{\mathbf{x}}_{\gamma,j}(t) = \hat{\mathbf{x}}_{\gamma,j}(t-1) + \boldsymbol{\nu}_j(t-1)\Delta t + \frac{1}{2}\boldsymbol{\gamma}_j(t)\Delta t^2, \quad (\text{III.6})$$

where $\boldsymbol{\gamma}_j(t) = (\gamma_{j,1}(t), \dots, \gamma_{j,D}(t))$ is the acceleration vector of node j at time t , $\boldsymbol{\nu}_j(t)$ is the node's velocity vector, Δt is the time interval between two consecutive time-steps and $\hat{\mathbf{x}}_{\gamma,j}(t-1)$ denotes the previous position estimate of node j obtained at time $t-1$

III.3 Centralized localization algorithm

while using only accelerations. Velocities $\boldsymbol{\nu}_j(t)$ are also computed at each time-step using accelerations as in Eq. III.7,

$$\boldsymbol{\nu}_j(t) = \boldsymbol{\nu}_j(t-1) + \boldsymbol{\gamma}_j(t)\Delta t. \quad (\text{III.7})$$

Note that the model of Eq. III.6 and Eq. III.7 assumes that the accelerations are constant between time-step $t-1$ and time-step t . With slightly varying curves of accelerations, this scheme leads to accurate estimates. However, with abrupt changes in accelerations or much noisy interrupted accelerations, estimates $\hat{\boldsymbol{x}}_{\gamma,j}(t)$ might be sensitively deviated from the exact trajectory, due to cumulative errors from $\boldsymbol{\gamma}_j(t)$ over time accumulation. One first-order more simplified model is an optional choice to compute acceleration-based estimates. According to this model, instantaneous velocities are first computed using Eq. III.7. Then, the position estimates are given by Eq. III.8,

$$\hat{\boldsymbol{x}}_{\gamma,j}(t) = \hat{\boldsymbol{x}}_{\gamma,j}(t-1) + \boldsymbol{\nu}_j(t)\Delta t. \quad (\text{III.8})$$

Here, accelerations are first assumed to be constant between time-steps $t-1$ and t to compute velocities at time t . Then, an additional approximation is performed by assuming that the velocities are constant between time-steps $t-1$ and t . Clearly, this model is less accurate, but runs faster than the first one. In some applications, it is more efficient since the estimation error stays within an acceptable range.

Having the values of accelerations at time-steps $t-1$ and t , one could propose a better third-order model taking the variation of the accelerations into consideration. Indeed, such a model is more suitable to trajectories where the measured accelerations have abrupt changes. The position estimates are formulated by the third-order model as in Eq. III.9,

$$\hat{\boldsymbol{x}}_{\gamma,j}(t) = \hat{\boldsymbol{x}}_{\gamma,j}(t-1) + \boldsymbol{\nu}_j(t-1)\Delta t + \frac{1}{2}\boldsymbol{\gamma}_j(t-1)\Delta t^2 + \frac{1}{6}\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}\Delta t^3, \quad (\text{III.9})$$

where $\boldsymbol{\nu}_j(t-1)$ is the velocity vector of the node j at time-step $t-1$ computed at iteration t as Eq. III.10,

$$\boldsymbol{\nu}_j(t) = \boldsymbol{\nu}_j(t-1) + \boldsymbol{\gamma}_j(t-1)\Delta t + \frac{1}{2}\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}\Delta t^2. \quad (\text{III.10})$$

It is worth noting that this model assumes that, between two consecutive time-steps $t-1$ and t , the accelerations are linear going from $\boldsymbol{\gamma}_j(t-1)$ to $\boldsymbol{\gamma}_j(t)$, with a slope of $\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}$. With a small value of Δt , this model works very well, since the approximated acceleration curves becomes very close to the real ones. In the following, the combination of both estimations using fingerprints and accelerations, as well as measurements incertitude, is presented.

III.3 Centralized localization algorithm

Nodes collect RSSI information and measure their accelerations while moving in the network. Under such consideration, the real-time measured RSSI are used with the

III. CENTRALIZED LOCALIZATION

fingerprints database to compute first estimates of nodes positions, whereas measured accelerations meanwhile are used to compute second positions estimates. The combination of the first, fingerprints-based, and the second, accelerations-based, estimates, is performed in the interval framework [Jaulin *et al.* 2001; Mourad *et al.* 2009]. Based on interval analysis, the combination employs interval information which could include all incertitude over measurements and computations data. Practically, instead of taking the exact estimated positions computed above, one could take the incertitude over measurements and computations into consideration leading to estimated boxes including the actual positions of the nodes inside. In the following, the computations using fingerprints and accelerations shown above are reformulated in the interval framework.

III.3.1 Reformulation over accelerations

Let $\pm\delta_\gamma$ be the incertitude over the measured acceleration values, then one could define acceleration boxes (or interval vectors) as follows.

$$[\gamma_j](t) = [\gamma_{j,1}](t) \times \cdots \times [\gamma_{j,D}](t) = [\gamma_{j,1}(t) - \delta_\gamma, \gamma_{j,1}(t) + \delta_\gamma] \times \cdots \times [\gamma_{j,D}(t) - \delta_\gamma, \gamma_{j,D}(t) + \delta_\gamma]. \quad (\text{III.11})$$

$[\gamma_j](t)$ is a D -dimensional box, assumed to include the actual value of the accelerations vector of node j at time t with extension down to lower boundary $\gamma_j(t) - \delta_\gamma \mathbf{1}$ and up to upper boundary $\gamma_j(t) + \delta_\gamma \mathbf{1}$ where $\mathbf{1}$ is a vector of D ones.

The quantity δ_γ is taken as the measurement incertitude defined by the manufacturer of the used accelerometer device. Alternatively and more practically, it could also be defined by a prior calibration of the accelerometer, before the localization procedure, where it would be set to its maximal measurement error. More details for computation of δ_γ are given in Remark 1 below. Under such assumption, it is possible to rewrite the acceleration-based position estimates of the second-order model of Eq. III.6 using $[\gamma_j](t)$ in the interval framework as follows,

$$[\hat{\mathbf{x}}_j]_\gamma(t) = [\hat{\mathbf{x}}_j](t-1) + [\boldsymbol{\nu}_j](t-1)\Delta t + \frac{1}{2}[\gamma_j](t)\Delta t^2, \quad (\text{III.12})$$

where the previous acceleration-based position estimate of node j is replaced by the previous final estimated box $[\hat{\mathbf{x}}_j](t-1)$, obtained after the combination with the fingerprinting estimate. $[\boldsymbol{\nu}_j](t)$ is its boxed velocity, computed as follows,

$$[\boldsymbol{\nu}_j](t) = [\boldsymbol{\nu}_j](t-1) + [\gamma_j](t)\Delta t. \quad (\text{III.13})$$

It is also possible to rewrite the first-order simplified model as follows,

$$[\hat{\mathbf{x}}_j]_\gamma(t) = [\hat{\mathbf{x}}_j](t-1) + [\boldsymbol{\nu}_j](t)\Delta t, \quad (\text{III.14})$$

where velocities boxes are first updated using Eq. III.13.

Similarly for the third-order mobility model, nodes movements are reformulated into interval framework as follows,

$$[\hat{\mathbf{x}}_j]_\gamma(t) = [\hat{\mathbf{x}}_j](t-1) + [\boldsymbol{\nu}_j](t-1)\Delta t + \frac{1}{2}[\gamma_j](t-1)\Delta t^2 + \frac{1}{6} \frac{[\gamma_j](t) - [\gamma_j](t-1)}{\Delta t} \Delta t^3, \quad (\text{III.15})$$

where $\boldsymbol{\nu}_j(t)$ is equally boxed by

$$[\boldsymbol{\nu}_j](t) = [\boldsymbol{\nu}_j](t-1) + [\boldsymbol{\gamma}_j](t-1)\Delta t + \frac{1}{2} \frac{[\boldsymbol{\gamma}_j](t) - [\boldsymbol{\gamma}_j](t-1)}{\Delta t} \Delta t^2. \quad (\text{III.16})$$

Remark 1 Computation of δ_γ

The reliability of final estimation relies on the uncertainties introduced to the measurements, while applying the interval computations. Applying the interval analysis, it is hoped that all the incertitude is included inside the constructed measurements boxes. The quantity δ_γ is either considered to be the measurement incertitude defined by the instrument manufacturer, or determined by an anticipated calibration procedure of the accelerometer. Considering the impacts possibly generated at surroundings, the latter is supposed to be more accurate and comprehensive. In other words, one could carry out a preliminary measurement over the uncertainty of accelerations in specific scenario. By comparing with the observed acceleration values, it is possible to have a set of acceleration errors values. Let $\tilde{\gamma}_{j,\ell}(t)$, $\ell \in \{1, \dots, D\}$ be the correct noiseless accelerations of node j among the ℓ -th coordinate at time-step t , and let $\gamma_{j,\ell}(t)$ be its corresponding measured noisy value. Then,

$$\gamma_{j,\ell}(t) = \tilde{\gamma}_{j,\ell}(t) + \varepsilon_{\gamma,j,\ell}(t), \quad \ell \in \{1, \dots, D\}, \quad (\text{III.17})$$

where $\varepsilon_{\gamma,j,\ell}(t)$ is the additional acceleration noise. To get the parameters of $\varepsilon_{\gamma,j,\ell}(t)$, several values could be obtained by performing the anticipated calibration phase of the accelerometer as said before. Consider that $\varepsilon_{\gamma,j,\ell}(t)$ is a zero-mean gaussian noise, with a standard deviation denoted by σ_γ . Accordingly, the incertitude δ_γ could be taken to $3\sigma_\gamma$. It is possible to justify this choice by computing the probability of including the real noiseless acceleration $\tilde{\gamma}_{j,\ell}(t)$ into the acceleration interval to be considered $[\gamma_{j,\ell}(t)] = [\gamma_{j,\ell}(t) - \delta_\gamma, \gamma_{j,\ell}(t) + \delta_\gamma]$ as follows,

$$\begin{aligned} \Pr(\tilde{\gamma}_{j,\ell}(t) \in [\gamma_{j,\ell}(t)]) &= \Pr(\gamma_{j,\ell}(t) - \delta_\gamma \leq \tilde{\gamma}_{j,\ell}(t) \leq \gamma_{j,\ell}(t) + \delta_\gamma) \\ &= \Pr(-\delta_\gamma \leq \varepsilon_{\gamma,j,\ell}(t) \leq \delta_\gamma) \\ &= \text{erf}\left(\frac{\delta_\gamma}{\sigma_\gamma \sqrt{2}}\right) \end{aligned} \quad (\text{III.18})$$

where $\text{erf}(\cdot)$ is the Gauss-error function. With $\delta_\gamma = 3\sigma_\gamma$, $\Pr(\tilde{\gamma}_{j,\ell}(t) \in [\gamma_{j,\ell}(t)]) = 0.997$, that is a probability of 99.7% ensuring the real acceleration is included within the considered interval. One could also take δ_γ to the maximal acceleration error value. If the error has not a zero mean then one could set an acceleration bias equal to the error mean and then update the measured acceleration values by subtracting this bias in the way to have a zero-mean error.

Remark 2 Update of velocity boxes

While the RSSI-based estimation uses only information of current time, the acceleration-based estimation employs previous interval information, leading to incertitude accumulation. Indeed, models of Eq. III.12, Eq. III.14 and Eq. III.15 use boxed velocities

III. CENTRALIZED LOCALIZATION

$[\boldsymbol{\nu}_j](t)$ and $[\hat{\boldsymbol{x}}_j](t)$ where all previous uncertainty values are considered. While final boxes $[\hat{\boldsymbol{x}}_j](t)$ have variable bit widths, velocity boxes $[\boldsymbol{\nu}_j]$ grow with time, leading to larger acceleration-based position boxes from one time-step to the other. In order to reduce the impact of uncertainty accumulation, one could correct the velocity boxes while using the second-order model after computing $[\hat{\boldsymbol{x}}_j](t)$ as follows,

$$[\hat{\boldsymbol{\nu}}_j](t-1) = [\boldsymbol{\nu}_j](t-1) \cap \frac{[\hat{\boldsymbol{x}}_j](t) - [\hat{\boldsymbol{x}}_j](t-1) - \frac{1}{2}[\boldsymbol{\gamma}_j](t)\Delta t^2}{\Delta t}. \quad (\text{III.19})$$

The corrected velocities substitute previous velocities in Eq. III.13 to compute velocities of time t . Similarly with the third-order Eq. III.15, the velocity boxes are corrected as follows,

$$[\hat{\boldsymbol{\nu}}_j](t-1) = [\boldsymbol{\nu}_j](t-1) \cap \frac{[\hat{\boldsymbol{x}}_j](t) - [\hat{\boldsymbol{x}}_j](t-1) - \frac{1}{2}[\boldsymbol{\gamma}_j](t-1)\Delta t^2 - \frac{1}{6}\frac{[\boldsymbol{\gamma}_j](t) - [\boldsymbol{\gamma}_j](t-1)}{\Delta t}\Delta t^3}{\Delta t}. \quad (\text{III.20})$$

For the first-order simplified model, the velocity boxes are corrected after computing $[\hat{\boldsymbol{x}}_j](t)$ as follows,

$$[\hat{\boldsymbol{\nu}}_j](t) = [\boldsymbol{\nu}_j](t) \cap \frac{[\hat{\boldsymbol{x}}_j](t) - [\hat{\boldsymbol{x}}_j](t-1)}{\Delta t}. \quad (\text{III.21})$$

In this case, the corrected velocities are used in the estimation of nodes positions in the following time-step.

Remark 3 Acceleration-based uncertainty $\delta_{x_\gamma}(t)$ over estimates

By observing Eq. III.11-III.16, one can see that the uncertainty introduced to accelerations is in fact inevitably accumulated to the upper computation until the final displacement as time goes on, especially when acting over the highest order computations such as in Eq. III.15. This means that the acceleration-based position boxes would be wider as time goes on. By correcting the velocities intervals as in Remark 2, one might possibly reduce the accumulated uncertainty. On the other hand, if we refer to Chapter II at Section II.3.1.4, there exist different ways to represent the uncertainty, some more optimal than the others. The key idea consists here of computing the uncertainty over the acceleration estimated position in a different manner, to check later whether this solution is more accurate or not. Let the equations of position estimation using accelerations Eq. III.6, Eq. III.8 and Eq. III.9 be rewritten as follows,

$$\hat{x}_{\gamma,j,\ell}(t) = \hat{x}_{\gamma,j,\ell}(t-1) + B_{j,\ell}(t), \quad (\text{III.22})$$

where ℓ is the coordinate index having values in $\{1, \dots, D\}$ and $B_{j,\ell}(t) = \nu_{j,\ell}(t)\Delta t$ for the 1st-order model, $B_{j,\ell}(t) = \nu_{j,\ell}(t-1)\Delta t + \frac{1}{2}\gamma_{j,\ell}(t)\Delta t^2$ for the 2nd-order model or $B_{j,\ell}(t) = \nu_{j,\ell}(t-1)\Delta t + \frac{1}{2}\gamma_{j,\ell}(t-1)\Delta t^2 + \frac{1}{6}\frac{\gamma_{j,\ell}(t) - \gamma_{j,\ell}(t-1)}{\Delta t}\Delta t^3$ for the 3rd-order model, the velocities $\nu_{j,\ell}(t)$ being computed according to Eq. III.7 for the 1st-order and the 2nd-order models and Eq. III.10 for the 3rd-order model. When applying the interval computations, the acceleration-based estimation equation would be given by

$$[\hat{x}_{j,\ell}]_\gamma(t) = [\hat{x}_{j,\ell}](t-1) + [B_{j,\ell}](t), \quad (\text{III.23})$$

III.3 Centralized localization algorithm

where $[B_{j,\ell}](t) = [B_{j,\ell}(t) - \delta_{x_\gamma}(t), B_{j,\ell}(t) + \delta_{x_\gamma}(t)]$. The quantity $B_{j,\ell}(t)$ remains scalar and $\delta_{x_\gamma}(t)$ is the incertitude coming from the acceleration measurements to be computed in this section to cover the corresponding model incertitude, in a different manner than before.

To compute $\delta_{x_\gamma}(t)$, let us start with the standard deviation of the velocity error. Having σ_γ the standard deviation of the noise contained in the acceleration measures, it is possible to compute the standard deviation of the velocity noise σ_ν . Consider the first-order or the second-order mobility models, with Eq. III.7, then σ_ν would be function of the time-step t as follows,

$$\begin{cases} \sigma_\nu^2(1) = \sigma_\gamma^2 \Delta t^2, \\ \sigma_\nu^2(2) = \sigma_\nu^2(1) + \sigma_\gamma^2 \Delta t^2 = 2\sigma_\gamma^2 \Delta t^2, \\ \sigma_\nu^2(3) = \sigma_\nu^2(2) + \sigma_\gamma^2 \Delta t^2 = 3\sigma_\gamma^2 \Delta t^2, \\ \vdots \\ \sigma_\nu^2(t) = \sigma_\nu^2(t-1) + \sigma_\gamma^2 \Delta t^2 = t\sigma_\gamma^2 \Delta t^2, \end{cases} \quad (\text{III.24})$$

where the initial velocity is null with no noise. This leads to

$$\sigma_\nu(t) = \sqrt{t}\sigma_\gamma \Delta t. \quad (\text{III.25})$$

Now if we consider the position estimation using the first-order model, the noise over the additive quantity $B_{j,\ell}(t)$ would result from the current time velocity noise and thus its standard deviation denoted by $\sigma_{x_\gamma}(t)$ is equal to $\sigma_{x_\gamma}(t) = \sigma_\nu(t)\Delta t = \sqrt{t}\sigma_\gamma \Delta t^2$. In terms of incertitude, the incertitude over $B_{j,\ell}(t)$ would then be induced by the velocity term incertitude and thus

$$\delta_{x_\gamma}(t) = \sqrt{t}\delta_\gamma \Delta t^2, \quad (\text{III.26})$$

similarly to the expression of $\sigma_{x_\gamma}(t)$.

If the second-order model is considered, the incertitude over the $B_{j,\ell}(t)$ term is induced by the sum of the previous velocity and the current acceleration obtained errors, whose variance $\sigma_{x_\gamma}^2(t) = \sigma_\nu^2(t-1)\Delta t^2 + \frac{1}{4}\sigma_\gamma^2 \Delta t^4 = (t - \frac{3}{4})\sigma_\gamma^2 \Delta t^4$, in other words $\sigma_{x_\gamma}(t) = \sqrt{t - \frac{3}{4}}\sigma_\gamma \Delta t^2$. Here the incertitude $\delta_{x_\gamma}(t)$ is given by

$$\delta_{x_\gamma}(t) = \sqrt{t - \frac{3}{4}}\delta_\gamma \Delta t^2, \quad (\text{III.27})$$

similarly to the expression of $\sigma_{x_\gamma}(t)$.

Consider now the third-order model. The expression of $B_{j,\ell}(t)$ could be rewritten as follows,

$$B_{j,\ell}(t) = \nu_{j,\ell}(t-1)\Delta t + \frac{1}{3}\gamma_{j,\ell}(t-1)\Delta t^2 + \frac{1}{6}\gamma_{j,\ell}(t)\Delta t^2. \quad (\text{III.28})$$

Also the velocity expression of Eq. III.10 could be reformulated as follows,

$$\nu_{j,\ell}(t) = \nu_{j,\ell}(t-1) + \frac{1}{2}\gamma_{j,\ell}(t-1)\Delta t + \frac{1}{2}\gamma_{j,\ell}(t)\Delta t. \quad (\text{III.29})$$

III. CENTRALIZED LOCALIZATION

Accordingly, the standard deviation of the velocities error for the 3rd-order model $\sigma_\nu(t)$ is also computed as follows,

$$\begin{cases} \sigma_\nu^2(1) = \frac{1}{2}\sigma_\gamma^2\Delta t^2, \\ \sigma_\nu^2(2) = \sigma_\nu^2(1) + \frac{1}{2}\sigma_\gamma^2\Delta t^2 = \sigma_\gamma^2\Delta t^2, \\ \sigma_\nu^2(3) = \sigma_\nu^2(2) + \frac{1}{2}\sigma_\gamma^2\Delta t^2 = \frac{3}{2}\sigma_\gamma^2\Delta t^2, \\ \vdots \\ \sigma_\nu^2(t) = \sigma_\nu^2(t-1) + \frac{1}{2}\sigma_\gamma^2\Delta t^2 = \frac{t}{2}\sigma_\gamma^2\Delta t^2. \end{cases} \quad (\text{III.30})$$

This leads to

$$\sigma_\nu(t) = \sqrt{\frac{t}{2}}\sigma_\gamma\Delta t. \quad (\text{III.31})$$

Here the incertitude of $B_{j,\ell}(t)$ is induced by the sum of the previous velocity and the previous and current accelerations obtained errors, whose variance $\sigma_{x_\gamma}^2(t) = \sigma_\nu^2(t-1)\Delta t^2 + \frac{1}{9}\sigma_\gamma^2\Delta t^4 + \frac{1}{36}\sigma_\gamma^2\Delta t^4 = (\frac{t}{2} - \frac{13}{36})\sigma_\gamma^2\Delta t^4$, in other words $\sigma_{x_\gamma}(t) = \sqrt{\frac{t}{2} - \frac{13}{36}}\sigma_\gamma\Delta t^2$. The incertitude $\delta_{x_\gamma}(t)$ is given here by

$$\delta_{x_\gamma}(t) = \sqrt{\frac{t}{2} - \frac{13}{36}}\delta_\gamma\Delta t^2, \quad (\text{III.32})$$

similarly to the expression of $\sigma_{x_\gamma}(t)$.

Having the expression of $\delta_{x_\gamma}(t)$ for each model, one could then compute $B_{j,\ell}(t)$ and update the previous position estimate using Eq. III.23 to have the acceleration-based boxed position estimate, instead of using Eq. III.12, Eq. III.14 or Eq. III.15. The performances of the proposed method using the new expression of the estimated position boxes would be compared to its other version using δ_γ in the simulations section, to check whether this new version leads to more estimation accuracy or not.

III.3.2 Reformulation over fingerprints

With the purpose of rewriting the fingerprints-based position estimates in the interval framework, an incertitude $\pm\delta_\xi$ is introduced over the measured RSSI values similarly to accelerations. This quantity could be determined by performing several measurements within the surveillance area preceding the localization process. Indeed, consider two sensors exchanging information in the network with the same initial power. For a given distance d separating the sensors, a fixed RSSI value is deemed to be obtained in a perfect environment, which is not the case in real environments resulting in the measurements incertitude. Thus, one can place the sensors at different distances, in different ways for each distance, and then measure the RSSI of exchanged signals. Let $[\xi]_d$ be the RSSI interval including all values of RSSI associated to a given distance d . The incertitude over RSSI then corresponds to the maximal gap between the RSSI per distance over all distances. That is to say, δ_ξ is set to be the maximal width of $[\xi]_d$ over all distances.

Based on this idea, suppose $[\xi_{Xj}](t)$ be the boxed RSSI vector of node j at time t , $[\xi_{Pk}]$ be the one associated to the reference point k . Then, $[\xi_{Xj}](t) = [\xi_{Xj,1}](t) \times \cdots \times$

III.3 Centralized localization algorithm

$[\xi_{X_j, N_A}](t)$ and $[\xi_{P_k}] = [\xi_{P_{k,1}}] \times \cdots \times [\xi_{P_{k, N_A}}]$ are actually obtained by introducing δ_ξ as in Eq. III.33,

$$\begin{aligned} [\xi_{X_j, i}](t) &= [\xi_{X_j, i}(t) - \delta_\xi, \xi_{X_j, i}(t) + \delta_\xi], \\ [\xi_{P_{k, i}}] &= [\xi_{P_{k, i}} - \delta_\xi, \xi_{P_{k, i}} + \delta_\xi], \end{aligned} \quad (\text{III.33})$$

where $i \in \{1, \dots, N_A\}$. Dealing with the uncertain RSSI values, the fingerprints-based position estimation is reformulated into box as follows,

$$[\hat{\mathbf{x}}_j]_f(t) = \sum_{k \in I_j(t)} [\omega_{j, k}](t) \mathbf{p}_k, \quad (\text{III.34})$$

where $[\omega_{j, k}](t)$ is the interval weight associated to the reference position k , in form of Eq. III.35 and Eq. III.36 according to the two definitions of inverting distances as shown in Eq. III.4 and exponential scheme as in Eq. III.5 given as follows,

$$[\omega_{j, k}](t) = \frac{\|[\xi_{X_j}](t), [\xi_{P_k}]\|^{-\alpha}}{\sum_{u \in I_j(t)} \|[\xi_{X_j}](t), [\xi_{P_u}]\|^{-\alpha}}. \quad (\text{III.35})$$

and,

$$[\omega_{j, k}](t) = \frac{\exp(-\alpha \|[\xi_{X_j}](t), [\xi_{P_k}]\|)}{\sum_{u \in I_j(t)} \exp(-\alpha \|[\xi_{X_j}](t), [\xi_{P_u}]\|)}. \quad (\text{III.36})$$

These quantities could be easily computed using interval tools as presented in Chapter II. The obtained boxes $[\hat{\mathbf{x}}_j]_f(t)$ are assumed to geographically cover the real positions of the nodes.

An alternative way to obtain the uncertain box $[\hat{\mathbf{x}}_j]_f(t)$ is to introduce another quantity of uncertainty $\pm \delta_{x_\xi}$ to the final estimate $\hat{\mathbf{x}}_{f, j}(t)$ by fingerprints-based method as shown in Eq. III.3. Indeed, it considers the uncertainty of localization over the network instead of the uncertainty of RSSI defined at the very beginning of the configuration. The quantity of $\pm \delta_{x_\xi}$ could be determined by executing a prior procedure of localization. To do this, generate a set of test positions over the surveillance area, measure their RSSI and localize them using the nearest neighbor algorithm. Estimation errors could then be obtained with the known test positions. The incertitude over the position estimation corresponds then to the maximal uncertainty in locating the test positions. That is to say, $\pm \delta_{x_\xi}$ is set to be the maximum value existing in the error set over all test positions. Consequently, given the position estimate $\hat{\mathbf{x}}_{f, j}(t)$ of node j at time-step t , an estimated box could be formulated as in Eq. III.37,

$$[\hat{\mathbf{x}}_j]_f(t) = [\hat{x}_{f, j, 1}(t) - \delta_{x_\xi}, \hat{x}_{f, j, 1}(t) + \delta_{x_\xi}] \times \cdots \times [\hat{x}_{f, j, D}(t) - \delta_{x_\xi}, \hat{x}_{f, j, D}(t) + \delta_{x_\xi}]. \quad (\text{III.37})$$

The two versions of the method will be compared in the simulations section to select the most accurate one.

III. CENTRALIZED LOCALIZATION

III.3.3 Combination step

Once both boxed estimates are computed for a given node j at time-step t , the final estimated box is given by intersecting both estimates as follows,

$$[\hat{\mathbf{x}}_j](t) = [\hat{\mathbf{x}}_j]_f(t) \cap [\hat{\mathbf{x}}_j]_\gamma(t). \quad (\text{III.38})$$

The center of $[\hat{\mathbf{x}}_j](t)$ yields the desired punctual position estimate of node j at time t . In case of empty intersection, there exist also two choices. The fingerprints-based box $[\hat{\mathbf{x}}_j]_f(t)$ could be chosen, since the fingerprints-based estimation is more reliable than the acceleration-based one, the latter making more assumptions to set its model. Otherwise, take the union \sqcup of these non-overlapped boxes to form a larger box which includes both possibilities inside. The design of the proposed interval based method is visualized in Figure III.2. It denotes the reference positions in black points, the anchors positions in black circles, the fingerprinting estimated box in light gray line, the acceleration estimated box in dark gray dashed line, the final estimated box in black line and in stars their corresponding estimated exact positions. By intersecting the estimated boxes, one obtains a smaller box with less incertitude, leading to more accurate position estimates.

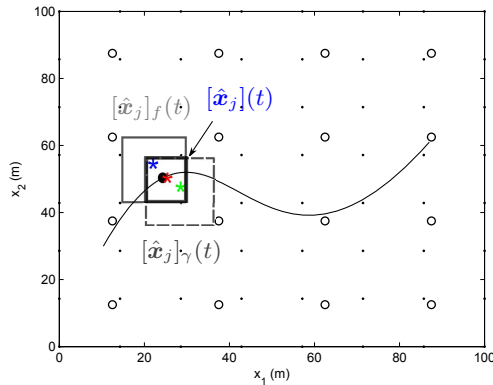


Figure III.2: Interval-based localization

III.4 Analysis and simulations

The performance of the proposed method is illustrated in detail and in all directions throughout the following simulation part. To this end, consider a surveillance area extending over a two-dimensional plan of dimensions $100m \times 100m$, where anchors and reference positions are both uniformly deployed. For simplicity and without loss of generality, consider the localization for only one mobile node, equipped with accelerometer yielding relevant readings, and thus, the j index will be withdrawn when needed in the following. This is possible in that each node localizes itself using only

anchors information, independently from other nodes on measurements and time-steps. It is worth noting that all simulations are implemented on Matlab platform of version 7.11.2(R2010B).

In what follows, the performance is put forward to be analyzed by all relevant parameters impacting the localization process. Particularly, optional decisions on the uncertainties δ_γ and δ_ξ are discussed at the first hand. An illustration of the method performances with different parameter values is also given. A comparison is made between the proposed method and a connectivity-based method in the last part.

III.4.1 Generation of simulated data

In order to simulate a node movement trajectory, the acceleration signals are firstly generated over the *1st* and the *2nd* coordinates varying over a period of 100s with time interval $\Delta t = 1s$ using the sine function. The trajectory is then obtained by integrating twice the acceleration functions at each axis and setting the initial velocities to null, meaning that the node is assumed to be fixed at a known position at the beginning of the localization. In fact, various trajectories are taken into account in the simulations. An example of acceleration signals is shown in Figure III.3, the corresponding trajectory of the node being shown in Figure III.4. The plot also shows 16 anchors and 100 reference positions uniformly deployed over the surveillance area.

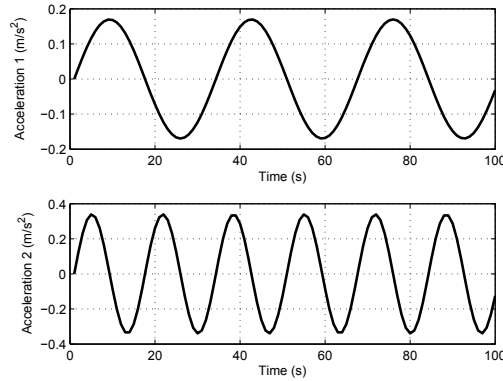


Figure III.3: An example of acceleration signals

In order to generate RSSI values, a pathloss model [Medeisis et Kajackas 2000] is employed here. According to it, distances separating the anchors from the reference positions and the node at all time-steps are required. Then, these distances and the pathloss model are used to reckon the RSSI fingerprints as follows,

$$\begin{aligned}\xi_{P_{k,i}} &= \xi_0 - 10n_P \log_{10} \|\mathbf{a}_i, \mathbf{p}_k\| + \varepsilon_{\xi,k,i}, \\ \xi_{X_i}(t) &= \xi_0 - 10n_P \log_{10} \|\mathbf{a}_i, \mathbf{x}(t)\| + \varepsilon_{\xi,i}(t),\end{aligned}$$

where the interruption considered here is presented in terms of noise obeying certain probability distribution. In this simulation, all RSSI are in dBm, ξ_0 is the initial transmission power in dBm, set to 100dBm, n_P is the pathloss parameter, set to 4. $\varepsilon_{\xi,k,i}$ and

III. CENTRALIZED LOCALIZATION

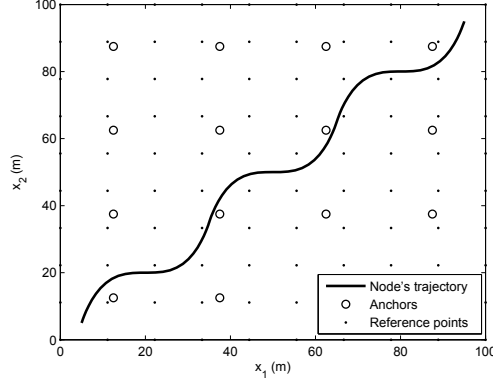


Figure III.4: An example of the node's trajectory

$\varepsilon_{\xi,i}(t)$ are zero-mean gaussian noises added to the original signals, with standard deviations denoted by σ_{ξ} . A zero-mean gaussian noise $\varepsilon_{\gamma,\ell}(t)$ is also added to acceleration values, with a standard deviation denoted by σ_{γ} :

$$\gamma_{\ell}(t) = \tilde{\gamma}_{\ell}(t) + \varepsilon_{\gamma,\ell}(t), \quad \ell \in \{1, 2\},$$

where $\tilde{\gamma}_{\ell}(t)$ denotes the noiseless acceleration signal of coordinate ℓ . The resulting noisy data are used as input signals in the algorithm during the computation. The numbers of anchors N_A , reference points N_P and neighbors K , as well as standard deviations σ_{γ} and σ_{ξ} , will be varied in the following. Note that, noises appended to the RSSI and acceleration values could also have diverse distributions which will be discussed in the next chapter.

Accordingly, the incertitude quantity δ_{γ} is set to $3\sigma_{\gamma}$, as illustrated in Remark 1. Similarly, δ_{ξ} is set to $3\sigma_{\xi}$ in simulated data. In real environments where σ_{γ} and σ_{ξ} are unknown, δ_{γ} and δ_{ξ} are set empirically as shown in the description of the algorithm. Nevertheless, there also exist optional adaptive choices making reform of this procedure which will be discussed later.

III.4.2 Illustration of the proposed method

The proposed method is illustrated using the trajectory of Figure III.4 in this section. In this context, set the number of anchors N_A to 49, the number of reference points N_P to 196, the number of neighbors K to be considered to 3 and the standard deviations of noises σ_{γ} and σ_{ξ} to $0.004m.s^{-2}$ and $0.1dB$ respectively. Figure III.5 shows the estimated boxes in red using the proposed method. The blue and green boxes respectively denote the boxes obtained from fingerprinting and accelerometer. The plot shows as well the estimated positions and the real trajectory, with the anchors and the reference positions. Consider that the estimation error is the average distance between the estimated positions and the real ones. Table III.1 shows the estimation errors in meters obtained

using the three acceleration-based mobility estimation models. It also shows the estimation errors using single fingerprinting information or accelerations. The table shows the average computation times in milliseconds per time-step as well. The results show that the proposed method leads to more accurate estimates than the only fingerprinting or accelerometer based methods, at the cost of an increase of the computation time that remains however very small. It also reveals that the third-order acceleration-model is more accurate than the others, but also less time-saving. In the following, the impacts of the mobility models as well as RSSI parameters will be discussed in details.

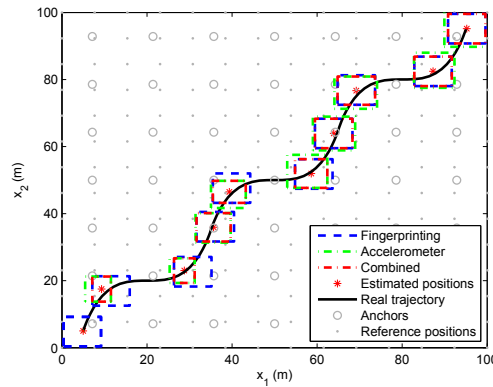


Figure III.5: Estimated boxes using the proposed method

Table III.1: Simulation results using trajectory 1

	Fingerprints	Accelerometer			Combined method		
		1st ord.	2nd ord.	3rd ord.	1st ord.	2nd ord.	3rd ord.
Error (m)	1.3873	2.8929	2.4772	2.1498	1.0358	1.0347	0.9995
Time (ms)	0.3118	0.0058	0.0060	0.0062	0.3472	0.3515	0.3594

III.4.3 Analysis of measurements incertitude

The accuracy of the localization process depends on the way the incertitude is considered while applying the interval computations. Indeed, with small incertitude values, the final estimated box could be empty or not including the correct node's position, while with high incertitude, the final box could be large with its center far away from the correct location, which leads to high estimation error. As depicted in section III.3.1 and section III.3.2, measured acceleration vector γ has an incertitude δ_γ away from the actual values, same as RSSI vectors ξ_P and ξ_X with an incertitude δ_ξ over measured RSSI. This section gives analysis and alternative predictions of the incertitude quantities to be taken in the interval computations.

III. CENTRALIZED LOCALIZATION

III.4.3.1 Accelerations incertitude

Two different ways to consider the accelerations incertitude are presented in the algorithm description. The first one introduces incertitude δ_γ over the accelerations, leading to acceleration boxes, velocity boxes and thus estimated position boxes. The second one consists of keeping scalar accelerations and velocities and then adding incertitude boxes employing the quantity δ_{x_γ} to the final computed boxes, as shown in Remark 3. In this section, the performances of the proposed method using either δ_γ in the first way or δ_{x_γ} are compared. To this end, six trajectories shown in Figure III.6 are considered with $N_A = 49$ anchors, $N_P = 100$ reference positions, $\sigma_\xi = 0.5dBm$, $\sigma_\gamma = 0.01m/s^2$, $K = 3$ neighbors and the first weights expression with $\alpha = 1$. The first three trajectories have slightly varying accelerations, whereas the trajectories 4 to 6 have more varying ones. The estimation errors of the only-fingerprinting based method, the only-acceleration based method with the 1st, the 2nd and the 3rd order models, and the combined proposed method with its δ_γ version and the δ_{x_γ} for the three models, are given in Table III.2 - Table III.7 with trajectories 1-6 respectively. All results are obtained as an average result of 50 implementations accompanying various noises with same standard deviations. The standard deviations σ_ξ and σ_γ of noises appended are equal to be around 2% – 5% of that of measurements $\xi_{X_j}(t)$, ξ_{P_k} and $\gamma(t)$. Parameters considered in the configuration are set casually here, nevertheless giving a reasonable estimation. A more detailed analysis on performance with respect to all these relevant parameters is illustrated in later part. The tables show also the areas of the boxes obtained with the combined method while using boxed accelerations, with δ_γ , or scalar accelerations with δ_{x_γ} . The estimation results show that the proposed method is more accurate while using the second version where incertitude is added using δ_{x_γ} at the position estimation. In addition to the smaller errors, this version leads to smaller boxes with less final estimation incertitude, and this for all the trajectories. Results also show that the 3rd order model in accelerations outperforms the other orders models for all the trajectories. This is due to the better approximation this model employs while depicting the node motion.

Consider now the velocity boxes modification shown in Remark 2, aiming at reducing more the incertitude and thus increasing the estimation accuracy. For this, we will consider the first version of the method using δ_γ leading to velocity boxes with the third-order mobility model. Table III.8 shows the estimation errors obtained using the method with and with velocity modification for all the trajectories. The table shows that the estimation error slightly varies with the velocity modification. In fact, this is due to the slight decrease of the estimated boxes areas, which does not decrease significantly the incertitude but moves the estimated position that is at the center of the box. With slight modification, the center could go away or come closer to the real positions but with no significant improvement in the estimation. It is obvious that the version of the method using δ_{x_γ} is much more accurate. For this reason, in the following sections, we will apply the version of the method with δ_{x_γ} for acceleration incertitude with the third-order mobility model.

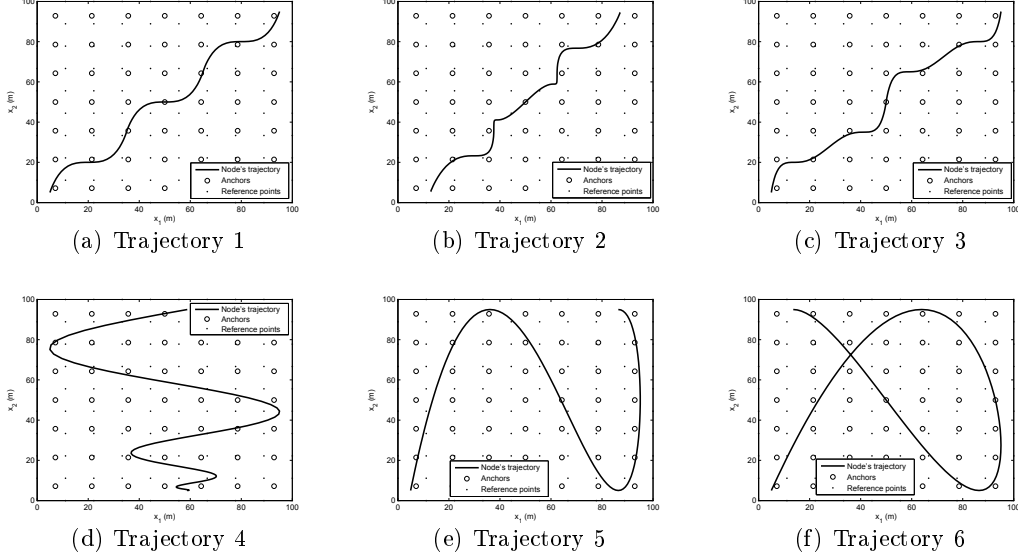


Figure III.6: Tested trajectories

III.4.3.2 Fingerprinting incertitude

The incertitude over the fingerprints algorithm is issued from the uncertainties of measured RSSI values influenced by surroundings. The interval box is employed helping to include such uncertainties inside. To determine the scale of boxes, incertitude must be first decided. It is more recommended that incertitude quantity will be decided empirically in certain scenarios since it heavily depends on the actual measured RSSI. Two possibilities for defining incertitude are presented in section III.3.2. The former works throughout the composition of RSSI $\xi_{X_j}(t)$ and ξ_{P_k} where $j \in \{1, \dots, N_X\}$, $k \in \{1, \dots, N_P\}$, however the latter works on an beforehand localization of test positions. Introducing quantity of incertitude δ_ξ to the measured RSSI values forms sequently the weights in terms of interval, as well as the interval estimations, as shown in Eq. III.35, Eq. III.36 and Eq. III.34. According to Eq. III.35, $\|[\xi_{X_j}](t), [\xi_{P_k}]\|$ computes the Euclidean distances between node j and reference position k by default. In an interval form, this is computed as:

$$\|[\xi_{X_j}](t), [\xi_{P_k}]\| = \sqrt{\sum_i ([\xi_{X_{j,i}}](t) - [\xi_{P_{k,i}}])^2}, \quad i \in 1, \dots, N_A. \quad (\text{III.39})$$

Here it could rise an computational challenge over the interval calculus with regarding to the "square" following the subtraction. Practically,

$$\begin{aligned} & [\xi_{X_{j,i}}](t) - [\xi_{P_{k,i}}] \\ &= [\text{center}([\xi_{X_{j,i}}](t)) - \text{center}([\xi_{P_{k,i}}]) - 2 * \delta_\xi, \text{center}([\xi_{X_{j,i}}](t)) - \text{center}([\xi_{P_{k,i}}]) + 2 * \delta_\xi]. \end{aligned}$$

III. CENTRALIZED LOCALIZATION

Table III.2: Accelerations incertitude - Estimation results using trajectory 1

Trajectory 1	Fingerprints	Accelerometer			Combined using δ_γ			Combined using δ_{x_γ}		
		1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.
Error (m)	2.1646	3.3453	3.0157	2.8854	1.6408	1.6072	1.6032	1.4259	1.0681	1.0591
Areas (m ²)	-	-	-	-	92.3933	91.8497	91.3933	50.9640	48.0413	34.7546

Table III.3: Accelerations incertitude - Estimation results using trajectory 2

Trajectory 2	Fingerprints	Accelerometer			Combined using δ_γ			Combined using δ_{x_γ}		
		1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.
Error (m)	1.8511	3.1313	2.8267	2.7198	1.5432	1.3929	1.2554	1.3498	1.0051	0.9720
Areas (m ²)	-	-	-	-	75.8825	73.9108	71.7928	47.7491	43.6521	34.9526

Table III.4: Accelerations incertitude - Estimation results using trajectory 3

Trajectory 3	Fingerprints	Accelerometer			Combined using δ_γ			Combined using δ_{x_γ}		
		1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.
Error (m)	2.0367	3.4681	3.1763	3.1387	1.5657	1.5075	1.4908	1.3184	1.0840	0.9038
Areas (m ²)	-	-	-	-	105.5747	105.5384	102.0451	57.5446	56.0571	42.1305

Table III.5: Accelerations incertitude - Estimation results using trajectory 4

Trajectory 4	Fingerprints	Accelerometer			Combined using δ_γ			Combined using δ_{x_γ}		
		1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.
Error (m)	1.7979	4.1974	3.4003	3.0990	1.5664	1.4023	1.3848	1.5310	1.2172	1.1186
Areas (m ²)	-	-	-	-	80.6611	72.7402	72.3717	32.0660	29.6008	27.2089

Table III.6: Accelerations incertitude - Estimation results using trajectory 5

Trajectory 5	Fingerprints	Accelerometer			Combined using δ_γ			Combined using δ_{x_γ}		
		1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.
Error (m)	2.0254	10.0993	10.0382	2.7566	1.6181	1.5154	1.4694	1.6098	1.5013	1.2935
Areas (m ²)	-	-	-	-	146.0893	127.8510	127.3997	70.0977	63.5796	48.6321

Table III.7: Accelerations incertitude - Estimation results using trajectory 6

Trajectory 6	Fingerprints	Accelerometer			Combined using δ_γ			Combined using δ_{x_γ}		
		1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.	1st-ord.	2nd-ord.	3rd-ord.
Error (m)	2.0911	10.6784	10.4047	3.0454	1.6733	1.5824	1.5516	1.6212	1.5412	1.5349
Areas (m ²)	-	-	-	-	158.1260	138.4202	137.8966	60.8952	59.6077	40.8662

Table III.8: Estimation errors using trajectories 1 to 6 with velocities modification

Error (m)	Trajectory 1	Trajectory 2	Trajectory 3	Trajectory 4	Trajectory 5	Trajectory 6
Without vel. modif.	1.6032	1.2554	1.4908	1.3848	1.4694	1.5516
With vel. modif.	1.5919	1.2839	1.5109	1.3826	1.4123	1.5355

In the case of two intervals situating very close, namely they are in fact owning certain overlapped region when $center([\xi_{X,j,i}](t)) - center([\xi_{P,k,i}]) < 2 * \delta_\epsilon$, the resulting interval

by subtraction includes zero inside. When it is required to calculate the Euclidean distance, the square operation for an interval including zero inside will influence the real interval structure, such as the displacement of interval center. In this situation, it only needs to consider the center of the computed interval, leading to an interval whose lower bound equals to its upper bound. Then, the Euclidean distance is computed by Eq. III.39. Alternatively, instead of using 2-norm (Euclidean distance), use 1-norm (Manhattan distance) and reformulate it as Eq. III.40.

$$\|[\xi_{Xj}](t), [\xi_{Pk}]\|_1 = \sum_i \|[\xi_{Xj,i}](t) - [\xi_{Pk,i}]\|, \quad i \in 1, \dots, N_A. \quad (\text{III.40})$$

which simplifies the interval computation but also keeps the interval structure. Once the distance is obtained, it is possible to compute the weight factors by Eq. III.35 and Eq. III.36.

One more thing to notice here is the division operation. When an interval is divided by another interval, the obtained interval is practically expanded which leads to large uncertainty interval. Without losing the system performance, the terms of denominator are replaced by the respective centers of computed distances. Consequently, Eq. III.35 and Eq. III.36 are replaced by

$$[\omega_{j,k}](t) = \frac{\|[\xi_{Xj}](t), [\xi_{Pk}]\|^{-\alpha}}{\sum_{u \in I_j(t)} \text{center} (\|[\xi_{Xj}](t), [\xi_{Pu}]\|^{-\alpha})}. \quad (\text{III.41})$$

and,

$$[\omega_{j,k}](t) = \frac{\exp(-\alpha \|[\xi_{Xj}](t), [\xi_{Pk}]\|)}{\sum_{u \in I_j(t)} \text{center} (\exp(-\alpha \|[\xi_{Xj}](t), [\xi_{Pu}]\|))}. \quad (\text{III.42})$$

alternatively in terms of 1-norm,

$$[\omega_{j,k}](t) = \frac{\|[\xi_{Xj}](t), [\xi_{Pk}]\|_1^{-\alpha}}{\sum_{u \in I_j(t)} \text{center} (\|[\xi_{Xj}](t), [\xi_{Pu}]\|_1^{-\alpha})}. \quad (\text{III.43})$$

and,

$$[\omega_{j,k}](t) = \frac{\exp(-\alpha \|[\xi_{Xj}](t), [\xi_{Pk}]\|_1)}{\sum_{u \in I_j(t)} \text{center} (\exp(-\alpha \|[\xi_{Xj}](t), [\xi_{Pu}]\|_1))}. \quad (\text{III.44})$$

Besides the definition $[\omega_{j,k}](t)$ in terms of 1-norm and 2-norm, one could alternatively add the uncertainty $\pm \delta_{x_\xi}$ to the scalar position estimates as mentioned in section III.3.2. To this end, we need to generate a set of randomly deployed test positions and take their maximal estimation error. In what follows, the performances are compared using δ_ξ , in both 2-norm and 1-norm, and estimated uncertainty δ_{x_ξ} with 50 test positions. The accelerometer method is fixed to the 3rd-order model with applying δ_{x_γ} for accelerations uncertainty in combination. The estimation errors are presented sequentially in Table III.9 - Table III.14. The number of anchors is set to 49, the number of reference positions to 100, $\sigma_\xi = 0.5 \text{dBm}$, $\sigma_\gamma = 0.01 \text{m/s}^2$, $K = 3$ and the weights with distance inversion are considered with $\alpha = 1$.

III. CENTRALIZED LOCALIZATION

Table III.9: Fingerprints incertitude - Estimation results using trajectory 1

Trajectory 1	Acc	Fingerprints			Combined using δ_{x_γ}		
		*	δ_ξ -2norm	δ_ξ -1norm	δ_ξ -2norm	δ_ξ -1norm	δ_{x_ξ}
Error (m)	2.7125	2.1713	2.1480	1.5635	1.9432	1.2857	0.9704

Table III.10: Fingerprints incertitude - Estimation results using trajectory 2

Trajectory 2	Acc	Fingerprints			Combined using δ_{x_γ}		
		*	δ_ξ -2norm	δ_ξ -1norm	δ_ξ -2norm	δ_ξ -1norm	δ_{x_ξ}
Error (m)	2.8626	1.8362	1.8594	1.5008	1.6201	1.2248	0.9273

Table III.11: Fingerprints incertitude - Estimation results using trajectory 3

Trajectory 3	Acc	Fingerprints			Combined using δ_{x_γ}		
		*	δ_ξ -2norm	δ_ξ -1norm	δ_ξ -2norm	δ_ξ -1norm	δ_{x_ξ}
Error (m)	3.1822	2.0353	2.0053	1.5954	1.6965	1.3206	1.0559

Table III.12: Fingerprints incertitude - Estimation results using trajectory 4

Trajectory 4	Acc	Fingerprints			Combined using δ_{x_γ}		
		*	δ_ξ -2norm	δ_ξ -1norm	δ_ξ -2norm	δ_ξ -1norm	δ_{x_ξ}
Error (m)	2.8526	1.7734	1.7917	1.6674	1.7530	1.5028	1.0154

Table III.13: Fingerprints incertitude - Estimation results using trajectory 5

Trajectory 5	Acc	Fingerprints			Combined using δ_{x_γ}		
		*	δ_ξ -2norm	δ_ξ -1norm	δ_ξ -2norm	δ_ξ -1norm	δ_{x_ξ}
Error (m)	2.2049	2.0457	2.0058	1.6743	1.8981	1.3220	1.1493

Table III.14: Fingerprints incertitude - Estimation results using trajectory 6

Trajectory 6	Acc	Fingerprints			Combined using δ_{x_γ}		
		*	δ_ξ -2norm	δ_ξ -1norm	δ_ξ -2norm	δ_ξ -1norm	δ_{x_ξ}
Error (m)	3.0467	2.0857	2.0776	1.5331	1.7342	1.3115	1.2666

By observing these tables, the performance varies with the different introduced incertitude, as well as with the different tested trajectories. Generally speaking, Euclidean distance cannot always give the suitable distance estimation anymore over the computation of intervals. Comparatively, Manhattan distance recovers such distortion in some ways. However, using the incertitude over the position estimation achieves better results in all cases. In the following simulations, the combination of finally added incertitude with $(\delta_{x_\gamma}, \delta_{x_\xi})$ is chosen to make further illustrations over the impact of all concerned parameters.

III.4.4 Impact of parameters

The method performance varies with different parameters considered in the localization algorithm, such as the number of neighbors K , the number of anchors N_A , the number of references positions N_P as well as coefficient α observed in both types of weights factors. Some of them are deemed to be determined at advance, while others need to meet the requirement of the network configuration. This part discusses the variation of performances with respect to these parameters using the fourth trajectory. The relevant acceleration curves are depicted in Figure III.7.

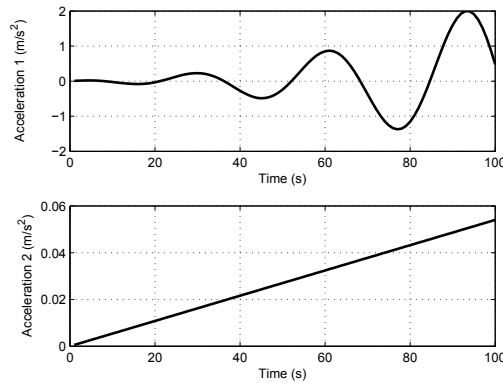


Figure III.7: Acceleration signals of trajectory 4

III.4.4.1 Impact of K

It is recommended to discuss the choice of parameter K to be considered in the RSSI-based computation. To this end, we vary the number of neighbors K from 1 to 10, we set meanwhile N_A to 49 and N_P to 100. The noise standard deviations corresponding to acceleration and RSSI are set separately to $\sigma_\gamma = 0.01m.s^{-2}$ and $\sigma_\xi = 0.1dBm$. While varying values of K , all corresponding results are performed 50 rounds relying on the average error. Figure III.8 illustrates the estimation errors as functions of K , obtained with the proposed combined method, the single acceleration-based one and the single fingerprinting-based one. When K is higher than 2, errors are slightly varying. The

III. CENTRALIZED LOCALIZATION

proposed method leads to more accurate estimates than the other methods in all cases. In the following simulations, $K = 3$ is chosen since with $K = 3$ the best results are obtained.

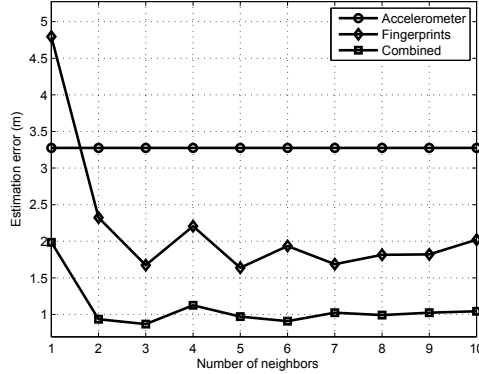


Figure III.8: Estimation errors as functions of K

III.4.4.2 Impact of N_A

The number of anchors N_A is varied in this section from 25 to 400, with N_P set to 100, K to 3, σ_γ to $0.01m.s^{-2}$ and σ_ξ to $0.1dBm$. Figure III.9 shows the curves of estimation errors with respect to the increasing number of anchors, obtained with the combined method, the single acceleration-based method and the single fingerprinting-based one. As expected, the estimation error generally decreases with the increase of the number of considered anchors, with the combined method outperforming the single methods in all cases. The error curves however show irregularities around $N_A = 81$. In fact, the estimation accuracy is highly influenced by the proximity of some reference points to the anchors, due to the logarithmic expression of the RSSI/distance. Indeed the logarithmic function highly decreases with slightly varying argument values when they are small. That is, when a distance separating a reference point to an anchor is too small, the corresponding RSSI is very high. A position not too far from this reference point could have highly varying RSSI from the reference point one, misleading this way the nearest neighbors algorithm. This explains why for some N_A values, the error increases instead of decreasing with increasing N_A .

III.4.4.3 Impact of N_P

The number of reference positions N_P is varied afterwards from 25 to 400, while setting K to 3, N_A to 49, σ_γ to $0.01m.s^{-2}$ and σ_ξ to $0.1dBm$. Figure III.10 shows the curves of estimation errors with respect to N_P . As expected, the estimation error decreases with the increase of the number of considered reference positions. The combined method achieves better performance, than the single ones. In all experiments, the reference

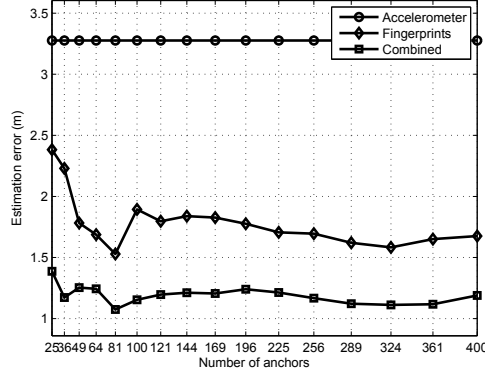


Figure III.9: Estimation errors as functions of N_A

positions are regularly spaced over the surveillance area. We now consider randomly distributed reference points according to a uniform distribution over the whole surveillance area. Figure III.11 shows the curves of estimation errors as functions of N_P , obtained with the only-fingerprinting and the combined methods with regularly spaced (RegS) reference points and randomly distributed (Rand) reference points. The simulation is run 50 times and the errors are average over all the results. The plot shows that with regularly spaced reference points, the localization is more accurate with all values of N_P . Indeed, with the regular distribution, the network is regularly covered, without having sub-covered regions.

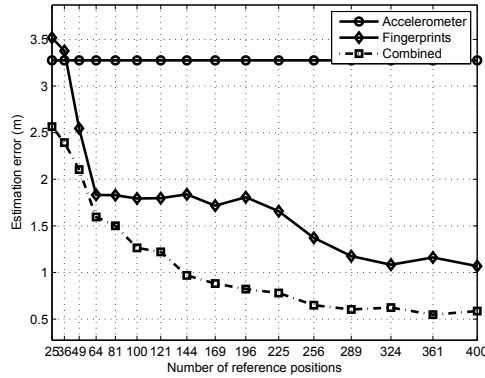


Figure III.10: Estimation errors as functions of N_P

III.4.4.4 Impact of σ_γ and σ_ξ

The additional noises are also an important aspect to assess algorithm adaptability. In the following, the impacts of added noises are discussed. Firstly, we vary the standard deviation of the acceleration noise σ_γ from $0.0035m.s^{-2}$ till $0.008m.s^{-2}$ with fixed $K =$

III. CENTRALIZED LOCALIZATION

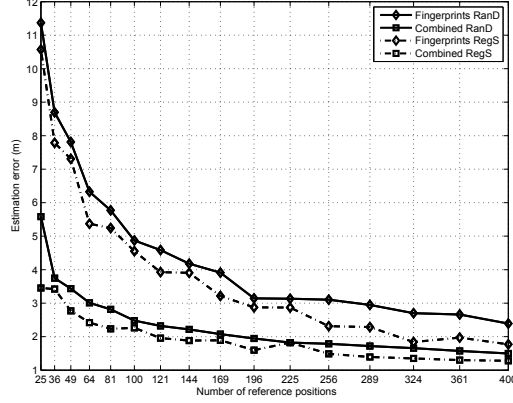


Figure III.11: Estimation errors with different distributions of reference points

3, $N_A = 49$, $N_P = 100$ and $\sigma_\xi = 1dBm$. The plot of Figure III.12 shows the curves of the estimation errors obtained with the combined method, the acceleration-based one and the fingerprinting-based one with respect to σ_γ . As expected, the estimation error increases with the increase of the noise in the methods using accelerations. However, the combined method remains much more accurate than the other ones.

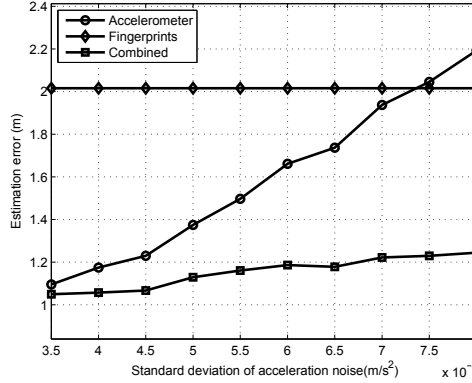


Figure III.12: Estimation errors as functions of noise standard deviations σ_γ

Afterwards, we vary the standard deviation of RSSI noise σ_ξ from $0.01dBm$ till $4dBm$ with $K = 3$, $N_A = 49$, $N_P = 100$ and $\sigma_\gamma = 0.01m.s^{-2}$. The plot of Figure III.13 shows the estimation errors with respect to σ_ξ . Also, the plot shows that the estimation errors increase with the increase of the RSSI noise. Moreover, the combined method outperforms the single ones.

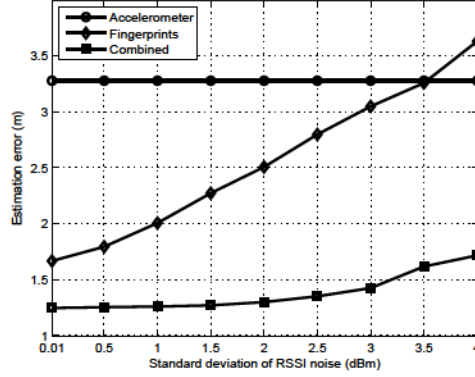


Figure III.13: Estimation errors as functions of noise standard deviations σ_ξ

III.4.4.5 Impact of weights parameters

The impact of the weights expressions is also worth to analyse. Indeed, the weights are used in the computations of the estimates while using the nearest neighbors algorithm with the RSSI fingerprints. To this end, we set parameters K to 3, N_A to 49, N_P to 100, and σ_γ and σ_ξ to $0.01m/s^2$ and $0.1dBm$ respectively. Firstly, we consider the inverting distance scheme given in Eq. III.4 and we vary the parameter α from 1 to 5. Then we consider the exponential weight scheme given in Eq. III.5 and we vary the parameter α from 1 to 5 also. Figure III.14 shows the estimation errors curves as functions of the parameters values, with both schemes as functions of α . The plot shows that with the inverting distance scheme, the proposed method outperforms its version with the exponential scheme. Also, the configuration with Eq. III.4 using $\alpha = 2$ receives best results. In the simulations presented later, this configuration of weight factor is adopted.

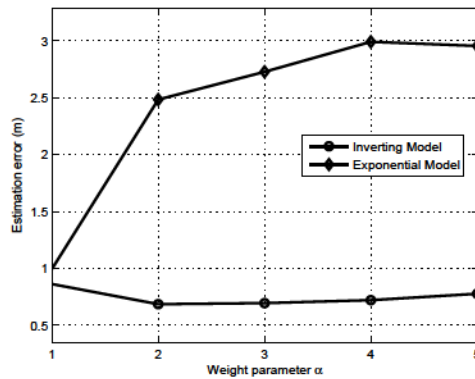


Figure III.14: Estimation errors as functions of weight parameters

III. CENTRALIZED LOCALIZATION

III.4.5 Comparison to a connectivity-based method

In this section, the proposed method is put forward to be compared with a connectivity-based method, using Monte-Carlo [Baggio et Langendoen 2006] where the strength of a signal is assumed to decrease monotonically with the increase of its traveled distance, as for the proposed method. According to [Baggio et Langendoen 2006], each mobile node compares its online RSSI measurements at each time-step to a certain threshold ξ_r , associated to a distance r . If the RSSI of a signal emitted by an anchor i is higher than ξ_r , the distance separating the anchor from the node at this time-step is less than r . In this case, the anchor i is assumed to be detected by the node, leading to a disk constraint centered on the anchor and having r as radius, including the node position. The solution of the problem is then set by intersecting all disks of detected anchors. In order to define it using Monte-Carlo, N particles are generated in the way to satisfy all detected anchors constraints. These particles are included in the intersection area of all disks. Finally, the estimated position is given by the barycenter of all generated particles. In order to compare this approach to our proposed method, set the number of anchors N_A to 49, the number of reference points N_P to 196, the number of neighbors K to 3, the standard deviations of noises σ_γ and σ_ξ to $0.001m.s^{-2}$ and $0.1dBm$ respectively. Accordingly, set the number of particles N to 100 and the range r to $15m$, leading to a power threshold equal to $53dBm$ according to the noiseless pathloss model with $\xi_0 = 100dBm$ and $n_P = 4$. With this value of r , an average of 3.33 anchors is detected by the node at each time-step. The connectivity-based method yields an estimation error of $3.6986m$, compared to $0.8216m$ with the combined method as shown in Table III.15. The average computation times are equal to $2.167ms$ and $0.252ms$ respectively. It is obvious that the proposed method is more accurate than the connectivity-based one using Monte-Carlo. Comparisons to other methods are given in the following chapter.

Table III.15: Comparison results

	Monte-Carlo	Combined method
Error (m)	3.6986	0.8216
Time (ms)	2.167	0.252

III.5 Conclusion

This chapter proposed a centralized localization technique for wireless sensor networks. The proposed method is an anchor-based method, using both fingerprinting and accelerometer information. Based on interval analysis, the proposed method computes two interval-estimates, then combines them, leading to position boxes including the actual locations of the nodes. It discussed several possible solutions in terms of different uncertainty modeling. Simulation results show that the proposed method outperforms

other methods using only accelerometer information or fingerprinting measures. It also illustrates the efficiency of the proposed approach, compared to a connectivity-based one using Monte-Carlo.

The centralized algorithm is easy to implement since all computations are done at the base station. However, it is more energy consuming and less robust than clusterized computations, especially in ad hoc networks. For this reason, a clusterized localization technique performing the fingerprints collection is further explored in the following chapter. Moreover, the measured accelerations values are no more reliable if some rotations occur to the mobile node having the accelerometer device. It is recommended to introduce some inertial measurements to make sure of the reliability of obtained movement data. The next chapter will focus on the relevant solutions to these problems.

REFERENCES

References

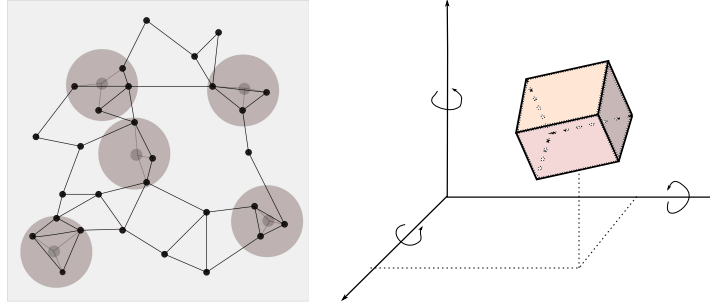
- [Akyildiz *et al.*, 2002] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam et E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38 (4): 393–422, 2002. 58
- [Arthi et Murugan, 2010] R. Arthi et K. Murugan. Localization in wireless sensor networks by hidden markov model. In *Advanced Computing (ICoAC), 2010 Second International Conference on*, pages 14–18. IEEE, 2010. 59
- [Baggio et Langendoen, 2006] A. Baggio et K. Langendoen. Monte-carlo localization for mobile wireless sensor networks. In *Mobile Ad-hoc and Sensor Networks*, pages 317–328. Springer, 2006. 58, 59, 84
- [Bahl et Padmanabhan, 2000] P. Bahl et V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. IEEE, 2000. 58
- [Camp *et al.*, 2002] T. Camp, J. Boleng et V. Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2 (5): 483–502, 2002. 59
- [Chandra-Sekaran *et al.*, 2009] A.-K. Chandra-Sekaran, P. Dheenathayalan, P. Weisser, C. Kunze et W. Stork. Empirical analysis and ranging using environment and mobility adaptive rssi filter for patient localization during disaster management. In *Networking and Services, 2009. ICNS'09. Fifth International Conference on*, pages 276–281. IEEE, 2009. 59
- [Choi *et al.*, 2011] B.-S. Choi, J.-W. Lee, J.-J. Lee et K.-T. Park. A hierarchical algorithm for indoor mobile robot localization using rfid sensor fusion. *Industrial Electronics, IEEE Transactions on*, 58 (6): 2226–2235, 2011. 59
- [Conner *et al.*, 2004] W. S. Conner, J. Heidemann, L. Krishnamurthy, X. Wang et M. Yarvis. Workplace applications of sensor networks. *USC/ISI Technical Report ISI-TR-2004-591*, 2004. 58
- [Czubak et Wojtanowski, 2009] A. Czubak et J. Wojtanowski. On applications of wireless sensor networks. In *Internet—Technical Development and Applications*, pages 91–99. Springer, 2009. 58
- [Fraile et Maybank, 1998] R. Fraile et S. J. Maybank. Vehicle trajectory approximation and classification. In *BMVC*, volume 98, pages 832–840, 1998. 59
- [Haverinen et Kemppainen, 2009] J. Haverinen et A. Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57 (10): 1028–1035, 2009. 59
- [Hofmann-Wellenhof *et al.*, 1993] B. Hofmann-Wellenhof, H. Lichtenegger et J. Collins. *Global Positioning System. Theory and Practice*. Springer, 1993. 58
- [Hsu et Yu, 2009] C.-H. Hsu et C.-H. Yu. An accelerometer based approach for indoor localization. In *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on*, pages 223–227. IEEE, 2009. 59, 62

- [Hu et Evans, 2004] L. Hu et D. Evans. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 45–57. ACM, 2004. 58, 59
- [Jaulin et al., 2001] L. Jaulin, M. Kieffer, O. Didrit et E. Walter. *Applied interval analysis*. Springer, 2001. 59, 64
- [Kaemarungsi et Krishnamurthy, 2004] K. Kaemarungsi et P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1012–1022. IEEE, 2004. 58
- [Mao et al., 2007] G. Mao, B. Fidan et B. Anderson. Wireless sensor network localization techniques. *Computer networks*, 51 (10): 2529–2553, 2007. 58
- [Medeisis et Kajackas, 2000] A. Medeisis et A. Kajackas. On the use of the universal okumura-hata propagation prediction model in rural areas. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 3, pages 1815–1818. IEEE, 2000. 71
- [Mourad et al., 2008] F. Mourad, H. Snoussi, F. Abdallah et C. Richard. Guaranteed boxed localization in manets by interval analysis and constraints propagation techniques. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5. IEEE, 2008. 58, 59
- [Mourad et al., 2009] F. Mourad, H. Snoussi, F. Abdallah et C. Richard. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *Signal Processing, IEEE Transactions on*, 57 (8): 3226–3239, 2009. 58, 59, 64
- [Navarro et al., 2010] E. Navarro, B. Peucker et M. Quan. Wi-fi localization using rssi fingerprinting. In *California Polytechnic State University, USA*, pages 1–6, 2010. 61
- [Ni et al., 2004] L. M. Ni, Y. Liu, Y. C. Lau et A. P. Patil. Landmarc: indoor location sensing using active rfid. *Wireless networks*, 10 (6): 701–710, 2004. 59
- [Pal, 2010] A. Pal. Localization algorithms in wireless sensor networks: Current approaches and future challenges. *Network Protocols and Algorithms*, 2 (1): 45–73, 2010. 58
- [Robles et al., 2010] J. J. Robles, M. Deicke et R. Lehnert. 3d fingerprint-based localization for wireless sensor networks. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, pages 77–85. IEEE, 2010. 58, 61
- [Schweitzer et Syafrudin, 2010] H. Schweitzer et M. Syafrudin. Losnus: An ultrasonic system enabling high accuracy and secure tdoa locating of numerous devices. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–8. IEEE, 2010. 58
- [Shorey et al., 2006] R. Shorey, A. Ananda, M. C. Chan et W. T. Ooi. *Mobile, wireless, and sensor networks: technology, applications, and future directions*. John Wiley & Sons, 2006. 58
- [Sifuentes et al., 2011] E. Sifuentes, O. Casas et R. Pallas-Areny. Wireless magnetic sensor node for vehicle detection with optical wake-up. *Sensors Journal, IEEE*, 11 (8): 1669–1676, 2011. 58

REFERENCES

- [Stojmenovic, 2005] I. Stojmenovic. *Handbook of sensor networks: algorithms and architectures*, volume 49. John Wiley & Sons, 2005. 58
- [Tapus et Siegwart, 2006] A. Tapus et R. Siegwart. A cognitive modeling of space using fingerprints of places for mobile robot navigation. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1188–1193. IEEE, 2006. 58
- [Teng et al., 2010] J. Teng, H. Snoussi et C. Richard. Decentralized variational filtering for target tracking in binary sensor networks. *Mobile Computing, IEEE Transactions on*, 9(10): 1465–1477, 2010. 58
- [Zhao et Guibas, 2004] F. Zhao et L. J. Guibas. *Wireless sensor networks: an information processing approach*. Morgan Kaufmann, 2004. 58

IV Clusterized localization



Contents

IV.1	Introduction
IV.2	Configuration of the network
IV.3	Clusterized localization method
IV.4	Resolution using the interval analysis
IV.5	Resolution using the Kalman filter
IV.6	Simulation results
IV.7	Conclusion
	References

Centralized localization is tackled in a rather simple way, permitting complex computational task at the central base station with sacrifice over the communication cost of data immigrations in the network. Tradeoff between cost of communication and computation turns to be of weakness once the network becomes larger or denser. Besides, many applications are oriented to the large-scale, or for harsh environmental contexts where the efficiency of centralized algorithms drops enormously. Clusterized localization circumvents this limitation by introducing massive parallel computations at nodes themselves or several computation points distributed over the network. Considering the latter solution, the surveillance area is required to be divided into several clusters, each equipped with a head of cluster. The proposed clusterized localization algorithm employs both RSSI and inertial information, to compute first estimates using RSSI-fingerprints, then correct them using inertial information leading to more accurate estimates. Computations are performed using both solutions of the interval analysis and the Kalman filter techniques.

IV.1 Introduction

The new generations of wireless sensor networks (WSNs) tend to be more flexible, with higher capacity and functionality. Advances in wireless networking, microfabrication and integration have enabled the massive-scale networks suitable for a range of commercial and military applications [Zhao et Guibas 2004; Roseveare et Natarajan 2012]. Localizing sensors in a centralized manner in such networks encounters thus a dilemma on communication cost inherited between sensors. Indeed, the energy consumption increases rapidly even enormously, especially for the sensors nearby the central station, with the normal growth of network scale in a centralized manner as mentioned in chapter I. Consequently, the clusterized localization algorithm is notably recommended here, since generally sensors are not only able to sense and communicate with each other, but they are also qualified to execute some complex computations and operations by themselves.

Technically speaking, the centralized architecture considers only a single fusion center that plays the unique role of processing all the information in the network [Tharmarasa et al. 2011]. Such architectures are high energy consuming since all information should be routed to the fusion center then relayed accordingly back into the network after the required analysis and processing. Nevertheless the fusion center undertakes all the computation task since it is specially designed. It is worth noting that the competition between computation capacity and communication cost turns to be unbalanced once the networks scale increases largely. By introducing individual agents, which could be smart sensors themselves, in the network, information could be analyzed and processed at hand or nearby, obtaining yet satisfying processing accuracy but saving communication consumption. Moreover, as the clusterized architecture considers several collection and computation points, it is usually more efficient and more robust to failure than the centralized approach [Tharmarasa et al. 2011; Ferrari et Pagliari 2006]. Consequently, clusterized network architectures not only relieve the computation pressure and increase the process efficiency but also they are especially susceptible of the unexpected malicious nodes.

Working through a clusterized localization manner, the entire surveillance area as talked in Chapter III is considered to be divided into several small clusters according to the location of individual agents. Under this context, compared with one single estimate of centralized locating, clusterized algorithm offers the network individual localization estimates in parallel corresponding to clusters. The individuals then manage to combine themselves into one solution in considering their relevant importance.

Under this context, this chapter presents a novel localization technique based on fingerprinting and inertial information in clusterized sensor networks. The proposed method is all the same an anchor-based range-free method performing fingerprinting collection but in a clusterized scheme. The method consists of the configuration phase, followed by the localization of the nodes. In the configuration phase, the surveillance area is divided into several clusters each of which equipped with a head of cluster. Then RSSI fingerprints are collected to model each cluster of the network, leading to a clusterized

power map. Afterwards, in order to localize a given node, the power map is used with the nearest neighbor algorithm to compute local position estimates, whose combination leads to a global estimate at each time step [Chen et Chang 2008; Aldhubaib et Shuley 2010]. The collected accelerations [Schuler et al. 1967; Lee et Huang 2002] are then used to correct the global estimate using either the interval analysis [Moore 1966; Jaulin et al. 2001; Mourad et al. 2009] which allows a guaranteed estimation of the target node position in terms of interval, or the Kalman filter [Kalman 1960; Chan et al. 1979; Honghui et Moore 2002] which performs the optimal recursive data processing and gives an efficient estimation by incorporating all information issued from different sources.

IV.2 Configuration of the network

Consider the network is composed of N_A fixed anchors and N_X mobile nodes, denoted respectively by $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,D})$, $i = 1, \dots, N_A$, and $\mathbf{x}_j(t) = (x_{j,1}(t), \dots, x_{j,D}(t))$, $j = 1, \dots, N_X$, t being the current iteration and D the dimension of the surveillance area ($D = 2$ or 3). The aim of the method is to localize the nodes with respect to the anchors using their exchanged information.

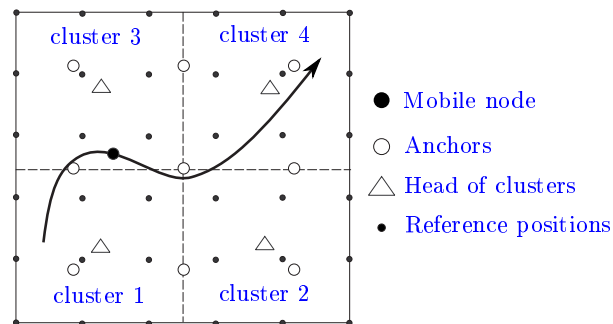


Figure IV.1: Composition of the clustered network

Being a clustered approach, the proposed method consists of dividing the surveillance area into several clusters. These clusters could be set according to the configuration of the obstacles in the surveillance area, or by dividing the whole area into equal clusters. Figure IV.1 shows an example of a network in a 2D environment, having nine anchors and one mobile node, and divided into four equal clusters. Each cluster is equipped with a head of cluster that could be a small computer or an anchor having computation capabilities. Let N_C be the number of clusters and let $\mathbf{h}_{(c)} = (h_{(c),1}, \dots, h_{(c),D})$, $c = 1, \dots, N_C$, be the head of clusters coordinates. The heads are usually positioned at the centers of the clusters and they are assumed to have sufficient communication ranges in the way to be able to communicate with each others. In order to localize the nodes, the network is first modeled by a collection of radio-fingerprints. To this end, $N_{P(c)}$ uniformly distributed reference positions are generated in each cluster c , $c = 1, \dots, N_C$.

IV. CLUSTERIZED LOCALIZATION

These positions are denoted by $\mathbf{p}_{(c),n} = (p_{(c),n,1}, \dots, p_{(c),n,D})$, $n = 1, \dots, N_{P(c)}$. Then anchors broadcast signals in the network with the same initial power. It is noted that their signal powers are inversely proportional to transmission distances. By placing a sensor consecutively at reference positions, one is able to receive the anchor signals and measure their Received Signal Strength Indicators (RSSIs) at these positions. Let $\boldsymbol{\xi}_{P(c),n}$ be the reference RSSI vector given by

$$\boldsymbol{\xi}_{P(c),n} = (\xi_{P(c),n,1}, \dots, \xi_{P(c),n,N_A}), \quad (\text{IV.1})$$

where $\xi_{P(c),n,i}$ is the strength of the signal emitted by the anchor i and received at the reference position $\mathbf{p}_{(c),n}$ of the cluster c . The configuration phase leads then to a fingerprinting database, namely the power map [Robles *et al.* 2010]. This map consists of the union of N_C submaps, having each $N_{P(c)}$ couples, composed of the reference positions with their associated RSSI vectors $(\mathbf{p}_{(c),n}, \boldsymbol{\xi}_{P(c),n})$, $n = 1, \dots, N_{P(c)}$ and $c = 1, \dots, N_C$. Each fingerprinting submap is stored in the corresponding head of cluster, to be used hereafter with the online data in order to perform the localization computation.

IV.3 Clusterized localization method

Once the clusterized configuration is performed, the algorithm proceeds to both estimations by fingerprints and inertial readings. In what follows, first estimates are presented by using fingerprints in a clusterized manner. After, inertial information will be taken into consideration to correct the fingerprints estimates. Solutions are given using interval analysis and Kalman filter.

IV.3.1 Estimation using fingerprints

Each mobile node collects while moving the RSSIs of the signals it receives from the anchors, leading to online RSSI vectors as follows,

$$\boldsymbol{\xi}_{X,j}(t) = (\xi_{X,j,1}(t), \dots, \xi_{X,j,N_A}(t)), \quad j \in \{1, \dots, N_X\}, \quad (\text{IV.2})$$

where $\xi_{X,j,i}(t)$ is the strength at reception of the signal emitted by the anchor i and received by the node j at iteration t . Then, nodes send their RSSI vectors to the heads of clusters within their communication ranges. Let $I_{C_j}(t)$ be the set of indices of the heads receiving the node j RSSI online vector at iteration t . In order to localize the node j , each head of $I_{C_j}(t)$ computes the Euclidean distances $\|\boldsymbol{\xi}_{X,j}(t), \boldsymbol{\xi}_{P(c),n}\|$ between the online RSSI vectors $\boldsymbol{\xi}_{X,j}(t)$ and all reference RSSI vectors $\boldsymbol{\xi}_{P(c),n}$ of its cluster. Then, according to the nearest neighbor algorithm [Chen *et al.* 2008; Aldhubaib *et al.* Shuley 2010], the K reference positions giving the K smallest distances are selected. Let $I_{P(c),j}(t)$, $c \in I_{C_j}(t)$, be the set of indices of these reference positions. Local position estimates are then computed as follows,

$$\hat{\mathbf{x}}_{(c),j}(t) = \sum_{n \in I_{P(c),j}(t)} \omega_{(c),j,n}(t) \mathbf{p}_{(c),n}, \quad c \in I_{C_j}(t), \quad (\text{IV.3})$$

where $\omega_{(c),j,n}(t)$ is the weight associated to the reference position $\mathbf{p}_{(c),n}$ using the node j information at iteration t . The weights, defined similarly as in Chapter III, are chosen in the way to be inversely proportional to the RSSI distances. Indeed, having a small distance between the RSSI vectors means that the node is close to the corresponding reference position, and thus its corresponding weight should be high. Also, the weights should be normalized, $\sum_{n \in I_{P(c),j}(t)} \omega_{(c),j,n}(t) = 1$. One could choose for instance the inverting distance scheme

$$\omega_{(c),j,n}(t) = \frac{\|\boldsymbol{\xi}_{Xj}(t), \boldsymbol{\xi}_{P(c),n}\|^{-\alpha}}{\sum_{u \in I_{P(c),j}(t)} \|\boldsymbol{\xi}_{Xj}(t), \boldsymbol{\xi}_{P(c),u}\|^{-\alpha}}, \quad (\text{IV.4})$$

where α denotes the parameter of the algorithm, or the exponential scheme

$$\omega_{(c),j,n}(t) = \frac{\exp\left(-\alpha \|\boldsymbol{\xi}_{Xj}(t), \boldsymbol{\xi}_{P(c),n}\|\right)}{\sum_{u \in I_{P(c),j}(t)} \exp\left(-\alpha \|\boldsymbol{\xi}_{Xj}(t), \boldsymbol{\xi}_{P(c),u}\|\right)}. \quad (\text{IV.5})$$

Now that local estimates are computed at the heads of clusters neighboring the node j , a first estimate is computed by combining these local estimates. To do this, heads use the strength of the messages including the online RSSI vectors they receive from the node. Let $\xi_{H(c),j}(t)$, $c \in I_{Cj}(t)$ and $j = 1, \dots, N_X$, be these strengths. Heads exchange then their measured RSSIs. The closest one to the node having the greatest RSSI collects all the local estimates to combine them. To this end, second weights $\lambda_{(c),j}(t)$ are computed using the exponential scheme as follows,

$$\lambda_{(c),j}(t) = \frac{\exp\left(\xi_{H(c),j}(t)\right)}{\sum_{u \in I_{Cj}(t)} \exp\left(\xi_{H(u),j}(t)\right)}. \quad (\text{IV.6})$$

A 1st global position estimate is computed at iteration t for each node j using local estimates as follows,

$$\hat{\mathbf{x}}_j^-(t) = \sum_{c \in I_{Cj}(t)} \lambda_{(c),j}(t) \hat{\mathbf{x}}_{(c),j}(k), \quad j \in \{1, \dots, N_X\}. \quad (\text{IV.7})$$

The λ -weight, used to combine the local estimates, should be high if the node is close to its corresponding head of cluster and small otherwise. This way the local estimates of the nearest heads are considered more than those of far ones. Since the RSSI decreases with the increase of the traveled distance, the higher $\xi_{H(c),j}(t)$ is, the closer the node j is to the head of cluster c and the higher its corresponding weight $\lambda_{(c),j}(t)$ should be. One could use for instance the normalized RSSIs as weights. However, by using the proposed exponential formulation of Eq. IV.6, one gives much more importance to the local estimate of the closest cluster to the node, which has the reference RSSI vectors generated in the same geographical conditions as for the node at time t .

IV. CLUSTERIZED LOCALIZATION

IV.3.2 Estimation using the inertial information

Previous Chapter III has discussed three mobility models applied in the localization algorithm. In the following, basic solution by these models is revised. Then an extension to the case of rotation occurrence is equally discussed.

IV.3.2.1 Basic solution

Each node is equipped with an accelerometer that yields its instantaneous accelerations over the D coordinates [Lee et Huang 2002; Hsu et Yu 2009]. Let $\boldsymbol{\gamma}_j(t) = (\gamma_{j,1}(t), \dots, \gamma_{j,D}(t))$ be the acceleration vector of the node j at iteration t . Using $\boldsymbol{\gamma}_j(t)$, one is able to estimate the position of the node j by using the second-order mobility model as follows,

$$\hat{\boldsymbol{x}}_j(t) = \hat{\boldsymbol{x}}_j(t-1) + \boldsymbol{\nu}_j(t-1)\Delta t + 0.5\boldsymbol{\gamma}_j(t)\Delta t^2, \quad (\text{IV.8})$$

where $\boldsymbol{\nu}_j(t)$ is the estimated velocity vector of the node j at iteration t , computed iteratively by

$$\boldsymbol{\nu}_j(t) = \boldsymbol{\nu}_j(t-1) + \boldsymbol{\gamma}_j(t)\Delta t, \quad (\text{IV.9})$$

with Δt the time period between two consecutive iterations. The nodes are assumed to be fixed at $t = 0$ at known positions with zero velocities. Note that model as in Eq. IV.8 assumes that the acceleration vector is constant between the iterations $t-1$ and t . With slightly varying accelerations, this model performs well. However, with abrupt changes in accelerations or noisy interrupted accelerations, estimates might be sensitively deviated from the exact trajectory, due to cumulative errors over time. Another more approximated model could also be used as follows,

$$\hat{\boldsymbol{x}}_j(t) = \hat{\boldsymbol{x}}_j(t-1) + \boldsymbol{\nu}_j(t)\Delta t, \quad (\text{IV.10})$$

where velocities are computed using Eq. IV.9. In model of Eq. IV.10, called first-order model, the velocity vector is assumed to be constant between the iterations $k-1$ and k . This model is less accurate, but runs faster than the first one. Alors, the third-order mobility model is equally proposed as shown in Eq. IV.11,

$$\begin{aligned} \hat{\boldsymbol{x}}_j(t) &= \hat{\boldsymbol{x}}_j(t-1) + \hat{\boldsymbol{\nu}}_j(t-1)\Delta t + \frac{1}{2}\boldsymbol{\gamma}_j(t-1)\Delta t^2 + \frac{1}{6}\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}\Delta t^3 \\ &= \hat{\boldsymbol{x}}_j(t-1) + \hat{\boldsymbol{\nu}}_j(t-1)\Delta t + \frac{1}{3}\boldsymbol{\gamma}_j(t-1)\Delta t^2 + \frac{1}{6}\boldsymbol{\gamma}_j(t)\Delta t^2, \end{aligned} \quad (\text{IV.11})$$

where $\hat{\boldsymbol{\nu}}_j(t-1)$ is the velocity vector of the node j at time-step $t-1$ computed at iteration t as in Eq. IV.12,

$$\begin{aligned} \hat{\boldsymbol{\nu}}_j(t) &= \hat{\boldsymbol{\nu}}_j(t-1) + \boldsymbol{\gamma}_j(t-1)\Delta t + \frac{1}{2}\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}\Delta t^2 \\ &= \hat{\boldsymbol{\nu}}_j(t-1) + \frac{1}{2}\boldsymbol{\gamma}_j(t-1)\Delta t + \frac{1}{2}\boldsymbol{\gamma}_j(t)\Delta t. \end{aligned} \quad (\text{IV.12})$$

The accelerations are linear going from $\gamma_j(t-1)$ to $\gamma_j(t)$ between two consecutive time-steps $t-1$ and t with a slope of $\frac{\gamma_j(t)-\gamma_j(t-1)}{\Delta t}$. With a small value of Δt , this model works rather good, since the approximated acceleration curves become very close to the real ones.

The aim of the proposed method is to take advantage of the accelerometer-based estimates to improve the 1^{st} global estimates obtained by using the fingerprints power map. Details of the combination of all types of information are given in the following two sections.

IV.3.2.2 Extension to rotation

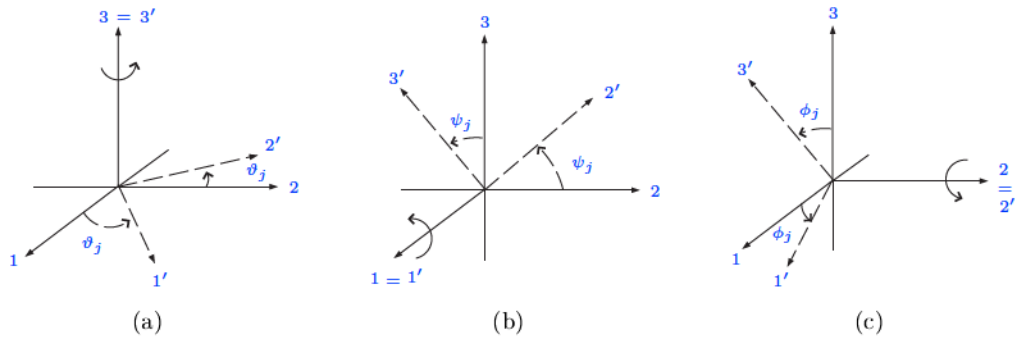


Figure IV.2: 3D rotations

By using the direct acceleration measurements made in the nodes, the method assumes that the node is rotationally constrained, that is the nodes coordinate system remains the same as the world coordinate system. However, while moving, the nodes could rotate and the D accelerations measured by each node are no more in world coordinates. To overcome this problem, the nodes should be equipped with gyroscopes, that yield their orientations with respect to the initial coordinate system. Assume that $D = 3$, that is the localization is performed in a 3D environment, and let $\vartheta_j(t)$, $\psi_j(t)$ and $\phi_j(t)$ be the angles of rotation of the node j at time t around the third coordinate axis of the world system, the first one and the second one respectively. The plots (a), (b) and (c) of Figure IV.2 illustrate these respective rotations, 1,2 and 3 being the world coordinate axes and 1',2' and 3' the node j ones. These rotations are assumed to be counter-clockwise when the axis around which they occur is pointed toward the observer, the coordinate system is right-handed and all angles are positive. For instance, $\vartheta_j(t)$ represents the rotation angle, around the third coordinate, of a vector aligned with the first coordinate and going toward the second one. In the following explanation, we will withdraw the notation (t) in the sake of clarity. Now let $\gamma'_{j,1}$, $\gamma'_{j,2}$ and $\gamma'_{j,3}$ be the measured accelerations of the node j at time t in its coordinate system. Then its

IV. CLUSTERIZED LOCALIZATION

accelerations $\gamma_{j,1}$, $\gamma_{j,2}$ and $\gamma_{j,3}$ in the world coordinates are given by

$$\begin{pmatrix} \gamma_{j,1} \\ \gamma_{j,2} \\ \gamma_{j,3} \end{pmatrix} = \mathfrak{R}(\vartheta_j, \psi_j, \phi_j) \begin{pmatrix} \gamma'_{j,1} \\ \gamma'_{j,2} \\ \gamma'_{j,3} \end{pmatrix}, \quad (\text{IV.13})$$

where the first column of the 3D rotation matrix is defined by

$$\begin{pmatrix} \cos \vartheta_j \cos \phi_j \\ \cos \vartheta_j \sin \psi_j \sin \phi_j + \sin \vartheta_j \cos \psi_j \\ -\cos \vartheta_j \cos \psi_j \sin \phi_j + \sin \vartheta_j \sin \psi_j \end{pmatrix}, \quad (\text{IV.14})$$

its second column is defined by

$$\begin{pmatrix} -\sin \vartheta_j \cos \phi_j \\ -\sin \vartheta_j \sin \psi_j \sin \phi_j + \cos \vartheta_j \cos \psi_j \\ \sin \vartheta_j \cos \psi_j \sin \phi_j + \cos \vartheta_j \sin \psi_j \end{pmatrix}, \quad (\text{IV.15})$$

and its third column is defined by

$$\begin{pmatrix} \sin \phi_j \\ -\sin \psi_j \cos \phi_j \\ \cos \psi_j \cos \phi_j \end{pmatrix}. \quad (\text{IV.16})$$

In 2D space, the expressions of $\gamma_{j,1}$ and $\gamma_{j,2}$ could be obtained by considering only the first and the second coordinates and by setting $\phi_j = \psi_j = 0$. Indeed, rotation is only possible in the plane with the rotation angle ϑ_j . This leads to the following,

$$\begin{pmatrix} \gamma_{j,1} \\ \gamma_{j,2} \end{pmatrix} = \begin{pmatrix} \cos \vartheta_j & -\sin \vartheta_j \\ \sin \vartheta_j & \cos \vartheta_j \end{pmatrix} \begin{pmatrix} \gamma'_{j,1} \\ \gamma'_{j,2} \end{pmatrix}. \quad (\text{IV.17})$$

At each iteration, each node measures its accelerations in its coordinate system and its orientations using its gyroscope and then it computes its accelerations in the world coordinate system. These computed accelerations would be used in the localization algorithm by applying Eq. IV.8 - IV.12. In the following two sections, the nodes are assumed to be rotationally constraints for simplicity. However, the computations could be easily modified for the other case, especially in the interval-based algorithm. Performance on the case of nodes rotation will be illustrated in simulations section.

IV.4 Resolution using the interval analysis

Resolving the localization problem using the interval analysis consists of computing the 1st global estimates using intervals and then correcting these estimates using the accelerations. Indeed, having measurements incertitude over the RSSIs and the accelerations, one could rewrite the RSSIs and the accelerations in the interval framework, letting possible the computations using intervals and thus leading to estimated boxes, instead of scalar ones. Let $[\xi_{P^{(c)},n}]$, $c = 1, \dots, N_C$ and $n = 1, \dots, N_{P^{(c)}}$, be the reference power boxes including all the RSSIs of the signals emitted by the anchors and received at the reference positions $\mathbf{p}_{(c),n}$. Indeed, due to signals reflection, multipaths

or additive noise, signals powers might vary even for the same traveled distances. The power map in the interval framework is then composed of N_C submaps having couples composed of reference positions and their corresponding power boxes. Assume that $[\boldsymbol{\xi}_{X_j}](t)$, $j = 1, \dots, N_X$, are the online power boxes including the RSSIs of signals received at the nodes j at iteration t . Then distances are computed between RSSI boxes leading to distances intervals $\|[\boldsymbol{\xi}_{X_j}](t), [\boldsymbol{\xi}_{P(c,n)}]\|$. The K indices of the reference positions yielding the smallest centers of distances intervals are then selected to define the $I_{P(c),j}(t)$ set. Local boxes estimates $[\hat{\boldsymbol{x}}_{(c),j}](t)$ are then given as follows,

$$[\hat{\boldsymbol{x}}_{(c),j}](t) = \sum_{n \in I_{P(c),j}(t)} [\omega_{(c),j,n}](t) \boldsymbol{p}_{(c),n}, \quad c \in I_{C_j}(t), \quad (\text{IV.18})$$

where the weights $[\omega_{(c),j,n}](t)$ are intervals obtained using either Eq. IV.4 or Eq. IV.5 in the interval framework with boxed RSSIs instead of vectors RSSIs with aid of δ_ξ introduced in Chapter III. The 1st global boxed estimate for a node j is then obtained by

$$[\hat{\boldsymbol{x}}_j^-](t) = \sum_{c \in I_{C_j}(t)} [\lambda_{(c),j}](t) [\hat{\boldsymbol{x}}_{(c),j}](t), \quad j \in \{1, \dots, N_X\}, \quad (\text{IV.19})$$

where $[\lambda_{(c),j}](t)$ are interval weights employing Eq. IV.6 in the interval framework with boxed RSSIs. Alternatively, as shown in Chapter III, one could compute the 1st estimated box by i) generating random test positions in the surveillance area, ii) computing their estimated scalar positions using fingerprints, iii) defining an incertitude δ_{x_ξ} as the maximal error obtained over the test positions estimation, afterwards iv) computing the scalar estimate $\hat{x}_j^-(t)$ using non-interval RSSIs, and then v) defining the 1st estimated box by $[\hat{\boldsymbol{x}}_j^-](t) = \prod_{d=1}^{d=D} [\hat{x}_{j,d}^-(t) - \delta_{x_\xi}, \hat{x}_{j,d}^-(t) + \delta_{x_\xi}]$, $\hat{x}_{j,d}^-(t)$ being the d -th coordinate of $\hat{\boldsymbol{x}}_j^-(t)$ and Π the cartesian product. Simulations of Chapter III showed that this other way of defining $[\hat{\boldsymbol{x}}_j^-](t)$ leads to more accurate estimates. For this reason, it would be applied in the interval algorithm.

On the other hand, a 2nd estimated position box could be obtained for each node using accelerometer information. Indeed, each node measures its acceleration vector $\boldsymbol{\gamma}_j(t) = (\gamma_{j,1}(t), \dots, \gamma_{j,D}(t))$ and defines its acceleration box as follows,

$$[\boldsymbol{\gamma}_j](t) = \prod_{d=1}^{d=D} [\gamma_{j,d}](t) = \prod_{d=1}^{d=D} [\gamma_{j,d}(t) - \delta_\gamma, \gamma_{j,d}(t) + \delta_\gamma], \quad (\text{IV.20})$$

where δ_γ is the acceleration incertitude set as shown in Remark 1 of Chapter III. The 2nd estimated box at iteration t is then obtained according to the second-order model of Eq. IV.8 as follows,

$$[\hat{\boldsymbol{x}}_j^+](t) = [\hat{\boldsymbol{x}}_j](t-1) + [\boldsymbol{\nu}_j](t-1)\Delta t + 0.5[\boldsymbol{\gamma}_j](t)\Delta t^2, \quad (\text{IV.21})$$

where $[\hat{\boldsymbol{x}}_j](t-1)$ is the final estimated box, obtained at iteration $t-1$ by combining all information, and $[\boldsymbol{\nu}_j](t)$ is the box including the velocity of node j at iteration t ,

$$[\boldsymbol{\nu}_j](t) = [\boldsymbol{\nu}_j](t-1) + [\boldsymbol{\gamma}_j](t)\Delta t. \quad (\text{IV.22})$$

IV. CLUSTERIZED LOCALIZATION

By using the first-order model of Eq. IV.10, one obtains

$$[\hat{\mathbf{x}}_j^+](t) = [\hat{\mathbf{x}}_j](t-1) + [\boldsymbol{\nu}_j](t)\Delta t, \quad (\text{IV.23})$$

where $[\boldsymbol{\nu}_j](t)$ remains the same as in Eq. IV.22. Similarly for the third-order mobility model, nodes mobility is reformulated into interval framework as follows,

$$[\hat{\mathbf{x}}_j^+]_{\gamma}(t) = [\hat{\mathbf{x}}_j](t-1) + [\hat{\boldsymbol{\nu}}_j](t-1)\Delta t + \frac{1}{3}[\boldsymbol{\gamma}_j](t-1)\Delta t^2 + \frac{1}{6}[\boldsymbol{\gamma}_j](t)\Delta t^2, \quad (\text{IV.24})$$

where $\hat{\boldsymbol{\nu}}_j(t)$ is equally boxed by

$$[\boldsymbol{\nu}_j](t) = [\hat{\boldsymbol{\nu}}_j](t-1) + \frac{1}{2}[\boldsymbol{\gamma}_j](t-1)\Delta t + \frac{1}{2}[\boldsymbol{\gamma}_j](t)\Delta t. \quad (\text{IV.25})$$

Similarly to the 1st estimated box, one is able to compute the 2nd estimated box in an alternative manner as shown in Chapter III. Indeed, $[\hat{\mathbf{x}}_j^+](t)$ could be set as $[\hat{\mathbf{x}}_j](t-1) + [\mathbf{B}_j](t)$ where $[\mathbf{B}_j](t) = \prod_{d=1}^{d=D} [B_{j,d}(t) - \delta_{x_{\gamma}}(t), B_{j,d}(t) + \delta_{x_{\gamma}}(t)]$. The quantity $B_{j,d}(t)$ is equal to $\nu_{j,d}(t)\Delta t$ for the first-order model, $\nu_{j,d}(t-1)\Delta t + \frac{1}{2}\gamma_{j,d}(t)\Delta t^2$ for the second-order model or $\nu_{j,d}(t-1)\Delta t + \frac{1}{3}\gamma_{j,d}(t-1)\Delta t^2 + \frac{1}{6}\gamma_{j,d}(t)\Delta t^2$ for the third-order model, whereas the incertitude $\delta_{x_{\gamma}}(t)$ is equal to $\sqrt{t}\delta_{\gamma}\Delta t^2$ for the first-order model, $\sqrt{t - \frac{3}{4}\delta_{\gamma}\Delta t^2}$ for the second-order model or $\sqrt{\frac{t}{2} - \frac{13}{36}\delta_{\gamma}\Delta t^2}$ for the third-order model. For more details, refer to Remark 3 of Chapter III. Simulations of Chapter III showed that by using $\delta_{x_{\gamma}}(t)$, the algorithm leads to more accurate results. For this reason, the combination of finally added incertitude with $(\delta_{x_{\gamma}}, \delta_{x_{\xi}})$ is chosen to make illustrations of the interval-based solution.

Having two estimated boxes for each node j at iteration t , one could compute the final estimate by intersecting these boxes at each iteration as follows,

$$[\hat{\mathbf{x}}_j](t) = [\hat{\mathbf{x}}_j^-](t) \cap [\hat{\mathbf{x}}_j^+](t). \quad (\text{IV.26})$$

The exact position estimate is given by the center of the estimated box. If the intersection of these boxes is empty, one could choose to keep only the 1st estimate. Nevertheless, with this approach, one obtains boxes that usually include the exact positions of the nodes. Figure IV.3 shows an illustration of the proposed interval-based method. It shows in black points the reference positions, in black circles the anchors positions, in black triangles the heads of clusters, in light gray line the 1st fingerprinting estimated box, in dark gray dashed line the acceleration estimated box, in black line the final estimated box and in stars their corresponding estimated single positions. By intersecting the estimated boxes, one obtain smaller boxes with less incertitude leading to more accurate position estimates.

IV.5 Resolution using the Kalman filter

The process of combination could also be performed using the Kalman filter, being a linear estimation based on the criterion of Least squared errors [Kalman 1960]. The basic

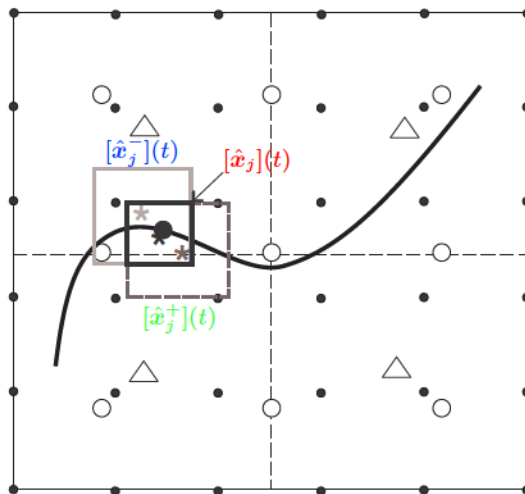


Figure IV.3: Interval-based localization

idea of the Kalman filter is to predict the unknown state using a given state equation, then to correct the predicted value by using some observations. In order to apply the Kalman filter to the localization problem, consider that the unknown states are the exact D -dimensions nodes' positions $\mathbf{x}_j(t)$ and their velocities $\mathbf{v}_j(t)$, that is, the state $\mathbf{X}_j(t)$ is set to $(\mathbf{x}_j(t)^T \mathbf{v}_j(t)^T)^T$. Assume that the state equation is the acceleration-based one and the observations are the 1st global positions estimates obtained using the radio-fingerprints map. The problem is then modeled by

$$\begin{aligned} \mathbf{X}_j(t) &= \mathbf{A}\mathbf{X}_j(t-1) + \mathbf{u}_j(t) + \zeta_j(t), \\ \hat{\mathbf{x}}_j^-(t) &= \mathbf{C}\mathbf{X}_j(t) + \varepsilon_j(t), \end{aligned} \quad (\text{IV.27})$$

where $\zeta_j(t) \sim \mathcal{N}(0, V)$ is the state equation noise of zero mean normal distribution with covariance matrix V of $(2D \times 2D)$ and $\varepsilon_j(t) \sim \mathcal{N}(0, R)$ is the observation noise of zero mean normal distribution with covariance matrix R of $(D \times D)$. \mathbf{A} is the state transition matrix, $\mathbf{u}_j(t)$ is the noisy control-input vector and \mathbf{C} is the observation matrix. \mathbf{A} and $\mathbf{u}_j(t)$ are determined by applying the mobility model, whereas \mathbf{C} is defined using the observation model.

IV.5.1 Kalman parameters definition

According to the problem definition, the matrix \mathbf{C} is a $(D \times 2D)$ matrix given by the identity D matrix \mathbf{I}_D , completed by zeros. The exact value of the covariance matrix of ε is not available. In order to approximate its value, a set of test positions are randomly generated and they are localized according to Section IV.3.1. Then, since the exact test positions are known, one computes the estimation error vector over each coordinate. The matrix \mathbf{R} is then estimated by computing the covariance of the D error vectors. This matrix is constant over all iterations and is the same for all nodes. The

IV. CLUSTERIZED LOCALIZATION

state equation is defined with respect to the mobility model. Considering the first-order acceleration model of Eq. IV.8, the corresponding kalman state equation is illustrated as,

$$\begin{aligned}\mathbf{X}_j(t) &= \begin{pmatrix} \mathbf{x}_j(t) \\ \boldsymbol{\nu}_j(t) \end{pmatrix} = A \begin{pmatrix} \mathbf{x}_j(t-1) \\ \boldsymbol{\nu}_j(t-1) \end{pmatrix} + \mathbf{u}_j(t) + \zeta_j(t), \\ &= \begin{pmatrix} \mathbf{I}_D & \Delta t \mathbf{I}_D \\ \mathbf{0}_D & \mathbf{I}_D \end{pmatrix} \mathbf{X}_j(t-1) + \begin{pmatrix} \Delta t^2 \boldsymbol{\gamma}_j(t) \\ \Delta t \boldsymbol{\gamma}_j(t) \end{pmatrix} + \zeta_j(t),\end{aligned}\quad (\text{IV.28})$$

where $\mathbf{0}_D$ is the $(D \times D)$ null matrix. As to the second-order model, the state equation is shown as,

$$\begin{aligned}\mathbf{X}_j(t) &= \begin{pmatrix} \mathbf{x}_j(t) \\ \boldsymbol{\nu}_j(t) \end{pmatrix} = A \begin{pmatrix} \mathbf{x}_j(t-1) \\ \boldsymbol{\nu}_j(t-1) \end{pmatrix} + \mathbf{u}_j(t) + \zeta_j(t), \\ &= \begin{pmatrix} \mathbf{I}_D & \Delta t \mathbf{I}_D \\ \mathbf{0}_D & \mathbf{I}_D \end{pmatrix} \mathbf{X}_j(t-1) + \begin{pmatrix} \frac{\Delta t^2}{2} \boldsymbol{\gamma}_j(t) \\ \Delta t \boldsymbol{\gamma}_j(t) \end{pmatrix} + \zeta_j(t).\end{aligned}\quad (\text{IV.29})$$

The state equation applying the third-order model is represented nevertheless as,

$$\begin{aligned}\mathbf{X}_j(t) &= \begin{pmatrix} \mathbf{x}_j(t) \\ \boldsymbol{\nu}_j(t) \end{pmatrix} = A \begin{pmatrix} \mathbf{x}_j(t-1) \\ \boldsymbol{\nu}_j(t-1) \end{pmatrix} + \mathbf{u}_j(t) + \zeta_j(t) \\ &= A \mathbf{X}_j(t-1) + \mathbf{u}_{a,j}(t-1) + \mathbf{u}_{b,j}(t) + \zeta_j(t), \\ &= \begin{pmatrix} \mathbf{I}_D & \Delta t \mathbf{I}_D \\ \mathbf{0}_D & \mathbf{I}_D \end{pmatrix} \mathbf{X}_j(t-1) + \begin{pmatrix} \frac{\Delta t^2}{3} \boldsymbol{\gamma}_j(t-1) \\ \frac{\Delta t}{2} \boldsymbol{\gamma}_j(t-1) \end{pmatrix} + \begin{pmatrix} \frac{\Delta t^2}{6} \boldsymbol{\gamma}_j(t) \\ \frac{\Delta t}{2} \boldsymbol{\gamma}_j(t) \end{pmatrix} + \zeta_j(t).\end{aligned}\quad (\text{IV.30})$$

IV.5.2 Algorithm using Kalman

Resolving the problem according to the Kalman filter consists of two phases. First, the nodes positions are predicted using the previous estimate and the state equation.

$$\hat{\mathbf{X}}_j^+(t) = A \hat{\mathbf{X}}_j(t-1) + \mathbf{u}_j(t), \quad (\text{IV.31})$$

where $\mathbf{u}_j(t)$ is computed beforehand at each iteration as shown in the previous section, $\hat{\mathbf{X}}_j(t-1)$ is the final previous estimate and $\hat{\mathbf{X}}_j(0)$ is assumed to be known. Then the predicted estimate covariance reflecting the accuracy of the state estimate is updated as follows,

$$\mathbf{P}_j^+(t) = A \mathbf{P}_j(t-1) A^T + Q(t) = \mathbf{P}_j(t-1) + Q(t), \quad (\text{IV.32})$$

where \bullet^T denotes the transpose of \bullet , $\mathbf{P}_j(t-1)$ is the final previous estimate covariance and $\mathbf{P}_j(0)$ is equal to the null matrix since the state is initially known. The matrix $Q(t)$ is the covariance of $\mathbf{X}_j(t)$ given $\mathbf{X}_j(t-1)$, that is

$$\begin{aligned}Q(t) &= \text{cov}(\mathbf{X}_j(t) | \mathbf{X}_j(t-1)) = \text{cov}(\mathbf{u}_j(t) + \zeta_j(t)) \\ &= \text{cov}(\mathbf{u}_j(t)) + \text{cov}(\zeta_j(t)) \\ &= \text{cov}(\mathbf{u}_j(t)) + V,\end{aligned}\quad (\text{IV.33})$$

where $cov(\cdot)$ computes the covariance matrix of its argument. Now if we consider the first-order model, and since each coordinate acceleration noise has a zero-mean normal distribution with a standard deviation σ_γ , and the D coordinates are statistically independent, the covariance matrix $Q(t)$ is given by

$$\begin{aligned} Q(t) &= V + cov \left(\begin{pmatrix} \Delta t^2 \gamma_j(t) \\ \Delta t \gamma_j(t) \end{pmatrix} \right) \\ &= V + \begin{pmatrix} \Delta t^4 \sigma_\gamma^2 \mathbf{I}_D & \mathbf{0}_D \\ \mathbf{0}_D & \Delta t^2 \sigma_\gamma^2 \mathbf{I}_D \end{pmatrix}. \end{aligned} \quad (\text{IV.34})$$

By taking the second-order model, the covariance matrix $Q(t)$ is given by

$$\begin{aligned} Q(t) &= V + cov \left(\begin{pmatrix} \frac{\Delta t^2}{2} \gamma_j(t) \\ \Delta t \gamma_j(t) \end{pmatrix} \right) \\ &= V + \begin{pmatrix} \frac{\Delta t^4}{4} \sigma_\gamma^2 \mathbf{I}_D & \mathbf{0}_D \\ \mathbf{0}_D & \Delta t^2 \sigma_\gamma^2 \mathbf{I}_D \end{pmatrix}. \end{aligned} \quad (\text{IV.35})$$

As to the third-order acceleration model, the covariance matrix $Q(t)$ is given by

$$\begin{aligned} Q(t) &= V + cov(\mathbf{u}_{a,j}(t-1) + \mathbf{u}_{b,j}(t)) \\ &= V + cov \left(\begin{pmatrix} \frac{\Delta t^2}{3} \gamma_j(t-1) \\ \frac{\Delta t}{2} \gamma_j(t-1) \end{pmatrix} + \begin{pmatrix} \frac{\Delta t^2}{6} \gamma_j(t) \\ \frac{\Delta t}{2} \gamma_j(t) \end{pmatrix} \right) \\ &= V + \begin{pmatrix} \frac{5\Delta t^4}{36} \sigma_\gamma^2 \mathbf{I}_D & \mathbf{0}_D \\ \mathbf{0}_D & \frac{\Delta t^2}{2} \sigma_\gamma^2 \mathbf{I}_D \end{pmatrix}. \end{aligned} \quad (\text{IV.36})$$

The predicted quantities are then corrected using the observation equation as follows,

$$\begin{aligned} \hat{\mathbf{X}}_j(t) &= \hat{\mathbf{X}}_j^+(t) + \mathbf{H}_j(t)(\hat{\mathbf{X}}_j^-(t) - C \cdot \hat{\mathbf{X}}_j^+(t)) \\ &= \hat{\mathbf{X}}_j^+(t) + \mathbf{H}_j(t)(\hat{\mathbf{X}}_j^-(t) - \hat{\mathbf{X}}_j^+(t)), \\ \mathbf{P}_j(t) &= (\mathbf{I}_D - \mathbf{H}_j(t)C)\mathbf{P}_j^+(t) \\ &= (\mathbf{I}_D - \mathbf{H}_j(t))\mathbf{P}_j^+(t), \end{aligned} \quad (\text{IV.37})$$

where $\mathbf{H}_j(t)$ is the optimal Kalman gain given by

$$\begin{aligned} \mathbf{H}_j(t) &= \mathbf{P}_j^+(t)C^T(C\mathbf{P}_j^+(t)C^T + R)^{-1} \\ &= \mathbf{P}_j^+(t)(\mathbf{P}_j^+(t) + R)^{-1}. \end{aligned} \quad (\text{IV.38})$$

Note that the correction of Eq. IV.37 is only valid for the optimal Kalman gain. Using other gain expressions needs more complicated correction equations.

IV.6 Simulation results

This section illustrates the performance of the proposed method using the interval analysis or the Kalman filter. To this end, we consider a $100m \times 100m$ two-dimensional

IV. CLUSTERIZED LOCALIZATION

surveillance area ($D = 2$). Only the localization problem of one mobile node is considered, since each node localizes itself using only anchors information, independently from other nodes, and the j index will be withdrawn when needed in the following. It is worth noting that all simulations are done on Matlab 7.11.2(R2010b). Two trajectories, selected from the sixth of chapter III, of 100 points with $\Delta t = 1s$ are generated using analytic expressions for both coordinates as functions of time. The trajectories are illustrated once more in Figure IV.4 and Figure IV.5, whereas their respective accelerations are given in the top plots and in the bottom plots of Figure IV.6. Random gaussian

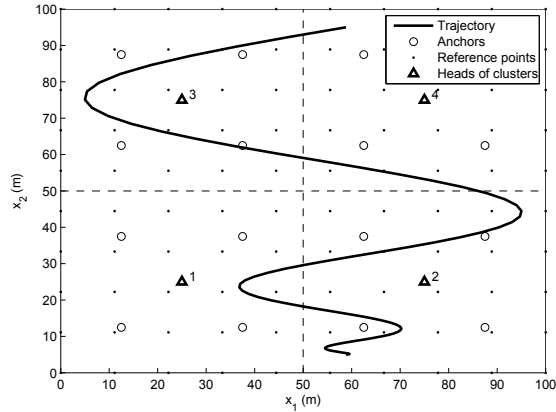


Figure IV.4: First trajectory

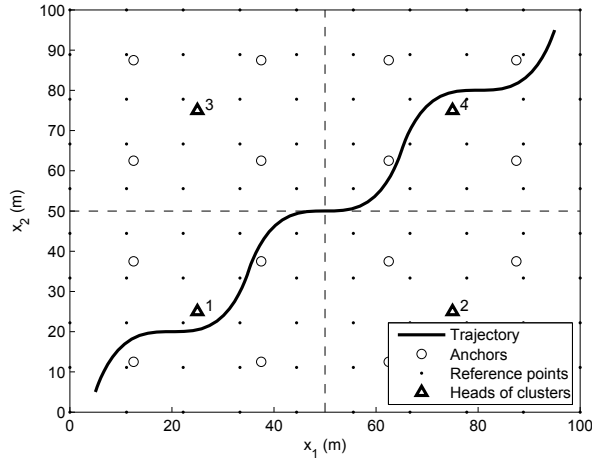


Figure IV.5: Second trajectory

white noises with standard deviation σ_γ are added to the accelerations. The surveillance area is divided into N_C equal clusters and the heads of clusters are positioned at the centers of the clusters. N_A anchors and N_P reference points are positioned in a regularly distributed manner in the surveillance area, leading to equal

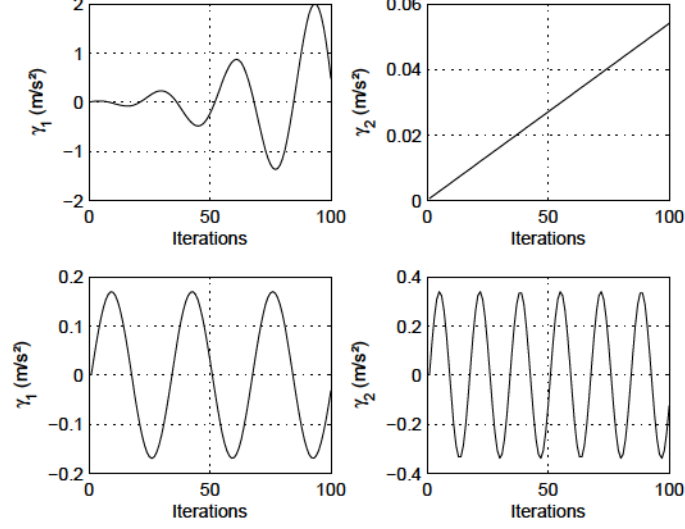


Figure IV.6: Acceleration signals for the first trajectory in the top plots and for the second trajectory in the bottom plots

$\frac{N_P}{N_C}$ reference points in each cluster. Figure IV.4 and Figure IV.5 show 16 anchors in black circles, 100 reference points in black points, 4 clusters and 4 heads of clusters in black triangles. The pathloss model [Medeisis et Kajackas 2000] is used to generate the RSSI values, given as follows,

$$\begin{aligned}\xi_{P(c),n,i} &= \xi_0 - 10n_P \log_{10} \|\mathbf{a}_i, \mathbf{p}_{(c),n}\|, \\ \xi_{X_i}(t) &= \xi_0 - 10n_P \log_{10} \|\mathbf{a}_i, \mathbf{x}(t)\|, \\ \xi_{H(c)}(t) &= \xi_0 - 10n_P \log_{10} \|\mathbf{h}_{(c)}, \mathbf{x}(t)\|,\end{aligned}\tag{IV.39}$$

where all RSSIs are in dBm, ξ_0 is the transmission power in dBm at $1m$ from the emitter, set to $100dBm$, and n_P is the pathloss parameter, set to 4. Random gaussian white noises with standard deviation σ_ξ are then added separately to all RSSIs.

IV.6.1 Illustration of the proposed method

Consider that the surveillance area is divided to $N_C = 4$ clusters, setting the number of heads of clusters to 4. Set the number of anchors N_A to 16, the number of reference points N_P to 100 and the number of neighbors K to be considered to 3. For the weights, use the inverting distance scheme given in Eq. IV.4, with α set to 2, and the standard deviations of noises σ_γ and σ_ξ are set to $0.01m/s^2$ and $1dBm$ respectively, which are around 2% and 10% of the deviations of real signals. The covariance matrix V of state equation noise is set to zero. Here the first scheme for the acceleration model is selected, however comparison between models will be illustrated further later. Figure IV.7 shows the estimated trajectories using the interval analysis (DWKNN-I) and the Kalman filter (DWKNN-K) and the estimated boxes for the first trajectory, whereas Figure IV.8 shows

IV. CLUSTERIZED LOCALIZATION

the results for the second trajectory. In both plots, the estimated trajectories follow very well the real trajectories.

Let the estimation error be the average distance between the estimated positions and the real ones over all time steps. Table IV.1 shows the estimation errors in meters and the average computation time in milliseconds per iteration obtained on the first trajectory using only fingerprinting-based method (DWKNN), only acceleration model (Acc.), the combined method based on intervals (DWKNN-I) and the combined method based on Kalman (DWKNN-K). The first, the second and the third orders models for accelerations are considered in the three latter methods. As expected, combining information leads to more accurate results at the cost of the increase of the estimation time, that remains very low. Moreover, the performances of the Kalman filter and the interval analysis are very close with a slightly better estimation accuracy with the Kalman filter. The table also shows that the third-order acceleration model performs much greater than the others. This is due to the assumption of null or constant accelerations made by the first-order and the second-order models respectively, which is not the case for this trajectory. In the following, both the second-order and the third-order acceleration models will be considered in the proposed method and the only-acceleration-based one. Moreover, we will only consider the first trajectory for performance illustration, since it is more complex than the second one and closer to an unexpected travel path.

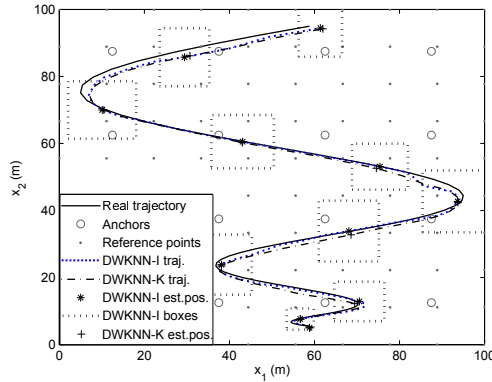


Figure IV.7: Estimated results using the interval analysis and the Kalman filter for the first trajectory

IV.6.2 Influences of K , N_C , N_A , N_P and weight parameters

In this section, we illustrate the performances of the proposed method with respect to the parameters K , N_C , N_A , N_P and the weight parameters of the algorithm. To this end, vary at first the number of neighbors K considered in the RSSI-based computation from 1 to 10. Set N_A to 25, N_P to 100 and N_C to 4. The noise standard deviations are set to $\sigma_\gamma = 0.01m/s^2$ and $\sigma_\xi = 1dBm$. Note that the same noisy data are used in all experiments with different values of K . Figure IV.9 illustrates the estimation

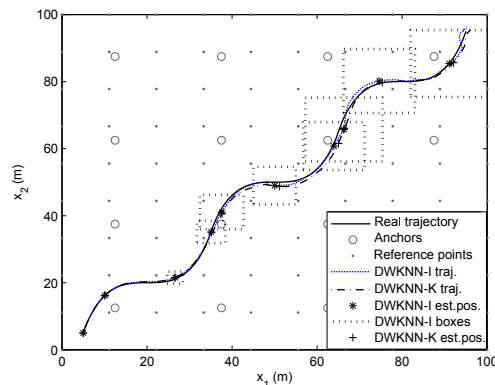


Figure IV.8: Estimated results using the interval analysis and the Kalman filter for the second trajectory

Table IV.1: Comparison of the proposed method to only fingerprinting-based and only acceleration-based methods

	DWKNN	Acc.	DWKNN-I	DWKNN-K
Error / Acc. 1st-order mod.	2.32m	3.94m	2.29m	2.28m
Error / Acc. 2nd-order mod.	2.32m	3.48m	1.73m	1.53m
Error / Acc. 3nd-order mod.	2.32m	3.17m	1.59m	1.41m
Time / Acc. 1st-order mod.	0.055ms	0.0067ms	0.12ms	0.11ms
Time / Acc. 2nd-order mod.	0.055ms	0.0071ms	0.12ms	0.12ms
Time / Acc. 3nd-order mod.	0.055ms	0.0073ms	0.13ms	0.13ms

errors in the left plot and the average computation times in the right plot as functions of K , for the proposed method using intervals (DWKNN-I) and Kalman (DWKNN-K), the acceleration-based one (Acc.) and the fingerprinting-based one (DWKNN) with both the second-order mobility model (mod2) in black and the third-order mobility model (mod3) in blue. It is observed that in all cases, the proposed method leads to more accurate estimates than the other methods, with slightly better performances with Kalman. Also, the performances are much better using the third-order model. The computation times increase using combining methods however they remain very low. The best results of the proposed method are obtained for $K = 3$. Consequently we choose $K = 3$ for the following simulations.

We verify then the influence of number of clusters N_C . We thus vary the number of clusters N_C for our clustered algorithm from 1^2 to 5^2 , where N_A is set to 25, N_P to 100 and K to 3. The noise standard deviations are set to $\sigma_\gamma = 0.01m/s^2$ and $\sigma_\xi = 1dBm$. Figure IV.10 shows the curves of estimation errors and computation times with respect to the number of clusters N_C . The estimation errors of our method vary slightly with the variation of the number of clusters, whereas the computation time increases with the increase of N_C . This is due to the fact that the time shown in the plot is the total time

IV. CLUSTERIZED LOCALIZATION

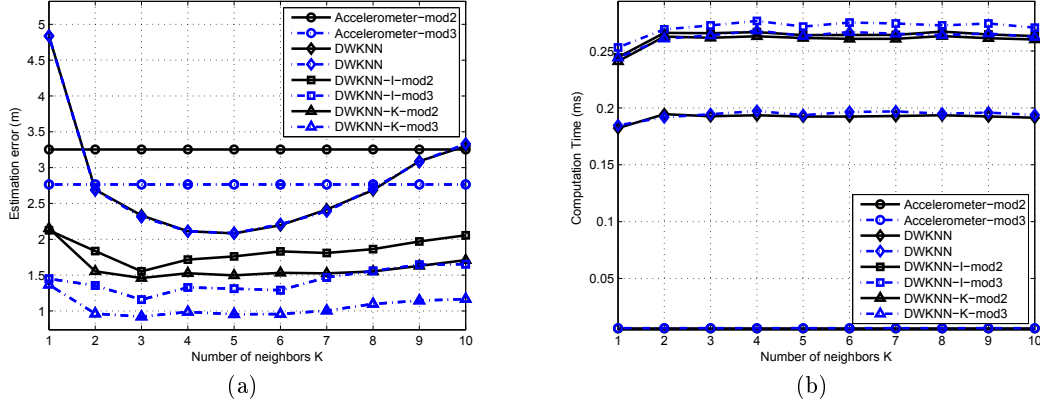


Figure IV.9: Estimation errors (a) and times (b) as functions of K

including all heads of clusters computation. By dividing the computation times by the number of heads, one could obtain the average computation time per head, which would decrease with the increase of N_C due to the decrease of the number of reference positions per cluster. The plot shows also that with several clusters, the clusterized localization method is less accurate than the centralized method with only one cluster, however with four clusters the difference in errors is very small while gaining in robustness and energy consumption. In the following, N_C is set to 4 since with this value we have the best results with our proposed method.

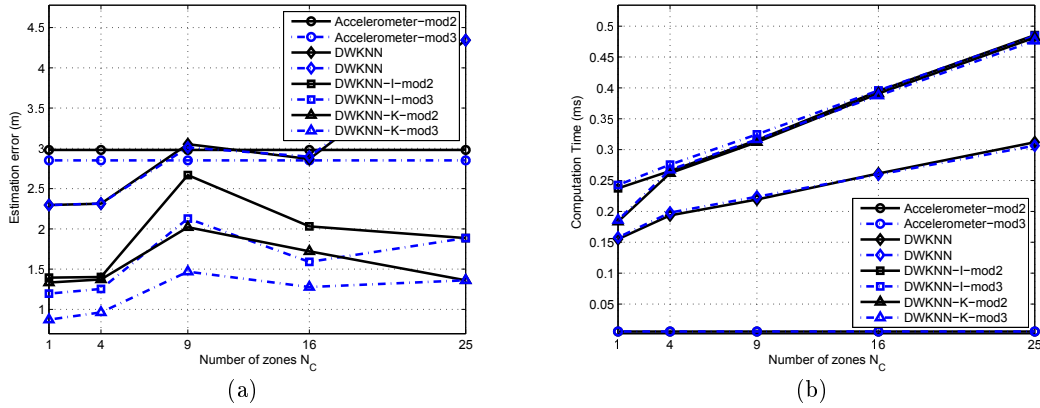


Figure IV.10: Estimation errors (a) and times (b) as functions of N_C

Afterwards, we vary the number of anchors N_A from 2^2 to 20^2 , while setting K to 3, N_C to 4, N_P to 100 and σ_γ and σ_ξ to $0.01m/s^2$ and $1dBm$ respectively. Figure IV.11 shows the curves of the estimation errors and the computation times with respect to the number of anchors N_A . As expected, the estimation error decreases with the increase of

the number of anchors, at the cost of the computation time that slightly increases. One could also see that our two approaches outperform the other methods especially with the third-order mobility model. The irregular error around $N_A = 81$ is equally caused by the intimacy between some reference points and the anchor due to the logarithmic expression of the RSSI/distance as analyzed in Chapter III.4.4.

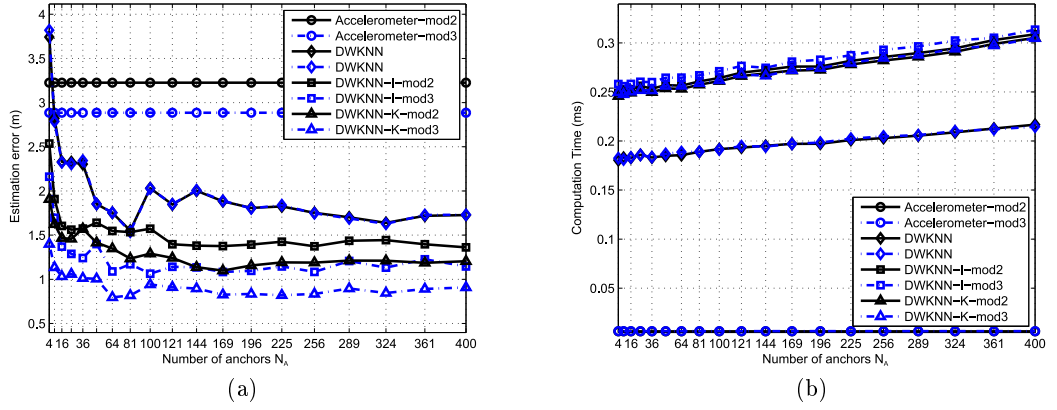


Figure IV.11: Estimation errors (a) and times (b) as functions of N_A

The number of reference positions N_P is varied afterwards from 6^2 to 20^2 , with K equal to 3, N_C to 4, N_A to 25, and σ_γ and σ_ξ equal to $0.01m/s^2$ and $1dBm$ respectively. Figure IV.12 shows the curves of estimation errors and computation times with respect to N_P . The plot shows that the estimation error decreases with the increase of the number of considered reference positions, at the cost of the computation time. The plot demonstrates also that the proposed method using either intervals or Kalman outperforms the other methods in all cases especially with its third-order model.

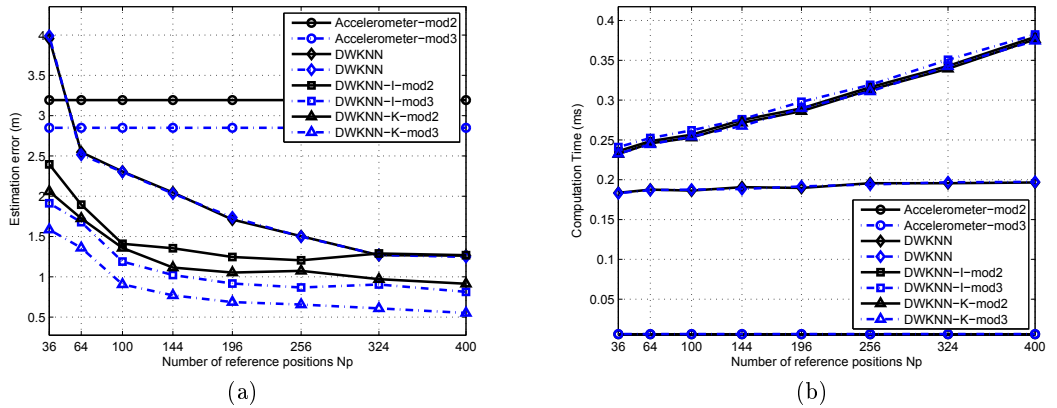


Figure IV.12: Estimation errors (a) and times (b) as functions of N_P

IV. CLUSTERIZED LOCALIZATION

We aim then at the study of the impact of the weights expression used in the computation of the 1st estimate based on the nearest neighbor algorithm and the RSSI fingerprints. Firstly consider the inverting distance scheme given in Eq. IV.4, and vary the parameter α from 1 to 5. Set K to 3, N_C to 4, N_A to 25, N_P to 100, and σ_γ and σ_ξ to $0.01m/s^2$ and $1dBm$ respectively. Then consider the exponential weight scheme given in Eq. IV.5, and vary the parameter α from 1 to 3. Figure IV.13 shows the estimation errors curves as functions of the parameters values, where the labels Inv-1,...,Inv-5 mean that we are taking the first scheme with $\alpha = 1, \dots, 5$ and the labels Exp-1,...,Exp-3 mean that we are taking the second scheme with $\alpha = 1, 2, 3$. The plot shows that with the inverting distance scheme, our proposed method and the only-fingerprinting-based method outperforms their versions with the exponential scheme. The best results are obtained with Eq. IV.4 using $\alpha = 2$ with the third-order mobility model. We thus take this configuration in the following simulations.

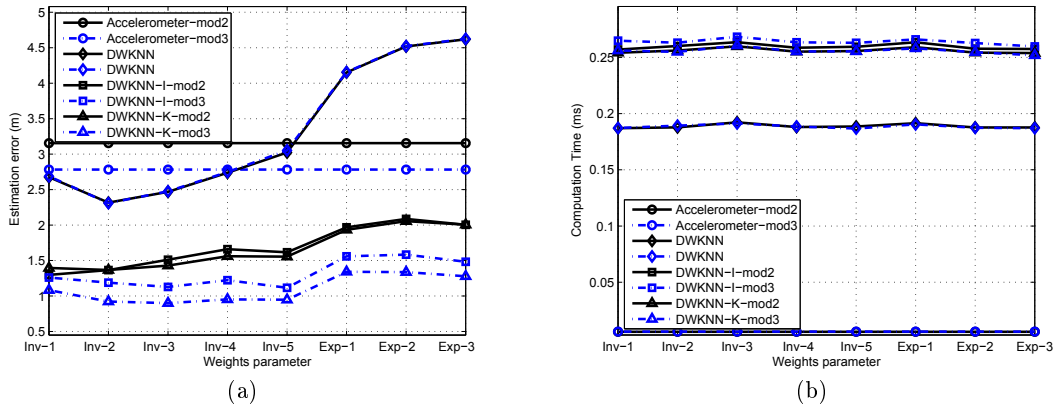


Figure IV.13: Estimation errors (a) and times (b) as functions of the weight parameter

IV.6.3 Impact of additional noises

This section aims at studying the impact of the noises standard deviations, σ_γ and σ_ξ on the localization accuracy. At the very first, we vary the standard deviation of the acceleration noise σ_γ from $0.005m/s^2$ to $0.02m/s^2$ with $K = 3$, $N_C = 4$, $N_A = 4$, $N_P = 36$ and $\sigma_\xi = 0dBm$. The left plot of Figure IV.14 shows the curves of the estimation errors obtained with the acceleration-based (Acc.), the fingerprinting-based (DWKNN), the interval-based (DWKNN-I) and the Kalman-based (DWKNN-K) approaches with respect to σ_γ with the second-order (in black) and the third-order mobility models (in blue). As expected, the estimation error increases with the increase of the noise in all of the three methods using accelerations information. However, the combined approaches remain always more accurate than the other approaches. Then we vary the standard deviation of RSSI noise σ_ξ from $0.1dBm$ to $4.5dBm$ with $K = 3$, $N_C = 4$, $N_A = 16$, $N_P = 100$ and $\sigma_\gamma = 0.01m/s^2$. The estimation errors are illustrated in the right plot

of Figure IV.14 with respect to σ_ξ . The proposed two-approaches method outperforms the other methods in all cases, as well. It is worth noting that the errors either over accelerations or RSSIs are generated 50 times for each value of the standard deviation and the errors are averaged over the 50 simulations.

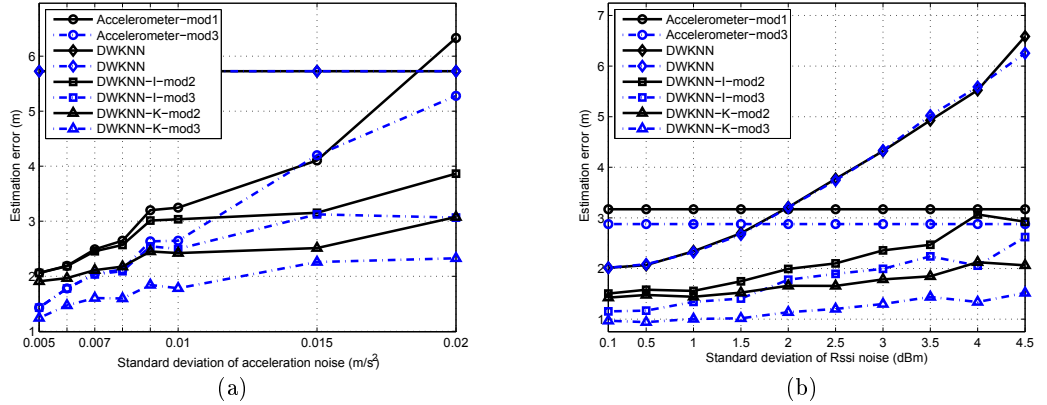


Figure IV.14: Estimation errors as functions of noise standard deviations

Next analysis aims afterwards to illustrate the performance of our proposed method with respect to the distribution of acceleration noise. Indeed, in the proposed method it assumes that the acceleration noise is gaussian with zero-mean and known standard deviation σ_γ , taken here equal to $0.01m/s^2$, that is the algorithm considers $\sigma_\gamma = 0.01m/s^2$. However, the acceleration noise could have, in real, different distribution or higher standard deviations at times. For this reason, it is suggested first to generate a gaussian acceleration noise with zero-mean and a standard deviation equal to $0.01m/s^2$ for the first 40 steps, $0.02m/s^2$ for the following 30 steps, then $0.03m/s^2$ for the rest, whereas our combined algorithms still take $\sigma_\gamma = 0.01m/s^2$. Table IV.2 shows the estimation errors for the acceleration-based (Acc.), the fingerprinting-based (DWKNN) and both combined method using interval (DWKNN-I) or Kalman (DWKNN-K) for this case. It also shows the errors for the case where the acceleration have a noise of $0.01m/s^2$ as taken in the algorithms. It is considered for all the simulations of this part $\sigma_\xi = 1dB$, $N_A = 16$ and $N_P = 100$. The table shows that the estimation errors increase with higher noises, compared to the case of gaussian noise with correct known standard deviation of $0.01m/s^2$, however the method remains accurate with its both schemes. We then take a gaussian noise with a standard deviation equal to $0.02m/s^2$ for all the steps (which is very high compared to the accelerations values), while the algorithms still take $0.01m/s^2$. Also the estimation errors increase, but remain acceptable. Then, we vary the distribution of the acceleration noise, but we kept the standard deviation equal to $0.01m/s^2$. Thus we take the Extreme Value distribution (EV), the Student distribution (St) and the F-distribution (F). We also generate acceleration errors with less standard deviation equal to $0.001m/s^2$ with Extreme Value and F distributions. The estimation errors are also shown in Table IV.2. According to these results, the method performs

IV. CLUSTERIZED LOCALIZATION

well even with different distribution of the acceleration noise. Moreover, the Kalman-based approach performs slightly better than the interval-based one, especially with the third-order mobility model.

Table IV.2: Impact of acceleration noises with $\sigma_\gamma = 0.01m/s^2$, $\sigma_\xi = 1dB$, $N_A = 16$, and $N_P = 100$

Distribution / real std. dev.	Acc.		DWKNN	DWKNN-I		DWKNN-K	
	mod2	mod3		mod2	mod3	mod2	mod3
Gauss / 0.01	3.1993	2.5419	2.3207	1.6079	1.1704	1.4665	0.8908
Gauss / 0.01,0.02, 0.03	3.4953	3.0145	2.3049	1.7097	1.3144	1.5550	1.0851
Gauss / 0.02	6.1020	6.0817	2.3334	2.2612	1.5810	2.0684	1.3161
EV / 0.001	1.7693	0.7984	2.3246	1.1234	0.5894	1.1711	0.4550
EV / 0.01	8.5138	7.7483	2.3246	2.3234	2.3036	2.2446	1.4720
St / 0.01	3.0920	2.8817	2.3060	1.5207	1.2100	1.4125	0.9438
F / 0.001	1.5500	0.3554	2.3195	1.0631	0.5044	1.1562	0.4578
F / 0.01	3.4487	3.2038	2.3195	1.5773	1.2141	1.4632	0.9782

Table IV.3: Impact of the orientation noise with $\sigma_\gamma = 0.01m/s^2$, $\sigma_\xi = 1dB$, $N_A = 16$, and $N_P = 100$

σ_θ / real acc. std. dev.	Acc.		DWKNN	DWKNN-I		DWKNN-K	
	mod2	mod3		mod2	mod3	mod2	mod3
0.01 / 0.01	3.1781	2.7278	2.2784	1.6813	1.2381	1.4455	0.9678
0.05 / 0.01	3.9786	3.6261	2.3399	1.8394	1.5611	1.6445	1.2564
0.08 / 0.01	4.1753	4.5114	2.3274	1.9434	1.9543	1.7822	1.5378
0.1 / 0.01	4.9913	3.7226	2.3153	2.2337	1.5996	1.9076	1.4010
0.15 / 0.01	6.2572	5.7387	2.3171	2.3002	2.2711	2.0581	1.9896
0.2 / 0.01	6.9692	6.8935	2.3263	2.3165	2.1399	2.2768	2.0732
0.1 / 0.01, 0.02, 0.03	5.0483	4.4645	2.3348	2.2780	1.8664	2.0272	1.6127
0 / 0.01	3.1248	2.6508	2.3164	1.5854	1.2273	1.4559	0.9946

In what follows, it is recommended also to consider the rotation problem of the sensor nodes. Thus assume that the node starts with an orientation equal to 0, then it rotates following its trajectory, that is the orientation of the node varies with its instantaneous velocity. Then add a zero-mean gaussian noise to the orientation with different standard deviations σ_θ going from 0.01 to 0.2 radian and we run our algorithms while considering $\sigma_\gamma = 0.01m/s^2$. The standard deviation of the acceleration noise is first set to the known value $0.01m/s^2$, then set it to 0.01 for the first 40 steps, 0.02 for the following 30 steps and 0.03 for the next 30 steps. The estimation errors are given in Table IV.3, where it also shows the error where no noise over orientation is considered. The table shows that the error increases with the increase of σ_θ , however it is still acceptable. This increase was expected since the algorithm considers that the orientation noise is null, while slight variation in the orientation increases significantly the acceleration noises. A solution to this problem consists of increasing automatically σ_γ for the combined approaches. The

results also show that the third-order model yields better performances. In the following, we will consider the method with the third-order mobility model in its Kalman-based approach.

IV.6.4 Comparison to range-based methods

After the performance analysis over all the parameters considered in the proposed method, it is then preferred to compare the proposed method with some RSSI range-based methods. Such methods estimate the distances separating the node from the anchors using measured RSSI and the pathloss model given in Eq. IV.39. The distances are then combined using lateration or min-max algorithm as shown in [Priwgharm et Chemtanomwong 2011] or particle filter as in [Farmani et al. 2012]. The advantage of such methods is that anchors could be mobile, whereas in our method anchors are stationary. For the lateration algorithm of [Priwgharm et Chemtanomwong 2011], the position of the node at each step is computed according to the least squares algorithm, whereas for the min-max algorithm, boxes over anchors positions are generated using the distances estimates, then the boxes are intersected leading to a smaller box, whose center corresponds to the estimate of the node's position at each step. To compare the proposed method to these methods, consider 16 anchors in the network of $100m \times 100m$. Table IV.4 shows the estimation errors in meters of lateration-based method (Lat) and min-max (MM) algorithm for different values of the standard deviation of RSSI errors σ_ξ . The table shows also the estimation errors of the proposed method using Kalman (DWKNN-K) and the fingerprinting method without using accelerations (DWKNN) with 100 reference positions and the acceleration noise standard deviation of $\sigma_\gamma = 0.01m/s^2$. In [Farmani et al. 2012], the nodes are assumed to be fixed, with one mobile anchor. At each time step, nodes estimate their distances to the mobile anchor and update their positions using the particle filter. Then, nodes exchange signals with their neighbors and correct their estimated positions using their inter-node distances with the Extended Kalman Filter. In order to compare this algorithm to the proposed method where nodes are mobile, localize the nodes at each step by moving the anchor over all its path. To this end, use Hilbert curve with order equal to 4 for the anchor as proposed in [Farmani et al. 2012] for the $100m \times 100m$ surveillance area. Adopt 50 nodes and the average estimation errors over all the nodes of this technique (PF) are shown also in Table IV.4. It is worth noting that, due to the randomness of the data, simulations are generated 200 times then average errors are given in the table. It is obvious that the proposed method outperforms the MM and PF methods in terms of accuracy, even with its basic form without the use of accelerations. For very low RSSI errors, the lateration method performs better, however this method is not robust to noise, since its estimation error highly increases with an increasing noise. Table IV.5 shows the estimation errors of Lat, MM, DWKNN and DWKNN-K while varying the number of anchors in the network. The proposed method outperforms the other methods, as well.

IV. CLUSTERIZED LOCALIZATION

Table IV.4: Comparison of the proposed method to range-based methods with $N_A = 16$

σ_ξ (dBm)	0.001	0.01	0.1	1	5
Lat	0.0048	0.0480	0.4765	4.8561	27.0405
MM	4.3678	4.3696	4.4654	6.0668	14.5170
PF	2.5997	3.6132	3.6806	10.3695	19.2065
DWKNN	2.0088	2.0089	2.0101	2.3222	7.1695
DWKNN-K	0.9785	0.9806	0.9823	1.0007	1.5400

Table IV.5: Comparison of the proposed method to range-based methods with $\sigma_\xi = 1dB$

N_A	4	9	16	25
Lat	5.4508	5.0054	4.7558	4.6723
MM	15.7203	9.0669	6.1119	5.3918
DWKNN	3.8013	2.7266	2.3442	2.2901
DWKNN-K	1.2491	1.1105	1.0058	0.9063

IV.6.5 Comparison to a connectivity-based method

As discussed in chapter I, the localization approach could be categorized into several groups, such as range-based and connectivity-based. This section shows the comparison results over the proposed method and a connectivity-based method, using Monte-Carlo [Baggio et Langendoen 2006]. In this approach, each node compares its online RSSI measurements at each iteration to a certain threshold ξ_r , associated to a distance r . If the RSSI of a signal emitted by an anchor i is higher than ξ_r , the distance separating the anchor from the node at this time-step is less than r . In this case, the anchor i is assumed to be detected by the node, leading to a disk constraint centered on the anchor and having r as radius, including the node position. The solution of the problem is then set by intersecting all disks of detected anchors. In order to define it using Monte-Carlo, N particles are generated in the way to satisfy all detected anchors constraints. These particles are included in the intersection area of all disks. Finally, the estimated position is given by the barycenter of all generated particles. In order to compare this approach to the proposed method, set the number of anchors N_A to 49, the number of reference points N_P to 196, the number of neighbors K to 3, the standard deviations of noises σ_γ and σ_ξ to $0.001m/s^2$ and $0.1dBm$ respectively. Set also the number of particles N to 100 and the range r to $15m$, leading to a power threshold equal to $52.96dBm$ according to the noiseless pathloss model with $\xi_0 = 100dBm$ and $n_P = 4$. With this value of r , an average of 3.33 anchors is detected by the node at each time-step. The connectivity-based method yields an estimation error of $3.7989m$, compared to $1.0554m$ and $0.9031m$ with the proposed method based on intervals and Kalman respectively. The average computation times are equal to $2.167ms$, compared to $0.1609ms$ and $0.1547ms$ respectively. Obviously the proposed method yields more accurate results with much less time, compared to the connectivity-based one using

Monte-Carlo.

Moreover, the performance of the connectivity-based one highly depends on the number of anchors considered in the network. Figure IV.15 shows the relative estimation errors of the Kalman-based method over the connectivity-based one, with respect to the number of anchors in the network. It also shows the relative computation times obtained with Kalman over those using connectivity. For this figure, vary the number of anchors from 2^2 to 10^2 , set also N to 100, the range r to 15, N_P to 100, N_C to 4, K to 3, σ_γ to $0.01m/s^2$ and σ_ξ to $1dBm$ as for Figure IV.11. Simulations are run 50 times and the illustrated errors and computation times are averaged over the 50 runs. As expected, the estimation error of the connectivity-based method decreases with the increase of the number of the considered anchors, and this with a higher speed compared to the proposed method. That is why the error ratio increases but it stays less than 1. Also, the computation time per time-step of the connectivity one highly increases, while the Kalman computation time slightly varies, leading to a decrease of the time ratio.

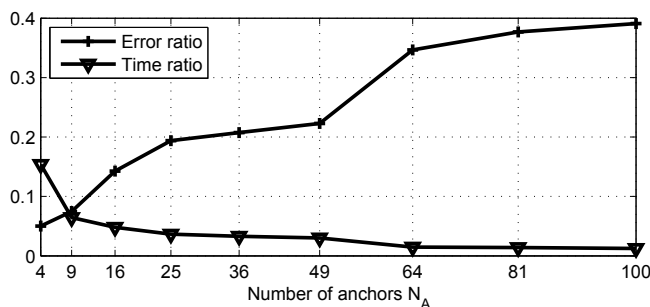


Figure IV.15: Ratios of estimation errors and computation times of the Kalman-based method over the connectivity-based one, with respect to anchors

IV.6.6 Comparison to fingerprint-based methods

In this section, comparison is made specially between the proposed method and other existing methods based on fingerprinting using RSSI. Some of these works use the Nearest-Neighbors algorithm (NN), the K-Nearest-Neighbors (KNN) [Navarro *et al.* 2010; Chuenurajit *et al.* 2013] and the Fuzzy-K-Nearest-Neighbors (FKNN) [Rozyyev *et al.* 2012]. Such methods determine the final estimation by selecting single nearest reference points for NN, by averaging the positions of the nearest reference points for KNN or by combining the nearest positions using fuzzy logic weights for FKNN. [Liu *et al.* 2012] and [Suroso *et al.* 2011] proposed localization techniques based respectively on Fuzzy Clusters (FC) and Fuzzy C-Means clustering (FCM). Using the fuzzy logic, these methods set the fingerprints into clusters, define for each mobile node at each time-step its cluster, then determine its position using the NN within its cluster. Other fingerprinting techniques use the learning process to define nodes positions. Based on the fingerprinting database, such methods define a model that takes the RSSI as input and yields the estimated positions as output. Learning such a model is performed using

IV. CLUSTERIZED LOCALIZATION

Table IV.6: Comparison to fingerprinting-based methods with $N_A = 16$ and $N_P = 100$ while varying σ_ξ

σ_ξ	0.001	0.01	0.1	1	5
NN	5.0204	5.0204	5.0228	5.1305	9.1649
KNN	3.1948	3.1993	3.1998	3.3092	6.5234
FKNN	3.1646	3.1656	3.1672	3.2999	6.3913
FC	3.8522	3.8854	3.8890	4.1691	9.5353
FCM	5.1772	5.2849	5.4013	5.2143	9.1850
NNet	3.2227	4.7924	4.8791	7.9965	17.9818
SVM	0.6573	0.74681	0.7681	2.8302	10.8531
RR	0.6298	0.7155	0.7359	2.7115	8.4821
AL	6.5124	6.4895	6.3618	6.6529	10.4949
DWKNN	2.0056	2.0087	2.0087	2.3315	7.0069
DWKNN-K	0.9958	0.9631	0.9905	1.0501	1.4992

a feed-forward Neural Network (NNet) in [Gogolak *et al.* 2011] or kernel-based machine learning methods, such as Support Vector Machines (SVM) in [Farjow *et al.* 2011] or Ridge Regression (RR) in [Mahfouz *et al.* 2013]. [Koyuncu *et al.* 2013] proposes an Analytical Localization method (AL), that uses the distances between the offline positions and the anchors to generate the fingerprinting database, instead of measuring RSSIs. Then the node while moving uses its collected RSSIs to generate a characteristic vector, that would be compared to the fingerprints to localize itself.

Table IV.6 shows the estimation errors for all these methods in addition to the proposed method with 16 anchors in the network, 100 reference positions, acceleration noise standard deviation equal to $0.01m/s^2$, 4 clusters and a varying standard deviation of RSSI noise. Our proposed method is more accurate than all methods except for the kernel-based methods (RR) and (SVM) with very low noise on RSSI. This was expected since with almost no noise, RR and SVM with their non-linear aspect learn exactly the pathloss model, that would be used later on, with very low additional perturbation, to simulate the data. However with higher RSSI noise, where we are closer to setups of real environments, our methods are more robust yielding more accurate results. Also, note that the learning and the cluster-based methods are very time-consuming, compared to the proposed methods. Table IV.7 shows the estimation errors while varying the number of anchors with $\sigma_\xi = 1dB$, $N_P = 100$, 4 clusters and $\sigma_\gamma = 0.01m/s^2$. Here also our methods are more accurate than the others. Table IV.8 shows the estimation errors with different values of the number of the reference positions. It is worth noting that with $N_P = 16$ and 4 clusters, only 4 reference positions are considered per cluster, which is very low for an accurate localization. This is the reason why it is obtained an estimation error around $11m$ with DWKNN, but this error, even high, is corrected to $1.7m$ with the acceleration information in DWKNN-K. The proposed method outperforms the other methods in terms of accuracy. Note that all simulations are iterated 100 times and average estimation errors over all the iterations are given in the tables.

Table IV.7: Comparison to fingerprinting-based methods with $\sigma_\xi = 1dB$ and $N_P = 100$ while varying N_A

N_A	4	9	16	25
NN	6.3822	5.2107	5.1156	4.9586
KNN	4.6715	3.7910	3.3132	3.1984
FKNN	4.6983	3.7640	3.29202	3.1784
FC	11.3826	5.8897	4.1130	3.1721
FCM	6.3459	5.8511	5.1857	5.1836
NNet	5.1654	5.8832	8.0229	8.8757
SVM	4.8710	3.7433	2.9537	2.6425
RR	4.0592	3.1192	2.5110	2.4351
AL	24.1827	9.0108	6.6009	6.2429
DWKNN	3.7170	2.8097	2.3282	2.3217
DWKNN-K	1.2876	1.1154	0.9915	0.9157

Table IV.8: Comparison to fingerprinting-based methods with $\sigma_\xi = 1dB$ and $N_A = 16$ while varying N_P

N_P	16	36	64	100	144	196
NN	12.7110	7.8066	6.2136	5.1100	4.1522	3.5443
KNN	9.2149	5.2929	4.5304	3.2974	2.9863	2.4691
FKNN	9.2191	5.2880	4.5127	3.2829	2.9722	2.4892
FC	12.8178	5.6345	4.3415	4.1824	3.9780	3.9253
FCM	13.7026	7.0653	6.0197	5.1857	3.8580	3.6333
NNet	23.9241	14.3839	11.1717	8.0981	5.2290	3.4888
SVM	10.2920	6.3294	5.4030	3.4090	2.9657	2.1145
RR	7.9169	4.8689	4.1563	2.7272	2.3352	1.9956
AL	14.5932	9.8631	7.7388	6.6051	6.8731	5.6665
DWKNN	10.9769	4.6650	3.2968	2.2991	2.1766	1.9344
DWKNN-K	1.7089	1.4830	1.1827	1.0321	0.9411	0.8735

IV.7 Conclusion

Clusterized localization algorithm owns better adaptability and stronger robustness in applications domain. This chapter presented an original clusterized localization technique for mobile wireless sensor networks. The proposed method is also an anchor-based method, combining both radio-fingerprints and accelerometer information but executing computations in a clusterized manner. The resolution of the problem is performed using both the interval analysis and the Kalman filter. In the interval-based approach, the proposed method computes position boxes including all the possible solutions of the problem. In the Kalman-based approach, the method predicts the positions using accelerations, then corrects them using the radio-fingerprints.

Simulations are operated over all parameters and are equally tested with different mo-

IV. CLUSTERIZED LOCALIZATION

bility models. Results show that the proposed method in both approaches outperforms other methods using only accelerometer information or fingerprinting measures. It also illustrates the efficiency of the proposed method, compared to several recent localization methods, including range-based methods, connectivity-based methods, and other fingerprints-based methods. Although the proposed method works well in the simulated environment, future works look for a chance to handle the localization problem using real data. A robust method would then be proposed to overcome real environments disturbance.

In the next chapter, a more practical case will be illustrated and tackled in a special application scenario. Based on the same idea of using both fingerprints and mobility information and solving by interval technique, it achieves the categorization function for a mobile node by addressing it to a certain block over the surveillance area.

References

- [Aldhubaib et Shuley, 2010] F. Aldhubaib et N. Shuley. Radar target recognition based on modified characteristic polarization states. *IEEE Transactions on Aerospace and Electronic Systems*, 46 (4): 1921–1933, 2010. [91](#), [92](#)
- [Baggio et Langendoen, 2006] A. Baggio et K. Langendoen. Monte-carlo localization for mobile wireless sensor networks. *Second international conference on mobile ad hoc and sensor networks (MSN)*, 2006. [112](#)
- [Chan et al., 1979] Y. Chan, A. G. Hu et J. Plant. A kalman filter based tracking scheme with input estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15 (2): 237–244, 1979. [91](#)
- [Chen et Chang, 2008] H. Chen et K. Chang. K-nearest neighbor particle filters for dynamic hybrid bayesian networks. *IEEE Transactions on Aerospace and Electronic Systems*, 44 (3): 1091–1101, 2008. [91](#), [92](#)
- [Chuenurajit et al., 2013] T. Chuenurajit, S. Phimmasean et P. Cherntanomwong. Robustness of 3d indoor localization based on fingerprint technique in wireless sensor networks. In *10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6, May 2013. [113](#)
- [Farjow et al., 2011] W. Farjow, A. Chehri, M. Hussein et X. Fernando. Support vector machines for indoor sensor localization. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 779–783, March 2011. [114](#)
- [Farmani et al., 2012] M. Farmani, H. Moradi et M. Asadpour. A hybrid localization approach in wireless sensor networks using a mobile beacon and inter-node communication. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 269–274, May 2012. [111](#)
- [Ferrari et Pagliari, 2006] G. Ferrari et R. Pagliari. Decentralized binary detection with noisy communication links. *IEEE Transactions on Aerospace and Electronic Systems*, 42 (4): 1554–1563, 2006. [90](#)
- [Gogolak et al., 2011] L. Gogolak, S. Pletl et D. Kukulj. Indoor fingerprint localization in wsn environment based on neural network. In *Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on*, pages 293–296. IEEE, 2011. [114](#)
- [Honghui et Moore, 2002] Q. Honghui et J. B. Moore. Direct kalman filtering approach for gps/ins integration. *IEEE Transactions on Aerospace and Electronic Systems*, 38 (2): 687–693, 2002. [91](#)
- [Hsu et Yu, 2009] C.-H. Hsu et C.-H. Yu. An accelerometer based approach for indoor localization. In *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, UIC-ATC'09*, pages 223–227. IEEE, 2009. [94](#)
- [Jaulin et al., 2001] L. Jaulin, M. Kieffer, O. Didrit et E. Walter. *Applied interval analysis*. Springer, 2001. [91](#)
- [Kalman, 1960] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82 (1): 35–45, 1960. [91](#), [98](#)

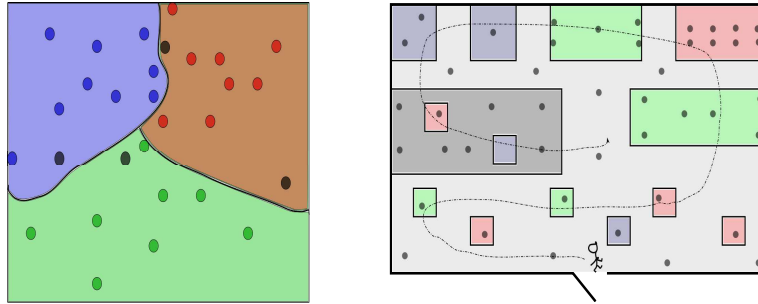
REFERENCES

- [Koyuncu et Yang, 2013] H. Koyuncu et S. H. Yang. Indoor positioning with virtual fingerprint mapping by using linear and exponential taper functions. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 1052–1057. IEEE, 2013. 114
- [Lee et Huang, 2002] S.-C. Lee et Y.-C. Huang. Innovative estimation method with measurement likelihood for all-accelerometer type inertial navigation system. *IEEE Transactions on Aerospace and Electronic Systems*, 38 (1): 339–346, 2002. 91, 94
- [Liu et al., 2012] D. Liu, D. Tang, Z. Xu, Y. Hu et C. Huang. The research of the fuzzy cluster algorithm for indoor location based on rssi. In *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pages 5–7. IEEE, 2012. 113
- [Mahfouz et al., 2013] S. Mahfouz, F. Mourad-Chehade, P. Honeine, H. Snoussi et J. Farah. Kernel-based localization using fingerprinting in wireless sensor networks. In *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 744–748, June 2013. 114
- [Medeisis et Kajackas, 2000] A. Medeisis et A. Kajackas. On the use of the universal okumura-hata propagation prediction model in rural areas. In *Vehicular Technology Conference Proceedings*, volume 3, pages 1815–1818, VTC Tokyo, May 2000. 103
- [Moore, 1966] R. E. Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966. 91
- [Mourad et al., 2009] F. Mourad, H. Snoussi, F. Abdallah et C. Richard. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on Signal Processing*, 57 (8): 3226–3239, 2009. 91
- [Navarro et al., 2010] E. Navarro, B. Peuker et M. Quan. Wi-fi localization using rssi fingerprinting. In *California Polytechnic State University, USA*, pages 1–6, 2010. 113
- [Priwgharm et Chemtanomwong, 2011] R. Priwgharm et P. Chemtanomwong. A comparative study on indoor localization based on rssi measurement in wireless sensor network. In *Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6, May 2011. 111
- [Robles et al., 2010] J. J. Robles, M. Deicke et R. Lehnert. 3d fingerprint-based localization for wireless sensor networks. In *International Workshop on Positioning, Navigation and Communication WPNC, Dresden, Germany (March 2010)*, pages 77– 85, 2010. 92
- [Roseveare et Natarajan, 2012] N. Roseveare et B. Natarajan. Distributed tracking with energy management in wireless sensor networks. *IEEE Transactions on Aerospace and Electronic Systems*, 48 (4): 3494–3511, 2012. 90
- [Rozyyev et al., 2012] A. Rozyyev, H. Hasbullah et F. Subhan. Combined k-nearest neighbors and fuzzy logic indoor localization technique for wireless sensor network. *Research Journal of Information Technology*, 4 (4): 155–165, 2012. 113
- [Schuler et al., 1967] A. R. Schuler, A. Grammatikos et K. A. FEGLEY. Measuring rotational motion with linear accelerometers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-3 (3): 465–472, 1967. 91

REFERENCES

- [Suroso *et al.*, 2011] D. J. Suroso, P. Cherntanomwong, P. Sooraksa et J.-i. Takada. Fingerprint-based technique for indoor localization in wireless sensor networks using fuzzy c-means clustering algorithm. In *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*, pages 1–5. IEEE, 2011. [113](#)
- [Tharmarasa *et al.*, 2011] R. Tharmarasa, T. Kirubarajan, A. Sinha et T. Lang. Decentralized sensor selection for large-scale multisensor-multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 47 (2): 1307–1324, 2011. [90](#)
- [Zhao et Guibas, 2004] F. Zhao et L. J. Guibas. *Wireless sensor networks: an information processing approach*. Morgan Kaufmann, 2004. [90](#)

REFERENCES



Contents

- V.1 Introduction
 - V.2 Belief functions theory
 - V.3 Zoning methodology
 - V.4 Simulation results
 - V.5 Conclusion
 - References
-

A mutation of the problem of self-localization in wireless sensor networks consists of addressing zone affiliation, especially applied in an indoor scenario where the whole surveillance area is actually composed of separated units. People sometimes pretty prefer to know a symbolic zone that denotes where the object resides instead of an exact position point, which is also called room-level localization. This chapter considers thus the zone localization problem in a mobile wireless sensor networks. Similarly to the preceding algorithms, the proposed method refers the attributed zone by using both signal strength and mobility information. The former offers a set of possible zones by weighted k nearest neighbors approach, and the latter gives another set of possible zones with aid of interval technique, both of them carrying with the confidence factors with respect to the selected zones. By intersecting the two sets of suggested zones and considering then corresponding weights, the proposed method is supposed to be capable to ensure a correct zone localization with an optional choice for each mobile node at each time-step in case that the first choice fails. Moreover, the room-level localization could be considered as a problem of multi-classification, thus the SVM-based method is equally introduced in order to make a comparison in the simulations part.

V.1 Introduction

Rapid developments of micro-electronics and wireless communication not only promote tremendously the technical revolution of wireless sensor networks, but also, in another hand, they push forward meanwhile the widespread applications of WSNs, even going into industrialization of wireless sensor networks. Meeting the requirements from all sides of living, many practical applications are emerging in recent years, such as habitat monitoring, smart building failure detections&reporting, and target tracking [Stojmenovic 2005]. All of these applications are in need beforehand of a meaningful geographical awareness of sensors themselves. Some pursue geographical information in terms of accurate orientation with respect to the global coordinate system such as those discussed in preceding chapters, while some aim to address a regional affiliation in terms of zone attribute with respect to the dimensionality. Differing with the former coordinate-level localization, the latter provides however objects room-level localization recognizing the space index where the object actually resides.

The room-level localization is especially recommended in the location-based services (LBS) applications [Schiller et Voisard 2004; Krisp 2013]. Such applications integrate the geographical location of monitored objects to provide real-time and necessary services for mobile users equipped with the handy terminals, such as for patients tracking in hospitals or person monitoring in buildings having room units as shown in Figure V.1. By means of interaction between the handy terminals and local monitoring devices, users could be localized either in terms of coordinates or just area. While in such indoors sectoral scenarios, the task of identification of the room area is much more meaningful and efficient than a precise coordinates location estimation for mobile users. There exist many researches over the room-level, or room-based localization which proceeds room identification ahead, such as entrance trigger based methods [Noury et al. 2002; Lo et al. 2012], RSS and pathloss compensation based methods, or path-restricted methods [D’hoel et al. 2008; Chiu et al. 2009] as well as probability based and SVM classifier methods [Castro et al. 2001; Hotta et al. 2012].

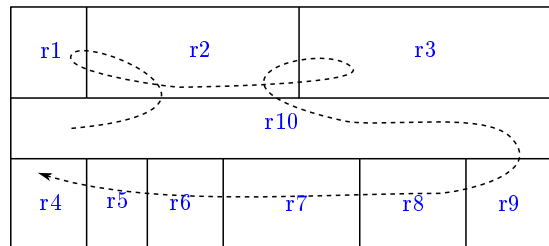


Figure V.1: Room-level localization

Managing from this localization design, a fingerprints-based area recognition technique, exploiting mobility information, is explored in this chapter. It proceeds the network configuration over the whole surveillance area. The first area estimates are made by using fingerprints information, considering confidence factor with respect to each possible zone. Meanwhile, obtaining the instantaneous mobility measurements, the second

estimates over areas are inferred by applying interval analysis technique over measured acceleration values, consequently extending to the possibly arrived neighboring zones. In this way, both of them hold respectively the set of the possible zones where object likely resides. To make the final decision, all these possibilities are put into refinement phase such as a combination phase by intersecting them and computing the appended confidence factors. Resulting zones are ranked in order of their respective confidence factors, where normally the first two meet the requirements in most cases. The solution proposed in this chapter lies on the belief function theory [Shafer *et al.* 1976], according to which the collected information are used to define belief functions tackled then with a combination step to pursue a possible zone where the considered node falls. The belief function theory is briefly introduced in the next part. The solution of zoning is addressed in what follows.

V.2 Belief functions theory

The confidence factor, assigned to each possible solution and mentioned in the preceding section, is decided effectively upon the *Belief Functions Theory* (BFT), called also the Dempster-Shafer theory. The BFT is introduced below, considered as a mathematical tool that permits the fusion of data issued from different sources in the way to make the right decisions concerning the nodes locations.

V.2.1 Description of the theory

The BFT is introduced by G. Shafer in 1976 [Shafer *et al.* 1976] based on the work developed by A. Dempster in 1967 [Dempster 1967]. In 1994, P. Smets accomplished this work by proposing a formal framework for the BFT, named as the *Transferable Belief Model* (TBM) [Smets et Kennes 1994], which is an non-probabilistic model permitting to represent the imperfection of collected information, to make a fusion of information then to help making decisions. It is worth noting that this theory has been employed in varied domains, such as fusion of multi-sensor information, fusion of classifiers, pattern recognition, etc [Denoeux 1997; Lefevre *et al.* 2002; Ristic et Smets 2005].

Before introducing some basic notions proposed in the framework of TBM, consider any kind of systems in which one looks for determining a status x . Let $\Omega = \{\omega_1, \dots, \omega_n\}$ be the definition set of x , known as the frame of discernment where the values $\{\omega_i\}_{i=1}^n$ are considered to be discrete and known. 2^Ω denotes the set of all parties of Ω , as below,

$$2^\Omega = \{\emptyset, \{\omega_1\}, \dots, \Omega\}. \quad (\text{V.1})$$

All the measurements concerning the status x are considered as sources of information in BFT which introduces the confidence notion, represented in different ways in the framework of TBM. In the following, several representations of belief functions are briefly introduced, such as m-function, belief function and plausibility function.

V. ZONING LOCALIZATION

- The m-function, called also the belief masses function (MF), is the most used formulation of the belief functions, denoted generally by m_S^Ω and defined as follows,

$$\begin{aligned} m_S^\Omega : 2^\Omega &\rightarrow [0, 1], \\ A &\mapsto m_S^\Omega(A), \text{ satisfying,} \\ \Sigma_{A \in 2^\Omega} m_S^\Omega(A) &= 1. \end{aligned} \quad (\text{V.2})$$

The symbol S denotes the source of information from which m_S^Ω is issued. The mass $m_S^\Omega(A)$ corresponds to the part of belief attributed by the source S to the hypothesis of the status x being included in A . Generally, notation $m_S^\Omega(A)$ could be simplified as $m^\Omega(A)$ when there is no need to specify the source of information. All the subsets A of Ω such that $m^\Omega(A) > 0$ are called the focal elements. If $\Omega = \mathbb{R}$, the focal elements turns to be in the framework of intervals [Moore et Moore 1979; Jaulin et al. 2001]. In the case of $\Omega = \mathbb{R}^n$, the elements are in terms of pavings of dimension n .

- The belief function, associated with an MF m^Ω , is equally defined from 2^Ω to $[0, 1]$ as follows,

$$bel^\Omega(A) = \Sigma_{\emptyset \neq B \subseteq A} m^\Omega(B), \quad \forall A \subseteq \Omega. \quad (\text{V.3})$$

The value $bel^\Omega(A)$ represents the level of belief with the assumption of $x \in A$. In this equation, the $m^\Omega(\cdot)$ is obtained by,

$$m^\Omega(A) = \begin{cases} \Sigma_{\emptyset \neq B \subseteq A} (-1)^{|A|-|B|} bel^\Omega(B) & \forall A \in \Omega, A \neq \emptyset, \\ 1 - bel(\Omega), & \text{if } A = \emptyset. \end{cases} \quad (\text{V.4})$$

- The plausibility function is denoted by pl^Ω , similarly associated with an MF m^Ω . It is defined as,

$$pl^\Omega(A) = \Sigma_{B \cap A \neq \emptyset} m^\Omega(B) = 1 - bel(\bar{A}), \quad \forall A \subseteq \Omega. \quad (\text{V.5})$$

The value of $pl^\Omega(A)$ represents the maximal belief level associated to the assumption of $x \in A$. $m^\Omega(\cdot)$ could be obtained from the plausibility function equation above as follows,

$$m^\Omega(A) = \begin{cases} \Sigma_{\emptyset \neq B \subseteq A} (-1)^{|A|-|B|+1} pl^\Omega(\bar{B}) & \forall A \in \Omega, A \neq \emptyset, \\ 1 - pl(\Omega), & \text{if } A = \emptyset, \end{cases} \quad (\text{V.6})$$

where \bar{B} is the complement of B in Ω .

It is worth noting that the sources of information yielding the MFs could be not totally trustful. In this case, it is possible to introduce a weakness coefficient $(1 - \alpha) \in [0, 1]$ to the MFs. This coefficient represents the reliability of the source. In this way, a weakened MF denoted by $m^{\alpha, \Omega}$ is deduced as follows,

$$m^{\alpha, \Omega}(A) = \begin{cases} (1 - \alpha)m^\Omega(A) & \text{if } A \subset \Omega, A \neq \Omega, \\ \alpha + (1 - \alpha)m^\Omega(A), & \text{if } A = \Omega. \end{cases} \quad (\text{V.7})$$

V.2.2 Combination rules

Obtaining the information issued from the different sources, the contribution [Smets 1993] proposed several operators to make a combination of all the information leading to more trustful results.

Let m_1 and m_2 be the simplified notations for $m_{S_1}^\Omega$ and $m_{S_2}^\Omega$ which are two MFs issued respectively from two different but trustworthy sources S_1 and S_2 . The conjunctive rule of combination (CRC) is defined as follows,

$$m_{1\ominus 2}(A) = (m_1 \ominus m_2)(A) = \sum_{i,j|B_i \cap C_j = A} m_1(B_i)m_2(C_j), \quad \forall A \subseteq \Omega. \quad (\text{V.8})$$

The two sources of information could be conflicting leading to some empty intersections. In this case, the definition of $m_{1\ominus 2}(\cdot)$ could be followed by a normalization step, to have a resulting MF having its sum equal to one.

In comparison, the disjunctive rule of combination (DRC) is alternatively used to combine two MFs of which at least one issue is trustworthy. In this case, the combination of two MFs m_1 and m_2 is represented as follows,

$$m_{1\oplus 2}(A) = (m_1 \oplus m_2)(A) = \sum_{i,j|B_i \cup C_j = A} m_1(B_i)m_2(C_j), \quad \forall A \subseteq \Omega. \quad (\text{V.9})$$

Note that the MF obtained by CRC is more informative than m_1 and m_2 since this rule transfers the masses of focal elements to a smaller cardinality set. On contrary, the MF issued by DRC is less informative due to an effect of generalization.

V.2.3 Decision rules

There exist several rules of decision utilized in the framework of the belief function theory, such as the maximum of belief masses, the maximum of plausibility, the maximum of the pignistic probability, etc [Denoeux 1997]. Since the mass and the plausibility functions are already introduced, let us consider here the pignistic probability denoted by $BetP$, computed using the pignistic transformation [Smets et Kennes 1994]. Let m^Ω be an MF associated to Ω and A_1, \dots, A_r be the focal sets of m^Ω . The pignistic transformation lies on the uniform division of the mass of each element A_i , $i \in \{1, \dots, r\}$, shown as follows,

$$BetP(\omega_k) = \sum_{i|A_i \subseteq \Omega, \omega_k \in A_i} \frac{m^\Omega(A_i)}{|A_i|}, \quad (\text{V.10})$$

where $|A_i|$ denotes the cardinality of A_i . Note that, in the case of $\Omega = \mathbb{R}$, the focal elements turn to be in the framework of intervals $[x]_1, \dots, [x]_r$, thus the pignistic probability is represented as follows,

$$BetP(x) = \sum_{i=1}^r \mathbb{I}_{[x]_i}(x) \frac{m^\Omega([x]_i)}{|[x]_i|}, \quad (\text{V.11})$$

where $\mathbb{I}_{[x]_i}(x)$ is the indicator function denoting the affiliation of x to $[x]_i$ with $\mathbb{I}_{[x]_i}(x) = 1$ or non-affiliation with $\mathbb{I}_{[x]_i}(x) = 0$, and $|[x]_i|$ denotes the width of the interval $[x]_i$.

V.3 Zoning methodology

The room-level localization, called here zoning localization, aims at determining the correct zone where the object actually resides at any time-step t , rather than an exact position estimate as discussed before. Correspondingly, the whole sensor network is composed of several zones on behalf of independent spaces, such as offices, class rooms, workshops and corridors. An object, that is usually taken as a mobile node, moves freely throughout the zones of the surveillance area. Fingerprints and inertial information employed in the preceding chapters are used to solve equally this zoning problem with aid of interval analysis [Jaulin *et al.* 2001], combined to the belief function theory [Smets *et Kennes* 1994].

V.3.1 Configuration of the network

The proposed method is an anchor-based centralized zoning technique, based on radio-fingerprints. Consider a D -dimensional surveillance area of \mathbb{R}^D having N_Z separate zones, each assumed to be an independent unit of space. All zones are assumed to be defined in the same spatial dimension, which is a particular problem of the zoning issue. A certain zone could be denoted by a D -elements attribute (z_1, \dots, z_D) , where $z_d \in \{1, \dots, n_{Z_d}\}$ denotes the integer coordinate of the zone according to the direction d , $d \in \{1, \dots, D\}$, with n_{Z_d} denoting the maximal value of z_d and $N_Z = \prod_{d=1}^D n_{Z_d}$. Being an anchor-based method, the proposed approach considers two kinds of sensors in the network, anchors $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,D})$, $i \in \{1, \dots, N_A\}$, with known fixed positions, and mobile nodes $\mathbf{x}_j(t) = (x_{j,1}(t), \dots, x_{j,D}(t))$, $j \in \{1, \dots, N_X\}$, with unknown positions moving freely over the surveillance area. The objective of the method is to determine the attribute $\mathbf{z}_{\mathbf{x}_j}(t) = (z_{x_{j,1}}(t), \dots, z_{x_{j,D}}(t))$ of the zone where the node j falls at time t . Similarly to the previously described methods, a fingerprinting database is required to process the computations later on. To do this, N_P reference positions are uniformly generated over the whole area, denoted by $\mathbf{p}_k = (p_{k,1}, \dots, p_{k,D})$, $k \in \{1, \dots, N_P\}$, with zone attributes $\mathbf{z}_{\mathbf{p}_k} = (z_{p_{k,1}}, \dots, z_{p_{k,D}})$, $k \in 1, \dots, N_P$, denoting in which zone the reference position \mathbf{p}_k resides. An example of a two-dimensional network with nine zones is shown in Figure V.2. Here, $n_{Z_1} = n_{Z_2} = 3$. For instance, the zone 4 has the attribute (1, 2), whereas the zone 8 has the attribute (2, 3).

Once the basic network is constructed, anchors broadcast radio signals over the whole network with the same initial power. The received signal strengths decrease with the increase of the signals traveled distance, which means that given a position, there exist theoretically a unique set of RSSI values corresponding to the distances separating this position from the anchors. According to this virtue, the power database is built up by measuring the RSSI $\xi_{P_k,i}$ at all reference positions \mathbf{p}_k , $k \in \{1, \dots, N_P\}$, of the signals sent from all anchors \mathbf{a}_i , $i \in \{1, \dots, N_A\}$. Let $\boldsymbol{\xi}_{P_k}$ be the N_A vector of the measured RSSI at the reference position \mathbf{p}_k , then

$$\boldsymbol{\xi}_{P_k} = (\xi_{P_k,1}, \dots, \xi_{P_k,N_A}), \quad k \in \{1, \dots, N_P\}. \quad (\text{V.12})$$

Consequently, the fingerprints database consists of N_P couples, each one denotes the

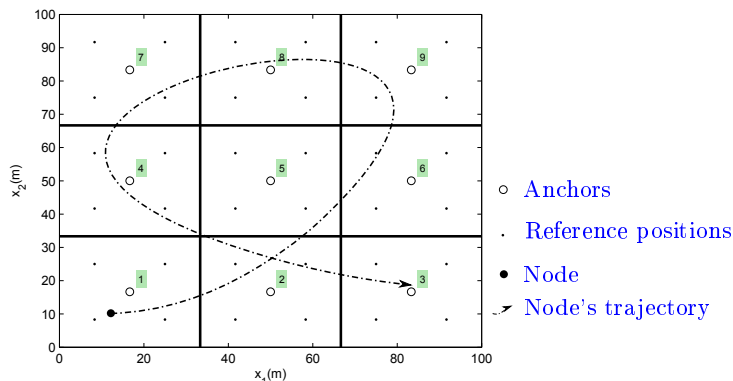


Figure V.2: An example of a centralized zone network

RSSI vector of a reference position with its zone attribute $(\boldsymbol{\xi}_{P_k}, z_{P_k})$. The database is then stored at the fusion center for further computations.

V.3.2 Fingerprints zoning

A node j measures the RSSI of the signals it receives from all sensed anchors at time-step t and constructs its RSSI vector as shown in Eq. V.13,

$$\boldsymbol{\xi}_{X_j}(t) = (\xi_{X_{j,1}}(t), \dots, \xi_{X_{j,N_A}}(t)), \quad j \in \{1, \dots, N_X\}, \quad (\text{V.13})$$

where $\xi_{X_{j,i}}(t)$ stands for the strength at reception of the signal emitted by the anchor i and received by the node j at time t . Similarly with the KNN technique mentioned in chapter III, to locate the mobile node j in terms of zones, the proposed approach proceeds to the computation of the Euclidean distances $\|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_k}\|$, $k \in 1, \dots, N_P$, then it selects the K reference positions yielding the K smallest strengths distances. Let $I_j(t)$ be the set of indices of these reference positions at time t . Given upon these selected indices, zone location estimates are then computed using the respective zone terms of the selected reference positions as follows,

$$\hat{z}_{x_{j,d}}(t) = \sum_{k \in I_j(t)} \omega_{j,k}(t) z_{P_k,d}, \quad d \in \{1, \dots, D\}, \quad (\text{V.14})$$

where $\omega_{j,k}(t)$ is a normalized weight factor inversely proportional to the RSSI Euclidean distances computed in Eq. V.15,

$$\omega_{j,k}(t) = \frac{\|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_k}\|^{-2}}{\sum_{u \in I_j(t)} \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_u}\|^{-2}}. \quad (\text{V.15})$$

Note that this expression of the weight factors is adopted here since analysis in Chapter IV showed that this expression yields the best estimation results. The quantities $\hat{z}_{x_{j,d}}(t)$, $d \in \{1, \dots, D\}$, are the estimates of the zones coordinates in the d directions

V. ZONING LOCALIZATION

within the D -dimensional surveillance area. However, in most of the cases, $\hat{z}_{x_j,d}(t)$ are not integer. Let $\underline{z}_{x_j,d}(t)$ and $\bar{z}_{x_j,d}(t)$ be the integers respectively preceding and following $\hat{z}_{x_j,d}(t)$, and let $\tilde{z}_{x_j,d}(t)$ denotes the integer estimate of the exact zone coordinate $z_{x_j,d}(t)$ of the node j at time t . Then $\tilde{z}_{x_j,d}(t)$ would have two values $\tilde{z}_{x_j,d}(t) \in \{\underline{z}_{x_j,d}(t), \bar{z}_{x_j,d}(t)\}$, leading to 2^D vector estimates of the zone attribute $\mathbf{z}_{x_j}(t)$. Based on the belief functions theory, the proposed approach consists then on assuming this estimation as a source of information, yielding a mass function $m_{f_j,t}(\cdot)$, having the 2^D obtained attributes as focal elements. Let $\tilde{\mathbf{z}}_{x_j}(t) = (\tilde{z}_{x_j,1}(t), \dots, \tilde{z}_{x_j,D}(t))$ be one of these estimates, then $m_{f_j,t}(\cdot)$ is defined as follows,

$$m_{f_j,t}(\tilde{\mathbf{z}}_{x_j}(t)) = \prod_{d=1}^{d=D} (1 - |\tilde{z}_{x_j,d}(t) - \hat{z}_{x_j,d}(t)|). \quad (\text{V.16})$$

The quantity $m_{f_j,t}(\tilde{\mathbf{z}}_{x_j}(t))$ stands for the confidence given to the zone attribute estimate $\tilde{\mathbf{z}}_{x_j}(t)$. Indeed, the closer an element $\hat{z}_{x_j,d}(t)$ of $\tilde{\mathbf{z}}_{x_j}(t)$ is to the integer $\tilde{z}_{x_j,d}(t)$, the smaller $|\tilde{z}_{x_j,d}(t) - \hat{z}_{x_j,d}(t)|$ is and hence the greater the confidence given to $\tilde{z}_{x_j,d}(t)$ should be. If only one zone attribute should be selected, it would be the estimate having the highest mass. If other choices are needed, one selects the attributes in the decreasing order of their masses. An illustration of this computation is given in Example 6. The estimation using fingerprints would be combined to the accelerometer information to refine more the selection.

Example 6 Consider the 2-dimensional surveillance area of Figure V.2 with only one mobile node. Assume that at $t = 50$, KNN computations lead for instance to $\hat{z}_{x_1}(t) = 2.3$ and $\hat{z}_{x_2}(t) = 2.9$, whereas the node resides for real in the 8-th zone of attribute (2,3) as shown in Figure V.3. The proposed approach leads to four attribute estimates (2,2), (3,2), (2,3) and (3,3) corresponding respectively to the zones 5, 6, 8 and 9 with the masses 0.07, 0.03, 0.63 and 0.27. The maximal mass decision rule leads to the selection of the attribute (2,3) with the highest confidence of 0.63, with a second choice of the attribute (3,3) with much smaller confidence of 0.27.

V.3.3 Accelerometer zoning

Mobile nodes equipped with accelerometer devices offer supplementary information for the zoning process. Let $\boldsymbol{\gamma}_j(t) = (\gamma_{j,1}(t), \dots, \gamma_{j,D}(t))$ be the acceleration vector of a node j , $j \in \{1, \dots, N_X\}$, at time t . This information would lead to an additional mass function for zone estimation, using the interval analysis.

As mentioned before, a mobile node j moves freely over all the surveillance area. For a node j at time instant t , given its instantaneous acceleration vector $\boldsymbol{\gamma}_j(t)$, it is possible to estimate the current position by the mobility model, such as the third order equations III.9 and III.10, as follows,

$$\hat{\mathbf{x}}_j(t) = \hat{\mathbf{x}}_j(t-1) + \hat{\mathbf{v}}_j(t-1)\Delta t + \frac{1}{2}\boldsymbol{\gamma}_j(t-1)\Delta t^2 + \frac{1}{6}\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}\Delta t^3, \quad (\text{V.17})$$

where $\hat{\boldsymbol{v}}_j(t)$ is the velocity vector of the node j at time-step t , computed by

$$\hat{\boldsymbol{v}}_j(t) = \hat{\boldsymbol{v}}_j(t-1) + \boldsymbol{\gamma}_j(t-1)\Delta t + \frac{1}{2} \frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t} \Delta t^2, \quad (\text{V.18})$$

where Δt is the time interval between two following time-steps. In real conditions, the measured accelerations could be noisy due to several factors. To avoid erroneous computations, an uncertainty quantity δ_γ is introduced to the measured accelerations, δ_γ being set by performing a calibration step of the accelerometer device. The accelerations are then reformulated in interval scheme as follows,

$$[\boldsymbol{\gamma}_j](t) = [\gamma_{j,1}](t) \times \cdots \times [\gamma_{j,D}](t) = [\gamma_{j,1} - \delta_\gamma, \gamma_{j,1} + \delta_\gamma] \times \cdots \times [\gamma_{j,D} - \delta_\gamma, \gamma_{j,D} + \delta_\gamma]. \quad (\text{V.19})$$

By substituting Eq. V.19 into Eq. V.17 and V.18, one obtains a D -dimensional box $[\hat{\boldsymbol{x}}_j](t)$ surrounding the position estimate $\hat{\boldsymbol{x}}_j(t)$ as follows,

$$[\hat{\boldsymbol{x}}_j](t) = [\hat{\boldsymbol{x}}_j](t-1) + [\hat{\boldsymbol{v}}_j](t-1)\Delta t + \frac{1}{2} [\boldsymbol{\gamma}_j](t-1)\Delta t^2 + \frac{1}{6} \frac{[\boldsymbol{\gamma}_j](t) - [\boldsymbol{\gamma}_j](t-1)}{\Delta t} \Delta t^3, \quad (\text{V.20})$$

where $\hat{\boldsymbol{v}}_j(t)$ is accordingly extended as,

$$[\hat{\boldsymbol{v}}_j](t) = [\hat{\boldsymbol{v}}_j](t-1) + [\boldsymbol{\gamma}_j](t-1)\Delta t + \frac{1}{2} \frac{[\boldsymbol{\gamma}_j](t) - [\boldsymbol{\gamma}_j](t-1)}{\Delta t} \Delta t^2. \quad (\text{V.21})$$

It would not be surprising that the box $[\hat{\boldsymbol{x}}_j](t)$ grows more and more with time accumulation. Consequently, the accelerometer-based zoning accuracy drops greatly through time. Considering this drawback, an alternative solution using additive uncertainty over the acceleration-based position scalar estimate is also applicable. Refer to Remark 3 of Chapter III for more details. This way, the resulting mobility based box grows more slowly and keeps its efficiency over longer observing periods.

Once the box $[\hat{\boldsymbol{x}}_j](t)$ is available and going back to the zoning problem, the possible landing zones of the node j at time t are obtained by intersecting $[\hat{\boldsymbol{x}}_j](t)$ with all the zones of the surveillance area. The intersection might lead to 1, 2, ..., or N_Z zones. Accordingly, a second mass function $m_{\gamma_j,t}(\cdot)$ is defined. Its focal elements would be the attributes of the zones having intersection with $[\hat{\boldsymbol{x}}_j](t)$, with their associated masses defined as the surface of the intersection area of $[\hat{\boldsymbol{x}}_j](t)$ with each zone, followed by a normalization phase. Consider the example of Figure V.3 with a 2-dimensional area and one mobile node located at time $t = 50$ in the 8-th zone with attribute (2, 3). With the position box as shown in gray in the plot, the proposed approach leads to two possible attribute estimates (2, 3) and (3, 3), corresponding to zones 8 and 9 respectively. The intersection of the position box with zone 8 has a surface of 4/5 of the total surface of the box and the intersection with zone 9 has a ratio of 1/5, leading to masses of zones 8 and 9 of 0.8 and 0.2 respectively. The maximal mass decision rule yields then zone 8 of attribute (2, 3).

V. ZONING LOCALIZATION

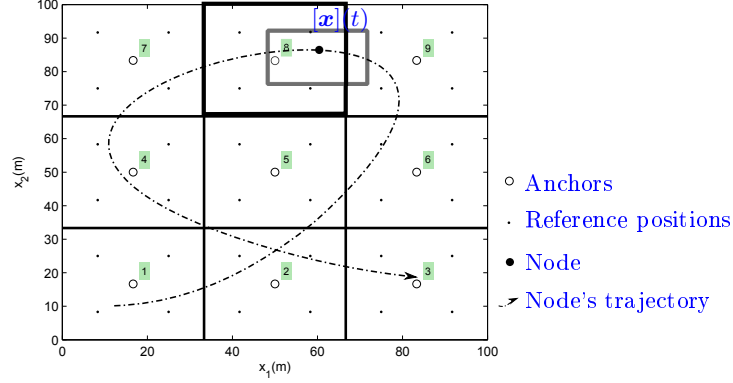


Figure V.3: An example of zoning using accelerometer information

V.3.4 Final zoning

The proposed method considers two sources of information, fingerprints and accelerations yielding two zoning sets. Gathering both information consists then of combining their defined masses functions using the conjunctive rule of combination. Firstly, the zones attributes estimated with fingerprints are intersected with the ones obtained using accelerations. Then, the masses of the resulting attributes are obtained by multiplying their corresponding masses as follows,

$$m_{\cap j,t}(\tilde{\mathbf{x}}_j(t)) = m_{f j,t}(\tilde{\mathbf{x}}_j(t)) * m_{\gamma j,t}(\tilde{\mathbf{x}}_j(t)), \quad (\text{V.22})$$

where $\tilde{\mathbf{x}}_j(t) = (\tilde{x}_{j,1}(t), \dots, \tilde{x}_{j,D}(t))$ is a zone attribute obtained in both fingerprints and accelerometer estimations. If $m_{f j,t}(\cdot)$ and $m_{\gamma j,t}(\cdot)$ are conflicting, there might be zones estimated by one and not the other. Here the computation of the global mass function $m_{\cap j,t}(\cdot)$ should be followed by a normalization phase. The global mass function could have a single zone or several zones with corresponding confidences. In this case, if only one zone is needed, the one having the greatest mass is selected. Otherwise, the zones are sorted in descend order of their masses to make a decision according to the probability confidence factor. The greater the mass is, the greater the probability of the node j to fall into the corresponding zone is. One could also set thresholds ϵ_{m_f} , ϵ_{m_γ} and ϵ_{m_\cap} limiting the selection among the confidence factors of the mass functions $m_f(\cdot)$, $m_\gamma(\cdot)$ and $m_\cap(\cdot)$ respectively in order to optimize the final estimation. That means that the proposed method considers only the zones attributes having masses higher than the thresholds, reducing this way the set of selected zones.

Example 7 Consider the case of Example 6 and Figure V.3. The fingerprints mass function has the focal elements (2,2), (3,2), (2,3) and (3,3) with their corresponding masses 0.07, 0.03, 0.63 and 0.27; whereas the accelerometer mass function has the focal elements (2,3) and (3,3) with their corresponding masses 0.8 and 0.2. The conjunctive rule of combination leads to the zones attributes (2,3) and (3,3) with the respective

masses 0.504 and 0.054. The normalization leads to the final mass values of 0.903 and 0.097 respectively. If a final mass threshold of $\epsilon_{m_\cap} = 0.1$ for instance is considered, only (2,3) is kept in the estimation, which is the correct zone attribute.

V.4 Simulation results

The performance of the proposed zoning method is illustrated in this part. To this end, consider a $100m \times 100m$ surveillance area composed of N_Z zones, where anchors and reference positions are uniformly deployed, especially the reference positions are equally uniformly deployed in each zone. Without loss of generality, consider the localization of only one mobile node, hence the j index will be withdrawn when needed in the following. This is possible since each node localizes itself using the anchors information and accelerations, independently from other nodes. It is worth noting that all simulations are done on Matlab R2010b platform.

V.4.1 Generation of simulated data

In order to produce the nodes trajectory, two acceleration signals are generated beforehand along the x_1 -axis and the x_2 -axis, varying over 100s with $\Delta t = 1s$ using the sine function. The trajectory is then obtained by integrating twice the acceleration analytical functions. The initial velocities are set to zero since the node is assumed to be fixed at a known position at the very beginning of the localization process. Acceleration signals are depicted in Figure V.4, whereas the trajectory of the node is shown in Figure V.5 with also 16 anchors and 81 reference positions uniformly deployed over the surveillance area.

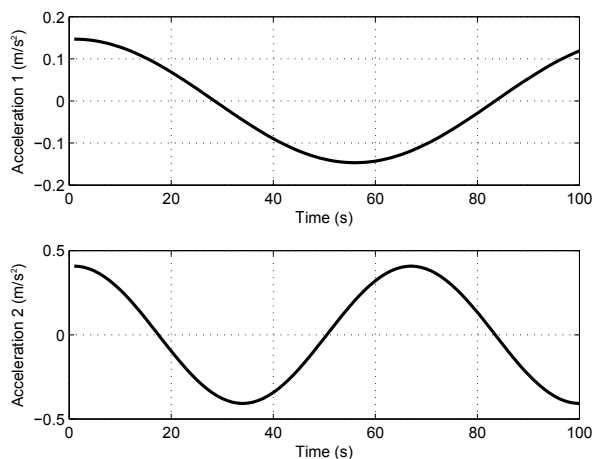


Figure V.4: Acceleration signals

The RSSI values are then generated by using the distances $\|\mathbf{a}_i, \mathbf{p}_k\|$, separating the anchors \mathbf{a}_i from the reference points \mathbf{p}_k , or $\|\mathbf{a}_i, \mathbf{x}(t)\|$, separating the anchors from the

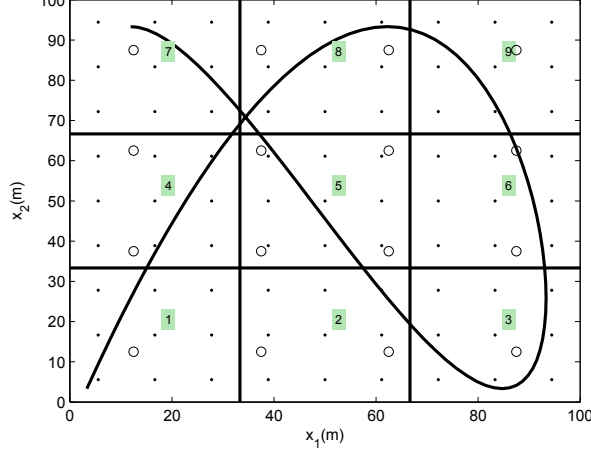


Figure V.5: Trajectory of the node

node, and the pathloss model as follows,

$$\begin{aligned} \xi_{P_{k,i}} &= \xi_0 - n_0 * \log(\|\mathbf{a}_i, \mathbf{p}_k\|/d_0) + \varepsilon_{\xi_{k,i}}, \\ \xi_{X_i}(t) &= \xi_0 - n_0 * \log(\|\mathbf{a}_i, \mathbf{x}(t)\|/d_0) + \varepsilon_{\xi_i}(t), \quad k \in 1, \dots, N_P, \quad i \in 1, \dots, N_A, \end{aligned} \quad (\text{V.23})$$

where all RSSI values are in dBm unit. ξ_0 is the transmission power set to $100dBm$, measured at the distance d_0 set to $0.1m$. n_0 is the pathloss parameter, set to 4. $\varepsilon_{\xi_{k,i}}$ and $\varepsilon_{\xi_i}(t)$ are both zero-mean gaussian noises, with standard deviations denoted by σ_ξ . Similarly, a zero-mean gaussian noise $\varepsilon_{\gamma_d}(t)$ with a standard deviation σ_γ is added to acceleration values, then measured accelerations signals are represented as follows,

$$\gamma_d(t) = \tilde{\gamma}_d(t) + \varepsilon_{\gamma_d}(t), \quad d = 1, 2, \quad (\text{V.24})$$

where $\tilde{\gamma}_d(t)$ denotes the noiseless acceleration signal over the d direction. The resulting noisy accelerations are used in the algorithm during the computation. Accordingly, the incertitude quantity δ_γ is set to $3\sigma_\gamma$ in simulated data. In order to justify this choice of value, refer to Remark 1 of Chapter III. It is worth noting that in real environments where σ_γ is actually unknown, δ_γ is then set beforehand by the calibration of the accelerometer device, ideally as the maximum of device measurement error.

V.4.2 Illustration of the proposed method

An illustration over the proposed zoning method is given in a general way at the very first, thus we set the relevant parameters $N_A = 16$, $N_P = 81$, $N_Z = 9$ and $K = 3$. With $N_Z = 9$, $n_{Z_1} = n_{Z_2} = 3$. The additional noise standard deviation σ_ξ is set to be the value within the range of 1% ~ 10% of the standard deviation of the RSSI database ξ_P and ξ_X , and σ_γ is equally set to be the value within the range of 1% ~ 10% of the standard deviation of accelerations γ . In this context, consider the performance of

zone localization for the nodes trajectory within 100s over the area of $100m \times 100m$ generated above. Table V.1 shows the zoning correct hitting rate over the 100 positions of the node with different values of σ_ξ and σ_γ . The correct hitting rates are shown for the case where only the zone having the greatest mass is selected (1 zone) and also the case where the two zones having the two greatest masses are chosen (2 zones), compared among the single fingerprints method, accelerometer method and the combined method. It is observed that the combined method increases the hitting rate for the first choice and further locates nodes with success by the second choice if the first estimate is not the right zone location.

Table V.1: Zones hitting rate over 100 seconds trajectory

$\sigma_\gamma, \sigma_\xi$	Fingerpintings		accelerometer		combined	
	1 zone	2 zones	1 zone	2 zones	1 zone	2 zones
1%, 10%	90%	100%	80%	100%	95%	100%
5%, 5%	93%	100%	48%	65%	94%	100%
5%, 10%	90%	100%	45%	63%	91%	100%
10%, 5%	92%	100%	29%	40%	92%	100%
10%, 10%	91%	100%	21%	40%	93%	100%

V.4.3 Impacts of parameters

In this part, the performance over zone recognition is further illustrated by the criteria of zoning hitting rate, while separately using varying values of the network parameters, including the zones number N_Z , reference positions number N_P , anchors number N_A and noise standard deviations σ_γ and σ_ξ . All the simulations are operated using the third mobility model as Eq. V.17 and the first weight expression with $\alpha = 2$ as shown in Eq. V.15. Besides, another trial with respect to the incertitude δ_γ over measured accelerations and δ_ν over computed velocities are equally proposed. Performance results are the average zone hitting rate over 50 attempts.

V.4.3.1 Impact of the zones number N_Z

At the very beginning, we set K to 3, N_P to 100, N_A to 16, and σ_γ and σ_ξ to 1% of the standard deviations of the noiseless accelerations and RSSI signals respectively. We then vary the total number of zones N_Z from 2^2 to 6^2 . Here the numbers of zones per directions are assumed to be equal with $n_{Z_1} = n_{Z_2} = \sqrt{N_Z}$. Comparison is made with varying N_Z among the methods of fingerprints, accelerometer and combination. Table V.2 shows the correct zones hitting rate in terms of first choice (1 zone), as well as the first and the second choices (2 zones), represented over the 100s trajectory as shown in Figure V.5. The table shows that the proposed combined method as well as its only fingerprinting based one perform well with very high accuracy, even while only selecting the first zone. It is worth noting that simulations are run 50 times and the results are averaged over all the times.

V. ZONING LOCALIZATION

Table V.2: Zones hitting rate while varying N_Z

N_Z	Fingerprints		Accelerometer		Combined	
	1 zone	2 zones	1 zone	2 zones	1 zone	2 zones
$N_Z = 4$	100%	-	97.4%	99.92%	100%	-
$N_Z = 9$	94.94%	100%	82.64%	99.66%	94.94%	100%
$N_Z = 16$	99.74%	100%	76.46%	89.28%	99.80%	100%
$N_Z = 25$	95.84%	100%	74.54%	83.06%	98.12%	100%
$N_Z = 36$	94.88%	100%	73.28%	82.96%	96.08%	100%

Moreover, the accuracy generally decreases with the increase of the number of zones. It is worthy explicating additionally here the reason why the results for $N_Z = 9$ turn to be the worst among all the selections. According to the K nearest neighbors selection method, the reference positions yielding the smallest RSSI distances with that of the mobile node are chosen. However, by applying the RSSI computation model up to Eq. V.23, the RSSI values decrease exponentially with the increase of the distance. This means that if two points are too close but having very small distances to an anchor, the difference between their RSSI would be large, which does not represent their closeness. Here in the case of 9 zones, and since the reference positions are uniformly distributed in each zone, some reference positions are very close to anchors, as illustrated in Figure V.6. The plot shows in black circle the considered anchor, in red a node's position whose zone estimate is wrong, in black points the reference positions and in green circles the selected reference positions using the K nearest neighbors algorithm for the considered node's position. The truly closest reference positions to the node are the two top ones and the right down one. However, since the right top one is too close to the anchor, and by taking RSSI distances, the algorithm falls on the reference positions encircled with green. Consequently, the mobile node in red drops into the neighboring zone of the true location, which decreases the correct zones hitting rate.

This phenomenon occurs thus when anchors reside too close to the reference points. It is unpreventable if the random distribution of the network is adopted. Otherwise, it could be considered as the case when additive noises are counted. Anyway, it offers rather a chance of testing the reliability and adaptability of the proposed method with the second choice of zones.

V.4.3.2 Impact of the number of reference positions N_P

This section aims at analyzing the impact of the number of reference positions on the proposed method accuracy. We thus set K to 3, the number of zones N_Z to 9 and the number of anchors N_A to 9, which leads to one anchor positioned at the center of each zone separately. σ_γ and σ_ξ are set equal to 1% and 5% of the standard deviations of the noiseless acceleration signals and RSSI signals respectively. The positions of anchors are required to be moved slightly in order to prevent the case when their positions are overlapped with reference positions. Table V.3 shows the correct zones hitting rate with the only fingerprinting method, the only accelerometer method and the combined method, with one and two selected zones. According to the plot, along with the increase

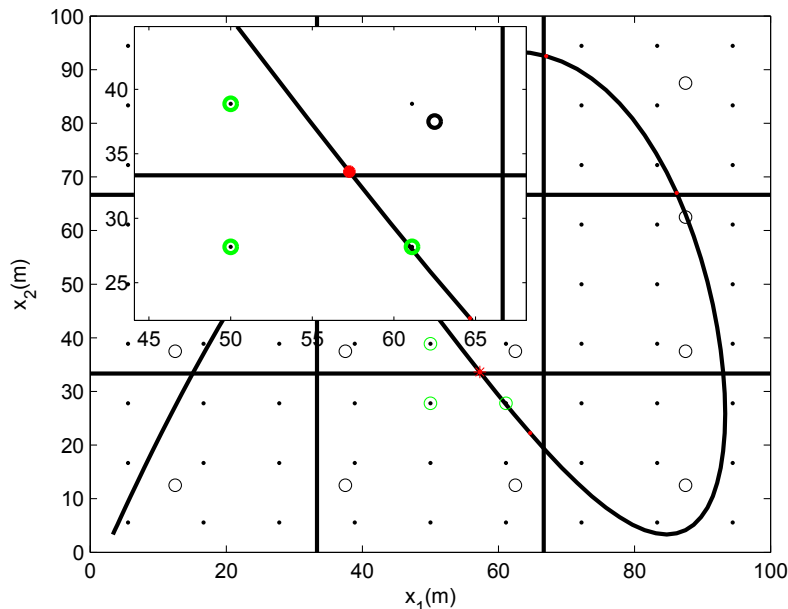


Figure V.6: An illustration of altered reference position

of the number of reference positions, the zone hitting rate of the proposed methods increases with slightly varying values since $N_P = 81$. Moreover, the second choice supplied is always 100% scoring the hit. It is worth noting that simulations are run 50 times and the results are averaged over all the times.

Table V.3: Zones hitting rate while varying N_P

N_P	Fingerprints		Accelerometer		Combined	
	1 zone	2 zones	1 zone	2 zones	1 zone	2 zones
$N_P = 9$	93.38%	99.98%	81.42%	99.10%	95.72%	100%
$N_P = 36$	97.20%	99.98%	80.86%	98.64%	97.44%	100%
$N_P = 81$	98.22%	100%	81.76%	98.80%	98.58%	100%
$N_P = 144$	97.78%	100%	80.42%	97.72%	97.98%	100%
$N_P = 225$	97.90%	99.98%	81.66%	99.42%	98.02%	100%
$N_P = 284$	97.64%	99.96%	80.94%	98.86%	97.88%	100%

V.4.3.3 Impact of the number of anchors N_A

Performance with varying anchors number carries on then. We thus set N_Z to 9 and N_P to 81. σ_γ and σ_ξ are equal to 1% and 5% of that of noiseless signals. The positions of anchors are similarly required to be moved slightly (+1) in order to prevent the case when their positions are overlapped with reference positions. Table V.4 shows the algorithm's performance while varying the number of anchors N_A . Note that when $N_A = 9$, anchors are situated at the center of each zone where no affection at the

V. ZONING LOCALIZATION

frontiers. Similarly when anchors are away from the zone frontiers, the zoning rate turns to be better. The table shows that the only fingerprinting-based method performs very well, with much better accuracy than the only accelerations-based one; however, the combination of both information still increases, even slightly, the performances.

Table V.4: Zones hitting rate while varying N_A

N_A	Fingerprints		Accelerometer		Combined	
	1 zone	2 zones	1 zone	2 zones	1 zone	2 zones
$N_A = 4$	95.30%	99.96%	80.98%	99.12%	96.96%	99.99%
$N_A = 9$	98.22%	100%	80.88%	99.34%	98.54%	100%
$N_A = 16$	94.54%	100%	81.64%	97.66%	94.62%	100%
$N_A = 25$	95.32%	100%	81.26%	99.12%	96.38%	100%
$N_A = 36$	98.38%	100%	81.48%	98.94%	98.94%	100%
$N_A = 49$	96.04%	99.98%	81.00%	99.20%	96.82%	100%

V.4.3.4 Impact of the noise parameters σ_γ and σ_ξ

The impacts of additive noises are then taken into consideration, while setting $N_Z = 9$, $N_A = 9$ and $N_P = 81$. We first set σ_γ equal to 1% of that of noiseless acceleration signals and look into that of σ_ξ . Since $N_A = 9$ and $N_P = 81$, the positions of anchors are required to be moved slightly (+1) in order to prevent the case when their positions are overlapped with reference positions. Table V.5 shows the algorithm's performance while varying σ_ξ . It is observed that the zoning hitting rate decreases with the increasing values of σ_ξ for only-fingerprints method and the combined one. By taking also the second choice zone, the hitting rate drops below 100% but is still very close to 100%. Note that, in these simulations, N_A is set to be 9 which enhances the hitting rate of the first choice but lacking robust when noises largely increase. Comparatively, increasing N_A to 16 leads to 100% hits for the two selected zones possibilities.

Table V.5: Zones hitting rate while varying σ_ξ

σ_ξ	Fingerprints		Accelerometer		Combined	
	1 zone	2 zones	1 zone	2 zones	1 zone	2 zones
1%	100%	100%	80.88%	98.74%	100%	100%
3%	99.08%	100%	80.78%	99.16%	99.22%	100%
5%	98.12%	99.92%	80.40%	98.32%	98.32%	99.98%
8%	97.24%	99.86%	80.86%	98.30%	97.58%	99.98%
10%	96.44%	99.66%	80.70%	98.62%	97.12%	99.88%

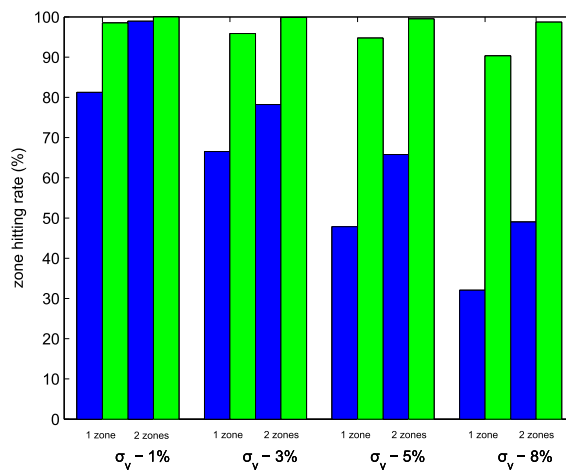
Afterwards, we keep $N_Z = 9$, $N_A = 9$, $N_P = 81$ and we set σ_ξ to 5% of that of noiseless RSSI signals and observe that of σ_γ . Table V.6 shows the algorithm's performance while varying the values of σ_γ . As expected, the zoning rate is inversely proportional to the noises standard deviation. However, the proposed combination method still outperforms the others.

Table V.6: Zones hitting rate while varying σ_γ

σ_γ	Fingerprints		Accelerometer		Combined	
	1 zone	2 zones	1 zone	2 zones	1 zone	2 zones
0.5%	98.06%	99.96%	98.38%	100%	98.50%	100%
1%	98.38%	99.96%	80.66%	99.02%	98.56%	100%
3%	98.22%	99.98%	66.92%	78.62%	98.36%	99.98%
5%	96.24%	99.96%	47.66%	66.02%	98.3%	100%

V.4.3.5 Impact of the acceleration incertitude

According to the results obtained in the sections above, the zoning hitting rate of acceleration based method drops greatly when the number of zone N_Z increases or when the acceleration noise standard deviation σ_γ becomes larger. Thus, this section aims at analyzing the accelerometer-based method, by comparing the computation of $[\hat{\mathbf{x}}](t)$ while using the boxed accelerations and velocities as shown in Section V.3.3 with δ_γ or using scalar positions to which the incertitude δ_{x_γ} is added as shown in Remark 3 of Chapter III. To do this, we set $N_Z = 9$ and we vary the value of σ_γ . Figure V.7 shows the correct zone hitting rates in blue for the boxed accelerations method using δ_γ and in green for scalar computations with additional incertitude using δ_{x_γ} . The results are illustrated with variable σ_γ for the accelerometer-based method while taking only the first zone (1 zone) or the first two zones (2 zones). Obviously, the zoning results by using δ_{x_γ} perform much better than that of boxed accelerations with δ_γ .


Figure V.7: Comparison between applications of acceleration incertitude

Moreover, the performances of the combined method is also analyzed using either δ_γ or δ_{x_γ} . To this end, we set equally $N_Z = 9$, $N_A = 9$, $N_P = 81$, then we vary the values of σ_ξ and σ_γ as shown in Table V.7. All results are the average of 100 attempts since the accelerometer method is pretty sensitive. Their zoning accuracies are actually very close but the latter with a very slim advantage.

V. ZONING LOCALIZATION

Table V.7: Zones hitting rate with δ_γ and δ_{x_γ}

$\sigma_\gamma, \sigma_\xi$	δ_\sim	Fingerprints		Accelerometer		Combined	
		1 zone	2 zones	1 zone	2 zones	1 zone	2 zones
1%, 5%	δ_γ	98.40%	99.98%	81.22%	98.92%	98.58%	100%
	δ_{x_γ}	98.34%	99.98%	98.49%	100%	99.05%	100%
3%, 5%	δ_γ	98.20%	99.97%	66.52%	78.21%	98.3200%	99.99%
	δ_{x_γ}	98.13%	99.99%	95.87%	99.87%	98.19%	100%
5%, 10%	δ_γ	96.70%	99.81%	47.80%	65.88%	96.84%	99.85%
	δ_{x_γ}	96.48%	99.73%	93.66%	99.41%	97.29%	99.87%

V.4.4 Comparison with SVM

The technique of zone localization could be categorized into a classification technique, since actually, it classifies the mobile node into its corresponding zone which is distinguished from other zones. On the contrary, when the zone number tends to be infinite, the zoning technique returns to be the normal localization technique as discussed before. A universal classification technique is known as the support vector machine (SVM) with kernel concept such as in [Sangnier *et al.* 2013; Tsochantaridis *et al.* 2004]. In the view of multi-classification, this part takes the comparison between the proposed method and the kernel-based classifier and the structural SVM as shown in [Alain Rakotomamonjy 2008; Thorsten Joachims 2008].

To this end, we configure the network using parameters $N_Z = 9$, $N_P = 81$ and $N_A = 9$. σ_γ is set to 1% of that of the noiseless acceleration signals and then the value of σ_ξ is varied simultaneously for the proposed method and SVM methods. Table V.8 reveals the simulation results in terms of zone hitting rate of 100 nodes positions over 50 attempts, where both fingerprints method and combined method outperform the SVM methods. Moreover, the choice of two zones provides quasi-100% hitting rate in the case of failure of the first choice in our method.

Table V.8: Comparison to the SVM methods

σ_ξ	Fingerprints		Combined		SVM-a ^a	SVM-b ^b
	1 zone	2 zones	1 zone	2 zones		
1%	100%	100%	100%	100%	99%	94%
3%	99.08%	100%	99.22%	100%	98.58%	94%
5%	98.12%	99.92%	98.32%	99.98%	97.54%	95%
8%	97.24%	99.86%	97.58%	99.98%	96.28%	94%
10%	96.24%	99.96%	98.30%	100%	95.98%	93%

^a Kernel-based SVM.

^b Structural SVM with parameter $c=1000$.

V.5 Conclusion

Based on the idea of room-level localization, this chapter discussed an original zoning localization technique for wireless sensor networks. The proposed method is an

anchor-based method, using both fingerprints and mobility information based on interval analysis and belief function theory. It gives sets of possible zones by fingerprints method and accelerometer method, then combine them by intersecting the zones while considering their confidence factors. Simulation results show that the proposed method could ensure, at more than 90%, finding the right zone location by the first choice and 100% if there exists a second choice in most cases. Besides, comparing with some SVM multi-classification methods, the proposed method performs better, and it is especially competing with respect to time saving. Nevertheless, the proposed zoning technique remains to be improved, since it is simulated under a rather simple partition of surveillance area. Moreover, the frontier points are always tough to be classified correctly at one time when the outside interruption is strong enough. Thus, it is expected, in the further, to develop a more sophisticated and more universal zoning technique under real scenarios conditions.

REFERENCES

References

- [Alain Rakotomamonjy, 2008] Alain Rakotomamonjy. SVM and Kernel Methods Matlab code and related softwares. 2008. <http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html>. 138
- [Castro *et al.*, 2001] P. Castro, P. Chiu, T. Kremenek et R. Muntz. A probabilistic room location service for wireless networked environments. In *UbiComp 2001: Ubiquitous Computing*, pages 18–34. Springer, 2001. 122
- [Chiu *et al.*, 2009] Y.-M. Chiu, K. Wang, R.-H. Jan, Y.-J. Hu et T.-H. Ku. An efficient room-based indoor localization scheme for wireless sensor networks. 2009. 122
- [Dempster, 1967] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, pages 325–339, 1967. 123
- [Denoeux, 1997] T. Denoeux. Analysis of evidence-theoretic decision rules for pattern classification. *Pattern recognition*, 30 (7): 1095–1107, 1997. 123, 125
- [D’hoë *et al.*, 2008] K. D’hoë, G. Ottoy, J. P. Goemaere et L. De Strycker. Indoor room location estimation. *Advances in electrical and computer engineering*, 8 (2): 78–81, 2008. 122
- [Hotta *et al.*, 2012] S. Hotta, Y. Hada et Y. Yaginuma. A robust room-level localization method based on transition probability for indoor environments. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–8, Nov 2012. 122
- [Jaulin *et al.*, 2001] L. Jaulin, M. Kieffer, O. Didrit et E. Walter. *Applied interval analysis*. Springer, 2001. 124, 126
- [Krisp, 2013] J. M. Krisp. *Progress in location-based services*. Springer, 2013. 122
- [Lefevre *et al.*, 2002] E. Lefevre, O. Colot et P. Vannoorenberghe. Belief function combination and conflict management. *Information fusion*, 3 (2): 149–162, 2002. 123
- [Lo *et al.*, 2012] C.-C. Lo, C.-C. Chen, Y.-C. Tseng, J.-C. Chiang, K.-C. Feng, L.-C. Kuo et Y.-C. Wang. A room-based localization system using wireless triggers and pattern matching techniques. In *IEEE VTS Asia Pacific Wireless Communications Symposium B. Kyoto: Susumu Yoshida*, volume 4, pages 11–13, 2012. 122
- [Moore et Moore, 1979] R. E. Moore et R. Moore. *Methods and applications of interval analysis*, volume 2. SIAM, 1979. 124
- [Noury *et al.*, 2002] N. Noury, G. Virone et T. Creuzet. The health integrated smart home information system (his 2): rules based system for the localization of a human. In *Microtechnologies in Medicine & Biology 2nd Annual International IEEE-EMB Special Topic Conference on*, pages 318–321. IEEE, 2002. 122
- [Ristic et Smets, 2005] B. Ristic et P. Smets. Target identification using belief functions and implication rules. *Aerospace and Electronic Systems, IEEE Transactions on*, 41 (3): 1097–1103, 2005. 123
- [Sangnier *et al.*, 2013] M. Sangnier, J. Gauthier et A. Rakotomamonjy. Filter bank kernel learning for nonstationary signal classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3183–3187. IEEE, 2013. 138

REFERENCES

- [Schiller et Voisard, 2004] J. Schiller et A. Voisard. *Location-based services*. Elsevier, 2004. [122](#)
- [Shafer *et al.*, 1976] G. Shafer *et al.* *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976. [123](#)
- [Smets, 1993] P. Smets. Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of approximate reasoning*, 9 (1): 1–35, 1993. [125](#)
- [Smets et Kennes, 1994] P. Smets et R. Kennes. The transferable belief model. *Artificial intelligence*, 66 (2): 191–234, 1994. [123](#), [125](#), [126](#)
- [Stojmenovic, 2005] I. Stojmenovic. *Handbook of sensor networks: algorithms and architectures*, volume 49. John Wiley & Sons, 2005. [122](#)
- [Thorsten Joachims, 2008] Thorsten Joachims. An implementation of Support Vector Machines (SVMs). 2008. <http://svmlight.joachims.org/>. [138](#)
- [Tsochantaridis *et al.*, 2004] I. Tsochantaridis, T. Hofmann, T. Joachims et Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004. [138](#)

REFERENCES



Contents

VI.1 Conclusion

VI.2 Perspectives

This manuscript has developed several state estimation techniques for wireless sensor networks, using fingerprints and acceleration information. It addressed the problems of indoor localization and zoning, in the framework of interval analysis. This thesis proposes, at the very beginning, a centralized localization technique which serves for the centralized topology in small scale sensor networks using different mobility models. Then it extends the centralized approach to a clusterized localization scheme considering the needs of large-middle-scale networks. Here the inertial information of mobile node is farther considered to overcome the sensors rotations. Additionally, a Kalman filter based method is developed using the same readings. It comes to an end by a zone localization technique, with considering the room-level localization requirements, categorizing the object to the zone to which it actually belongs by using the belief functions theory and the interval analysis. This part gives a brief recalling over all discussed methods in the preceding chapters, then proposes future research based on this work.

VI. CONCLUSION AND PERSPECTIVES

VI.1 Conclusion

The research on the localization techniques in wireless sensor networks has been developed since a couple of decades. All sorts of ideas, assays even implements, emerge rapidly and massively over these years. This work attempts, and would like to do one's bit in this domain. The proposed methods are explored around the concept of uncertainty over the estimated positions. They aim to gather all ambiguous position estimates then reduce their scope in a certain way. With aid of the interval analysis technique and couples of measurements, it comes true. In the following, the proposed localization methods are briefly looked back.

- This manuscript begins with a relatively simple network configuration where a centralized localization is permitted to be implemented. In this network, there exist two kinds of sensors, anchors with known positions and mobile nodes with unknown positions. Utilizing the reference positions, the fingerprints method localizes then the mobile node by using the measured received signal strengths and position pattern recognition. Meanwhile, mobile nodes equipped with accelerometers provide also the mobility information, such as the accelerations. According to the proposed mobility models, nodes positions are easily to be concluded. Due to the fact of existing inevitable interruption from outside even inside factors, the exact estimates are indeterminate anyway. The proposed method introduces uncertainty over the measurements, signal strengths and accelerations, leading to boxed estimates in which the real positions are supposed to be. Having two estimated positions boxes, each deduced from a kind of measurements, the proposed method only needs to intersect them consequently which in fact reduces the range of nodes possible locations. The updated box serves then for the localization in the next movement.
- A clusterized localization scheme is equally developed since the large-middle scale networks are more popular and adaptive in most cases. It differs in the network's configuration from the centralized method. Indeed, it proceeds to fingerprints collections in a clusterized framework where the surveillance area is divided into several individual blocks with central individual computation agents. Estimation at these agents leads to several local estimates of nodes' locations, whose combination using blocks weight factors leads to one global estimate. This scheme considers moreover the nodes inertial information, provided by another device, gyroscope, to take into consideration the occurring nodes rotation which is normal when nodes are moving or even static. Besides the solution by interval analysis, it proposes further a Kalman filter based method using respectively the information of RSSI and accelerations as the observation and the prediction equations.
- Room-level localization is meaningful and practical especially for the location-based services applications. The proposed localization idea is extended further into the zone localization scheme, where the entire surveillance area is composed of several zones, or considered to be in room units. The objective is to localize the

mobile object into the zone where it belongs. Similarly to the centralized localization, the zoning method performs fingerprints collection over all zones and nodes trajectory then infers the K nearest neighbors algorithm. Instead of utilizing the position coordinate information of neighbors positions, it uses their zone coordinates and their respective weights to make a conclusion of the concerned zones. At the same time, nodes infer their zone addressing by using acceleration information and the interval analysis technique. By appending an uncertainty over the measured accelerations values, the computed position turns to be also in terms of intervals, namely boxes with certain surfaces over specific zones. Using the belief functions theory, both kinds of measurements lead to sets of possible zones, with some associated confidence factors, whose combination narrows down the zones possibilities where the nodes reside. Generally, the proposed zoning method offers two choices of estimated zones in a specific order, ensuring by this a zone hitting rate with more than 90% for the first choice, and up to 100% for the second choice when the first is failed.

The performances of the proposed methods are analyzed with various settings of parameters. They are also taken into comparisons with kinds of other methods, either based on the same or even different ideas with diverse networks configurations. Generally speaking, the proposed methods harvest better localization accuracy and lower computation cost in most cases. The shortages exist however and remain to be discussed in the following.

VI.2 Perspectives

The proposed localization methods aim to locate a mobile node as accurate as possible but with time and energy consumption as poor as possible. The conception of combination and interval analysis furnishes a possible solution of realizing such objective. However, the proposed methods still have some aspects to be improved and worthy to work on in the future.

- KNN selection: There exists occasionally a phenomenon where the selected K nearest neighbors are not exactly the right ones. In other words, some of the geographically closest positions are excluded from the neighbors set when they are too close to an anchor compared to the others as shown in the example of Figure V.6. This problem leads to a slight geographical bias added to the final position estimate, however it might yield a totally wrong addressing for the room-level localization. In fact, this is caused by the nature of the relationship existing between RSSI and distances separating two sensors. The closer they are getting, the more rapidly the RSSI is growing. Thus, in the context of KNN, one imagined solution is to formulate further the obtained RSSI values until they could be correctly distinguished by KNN. Or else beginning with KNN method itself, revise it until it is possible to make correct selection from the obtained RSSI values. One

VI. CONCLUSION AND PERSPECTIVES

could also work on the optimization of the positioning of the reference positions with respect to anchors beforehand, to overcome such a problem.

- Interval computations: As it is discussed in Chapter III when computing the weight factors, there exist several problems over the division operation in terms of interval analysis. On the one hand, it cannot divide an interval including zero in the localization background which leads to an infinite weight value. On the other hand, it enlarges the range much more than expected which leads to the over-dilution interval/box. That is also why the weight equations are reformulated as shown in Chapter III in this case. A similar question raises equally as the incertitude δ_γ and δ_ξ are added. In fact, there exist a trade-off between the incertitude and the final estimates accuracy. If they are much smaller than normal values, the boxed estimates fail to include the real position. If, however, they are settled to be much larger, the resulting boxes are meaningless since any infinite box includes the real position. Based on the nature of interval computation, the determination of intervals remains to be an open question.
- Choices between both methods: The proposed algorithm considers an alternative picking-up strategy when there is not any intersection between two individual estimates in the unpredicted case. Practically, it is assumed that the fingerprints method is more reliable than the accelerometer based method since an abrupt change over the measured accelerations leads perhaps to an unexpected result. Hence, when the estimated boxes did not intersect with each other, the results from fingerprints will be adopted. However, this measure sometimes makes larger errors. It happens also even when there exists intersection between two boxes that the combined final error is larger than either of them. Thus, the accept-and-refusal strategy remains to be discussed.
- Zoning problem: The proposed algorithm makes the assumption of equal zones, which allows the problem addressing by using the KNN algorithm over the reference zones. However, in real scenarios, zones are rooms, offices, corridors or others, having different architectures and dimensions. One possible solution consists of dividing the whole space to equal zones, independently of the global architecture, or alternatively choosing specific dimensions for zones, allowing a subdivision of all rooms to equal zones. Future works will deal with the classification problem where any space structure could be handled.

Besides the existing technical aspects inside of the localization algorithms stated above, the proposed methods are expected to be further analyzed or improved as below.

- Real data: It is worth noting that all the simulations presented in this manuscript are based on the simulated data, like the trajectories, the RSSI measurements, the accelerations measurements and the rotation readings. This work merits then a further operation while using the real measurements in the real wireless sensor networks settings. Once the real fingerprints and inertial information are available, the proposed method in the real case will be put into effect.

- Accelerometer device: This work demonstrates the combined localization strategy using fingerprints and accelerometer readings. One thing that is controversial is the reliability of the accelerometer device. Massive implements and experiments illustrate that it is prone to deviate gradually from the real values, especially with its inevitable accumulation errors as time goes by. In this context, the estimates from accelerometer turns to be not so favorable compared with fingerprints methods when the mobile object goes farther. Thus, in some special cases where the unpredicted changes occur over the device, the results of accelerometer-based approach remain to be discussed.


VI. CONCLUSION AND PERSPECTIVES

A

Appendix: Résumé en Français

Contents

- A.1 Introduction aux réseaux de capteurs sans fil**
 - A.2 Théorie des intervalles**
 - A.3 Méthode de localisation centralisée**
 - A.4 Méthode de localisation clusterisée**
 - A.5 Extension au zonage**
 - A.6 Conclusion et perspectives**
 - References**
-

 réseaux de capteurs sans fil constituent une technologie en plein développement depuis les deux dernières décennies. Ils ont été votés comme l'une des dix premières technologies pouvant bouleverser le monde au vu de leur déploiement facile et de leurs capacités grandissantes de communication et de traitement de données. De nos jours, les réseaux de capteurs sans fil sont appliqués dans tous les horizons de la vie, comme par exemple dans la surveillance des machines dans les usines, le suivi des patients dans les hôpitaux, les reliefs de catastrophes géologiques, la surveillance de la migration ou l'exploration de terrains... Un problème fondamental existant dans les réseaux de capteurs sans fil est l'auto-localisation des capteurs, qui offre les emplacements géographiques des nœuds dans le réseau. Ce problème influence la performance ultime des réseaux puisque presque toutes les mesures effectuées par les capteurs sont liées aux positions où elles sont faites. Dans ce manuscrit, nous abordons le problème de la localisation indoor des capteurs mobiles dans les réseaux de capteurs sans fil. Les solutions proposées, basées sur des données de fingerprinting et de mobilité, se situent dans le cadre des intervalles. Cette partie résume les travaux réalisés autour de ce problème. Elle commence par une introduction générale sur les réseaux de capteurs, suivie par une brève description de la théorie des intervalles. Dans ce qui suit, les méthodes de localisation proposées sont présentées dans l'ordre, suivies enfin d'une conclusion et des perspectives.

A.1 Introduction aux réseaux de capteurs sans fil

Les réseaux de capteurs sans fil (RCSFs), nés dans les années 1990, sont composés d'un grand nombre de capteurs intelligents [Akyildiz *et al.* 2002; Zhao et Guibas 2004]. Ces capteurs sont de petites tailles, faciles à déployer, à bas coût et ayant la possibilité de communiquer, mesurer et traiter des informations [Bharathidasan et Ponduru 2002; Ilyas et Mahgoub 2004]. Grâce à ces atouts, les RCSFs fournissent une solution intelligente pour de nombreuses applications, contribuant ainsi à l'amélioration du mode de vie des personnes dans le monde. Cette section présente d'abord des informations générales sur les RCSFs et leurs applications et aborde par la suite le problème de localisation.

A.1.1 Présentation générale

Les différentes propriétés des RCSFs les rendent très attractifs dans un très grand nombre de domaines, couvrant le domaine militaire, l'environnemental, le médical, le civil, et bien d'autres, dans le but de la surveillance et du suivi. Dans le domaine militaire par exemple, les caractéristiques de déploiement rapide, d'auto-organisation et de tolérance de panne permettent aux RCSFs de répondre rapidement aux divers changements des champs de bataille. Dans ce sens, un système basé sur les RCSFs visant à la détection et la localisation des snipers [Simon *et al.* 2004] a été développé. L'armée des États-Unis a mené une série de tests à l'aide de RCSFs au désert de Californie [Hill 2003]. L'agence pour les projets de recherche avancée de défense ou DARPA des États-Unis avait lancé le projet SENSIT (technologie de l'information du capteur) pour construire des RCSFs ubiquitaires pour réaliser toute sorte de mesures comme la lumière, le son, les vibrations, le magnétisme, l'humidité, la pression, la température, la pollution, l'accélération, etc. Dans le domaine environnemental, un projet célèbre a été développé au cours des dernières années sur les habitats fauniques : le grand projet de l'île de canard [Mainwaring *et al.* 2002] mené par une équipe d'ingénieurs informaticiens de l'université de Californie à Berkeley pour améliorer la recherche des biologistes sur les comportements des oiseaux de mer. Dans le domaine médical, des RCSFs biomédicaux sont conçus pour mesurer une variété de signes vitaux des patients pour la surveillance de leurs états de santé en temps réel [Hande *et al.* 2006]. Ainsi, les RCSFs biomédicaux peuvent améliorer vivement la qualité des soins de santé. Ils pourraient servir également pour surveiller les signes vitaux de pompiers, suivre les handicapés dans un environnement sans surveillance, assurer le baby-sitting, suivre des activités des personnes atteintes d'Alzheimer, etc. Les RCSFs permettent également l'amélioration du mode de vie dans le domaine de la vie civile, impliquant la surveillance de la résidence, le transport, l'industrie et toute sorte d'infrastructures critiques dans la vie quotidienne. Par exemple, [Dagher *et al.* 2014] a conçu un système de navigation basé sur les RCSFs et le concept de ville intelligente. Il fournit un service de navigation locale pour les citoyens conduisant un véhicule à l'aide d'un système intelligent d'éclairage de rues. D'un autre côté, il existe principalement trois types de topologies dans les RCSFs définissant la façon dont les capteurs communiquent ensemble : centralisée, distribuée, et

hybride [Mourad 2010]. Dans la topologie centralisée, la plus simple, toutes les données mesurées sont acheminées à la station de fusion centrale où elles seront stockées et traitées. Les nœuds dans ce genre de réseaux n'effectuent pas de calcul ou de traitement. Ce genre de réseaux pourrait fournir une haute qualité de traitement mais tout en consommant énormément d'énergie, surtout pour les capteurs situés à proximité de la station centrale. Au contraire, les capteurs dans la topologie distribuée ont la capacité d'effectuer tous les traitements nécessaires, n'échangeant avec leurs voisins que des données traitées, d'habitude de taille réduite. Ce type de réseaux est beaucoup plus robuste à la défaillance que le mode centralisé, plus fragile face au risque de défaillance de la station centrale ou l'interférence des mesures longuement acheminées lors de la transmission. Toutefois, la topologie distribuée nécessite des algorithmes distribués qui pourraient être plus complexes à développer et parfois moins précis que leurs versions centralisées. La topologie hybride combine les idées des deux autres topologies. Pratiquement, la transmission de l'information consomme plus d'énergie que le calcul. De ce fait, les réseaux sont divisés en plusieurs groupes, ou clusters. Chacun d'entre eux possède une tête de cluster qui joue le rôle de la station de fusion centrale. Assurant un équilibre optimal entre la consommation d'énergie et la capacité de traitement, la topologie hybride convient à des réseaux à grandes échelles [Bandyopadhyay et Coyle 2003].

Sur la base de l'architecture des réseaux, la communication entre capteurs peut s'établir de différentes manières, permettant de maintenir la connectivité du réseau. La communication par la fréquence radio est le mode de communication le plus fréquent, grâce à sa faible consommation énergétique et la nécessité d'un matériel simple. Toutefois, ce type de communication a un faible débit de données pendant la transmission, se basant sur les techniques de modulation et de démodulation. En outre, la qualité des signaux pourrait être sérieusement dégradée avec l'augmentation de la distance parcourue [Akyildiz *et al.* 2002]. La technologie Zigbee [Cheong *et al.* 2011] basée sur le protocole IEEE 802.15.4 pourrait être un choix alternatif dans le cas de déploiement d'un grand nombre de capteurs. Cette technologie a également une basse consommation, un bas coût, mais une faible capacité de pénétration dans les environnements intérieurs. Par comparaison, la bande ultra large (UWB) [Chehri *et al.* 2009], qui sert aux mesures de temps d'arrivée, est capable de pénétrer dans des murs épais et est plus robuste à la dégradation des signaux, mais elle a un coût de consommation relativement plus élevé et est beaucoup plus chère. La technologie Bluetooth [Erasala et Yen 2002] est anti-interférence, mais nécessite un périphérique de plus, consomme plus et coûte plus cher que cette dernière. Le WiFi, basé sur le protocole IEEE 802.11, est très fiable, mais s'appuie sur une station de communication puissante et des nœuds d'accès [Li *et al.* 2011]. Autres options existent comme celles de l'utilisation de l'identification par fréquence radio (RFID) [Choi *et al.* 2011], infrarouge [Gao et Guo 2010] ou ultrasons [Shahid *et al.* 2010; Runge *et al.* 2011]. Ce type d'approches nécessite en conséquence des dispositifs supplémentaires intégrés au capteur. Elles peuvent fournir une assez bonne précision mais elles sont normalement limitées à certains scénarios spécifiques comme en sous-mer, dans les jungles ou les terminaux de hand-hold. Comme le proverbe dit,

"chacun son métier".

A.1.2 Illustration du problème de localisation

L'un des objectifs de la surveillance ou du suivi par RCSFs est de rapporter en temps réel l'endroit où se produit l'événement considéré. Cet objectif nécessite la connaissance préalable des emplacements géographiques des capteurs eux-mêmes. Cela définit le problème de localisation [Yick *et al.* 2008].

Selon les applications, les capteurs sont soit déployés manuellement à des positions souhaitées au préalable, soit sur demande, dispersés au hasard dans un lieu déterminé sous la forme de parachutage suivi d'une auto-organisation, ce qui est normal dans les champs intouchables. Dans le cas où les capteurs ont initialement des positions inconnues ou qu'ils se déplacent de façon forcée, à cause de facteurs non contrôlables tels qu'un vent soufflant, de la vibration..., les positions enregistrées des capteurs ne sont plus fiables à utiliser. De ce fait, la localisation des nœuds devient un problème essentiel pour les réseaux ayant pour but de réaliser la surveillance ou le suivi.

Une solution intuitive à ce problème est d'équiper tous les capteurs avec les systèmes de positionnement (GPS) [Hofmann-Wellenhof *et al.* 2004] pour avoir un positionnement absolu. Cependant, cette solution implique deux faits impraticables, l'un lié aux environnements indoor où les signaux GPS ne sont pas fiables, l'autre lié aux coûts et à la consommation d'énergie car il serait nécessaire d'avoir des récepteurs GPS au nombre des capteurs, ce qui coûte cher et consomme beaucoup d'énergie pendant l'acquisition des signaux.

Face à l'impraticabilité du GPS, des solutions alternatives servant à la localisation dans les réseaux de capteurs ont vu le jour. La majorité de ces approches résout ce problème à l'aide d'**ancres**, qui sont des capteurs ayant des positions connues à l'avance. Cela pourrait être au moyen du GPS en cas de réseaux mobiles extérieurs, ou alors les ancres ont des emplacements fixes. Les autres capteurs restants sont appelés **nœuds**, ayant des positions inconnues, fixes ou mobiles. Le problème de localisation reviendra ainsi à utiliser les signaux échangés entre ancres et nœuds avec les mesures relevées aux nœuds, comme la température ou autre, afin de déterminer leurs emplacements.

Dans le cadre de cette thèse, le problème de localisation est défini dans un scénario indoor où les ancres sont fixes et les nœuds se déplacent dans un espace D -dimensionnel de \mathfrak{R}^D , avec $D = 2$ ou $D = 3$. On considère que N_A ancres et N_X nœuds sont déployés dans le même champ. Les coordonnées des ancres et celles des nœuds sont alors données par des vecteurs de dimensions D , notés respectivement $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,D})$, $i \in \{1, \dots, N_A\}$, et $\mathbf{x}_j(t) = (x_{j,1}(t), \dots, x_{j,D}(t))$, $j \in \{1, \dots, N_X\}$, où t symbolise l'instant discret. L'objectif serait ainsi de calculer les $\mathbf{x}_j(t)$ sachant les positions \mathbf{a}_i et les signaux échangés entre eux. Il existe de nombreuses techniques de localisation à base d'ancres répondant aux exigences des réseaux de capteurs. Parmi celles-ci, on cite celles basées sur l'estimation de la distance, les mesures de connectivités et la collection des informations de scénarios ou fingerprinting [Mao *et al.* 2007]. Elles sont également catégorisées, selon le traitement des données, entre méthodes centralisées, clusterisées ou décentralisées [Stojmenovic 2005; Pal 2010], ou autrement en terme de signaux util-

isés, entre méthodes basées sur l’acoustique, l’optique, la puissance des signaux reçus (RSSI), etc... [Schweinzer et Syafrudin 2010; Sifuentes *et al.* 2011; Kaemarungsi et Krishnamurthy 2004; Bahl et Padmanabhan 2000], Plus de détails sur les méthodes de localisation existantes sont fournis dans le chapitre I.

Ce manuscrit traite le problème de localisation dans un environnement indoor, les solutions proposées étant basées sur la technique du fingerprinting utilisant les RSSIs. Elles combinent à celles-ci d’autres mesures utiles, telles que les informations d’inertie en provenance d’accéléromètres et de gyroscopes, dans le but de corriger et d’affiner les positions estimées. La résolution du problème se situe dans le cadre de la théorie des intervalles, permettant d’inclure toutes les solutions du problème ainsi que les incertitudes autour de leurs valeurs. La théorie des intervalles est introduite dans la section suivante.

A.2 Théorie des intervalles

Dans la majorité des applications d’ingénierie, le problème de non linéarité est fréquemment rencontré, avec une nécessité de solution robuste. Ce problème mathématique est normalement représenté sous la forme d’un ensemble d’équations non linéaires, d’inégalités non linéaires, ou d’une minimisation globale d’une fonction non convexe. L’analyse par intervalles fournit un cadre permettant de résoudre ce type de problème d’une manière garantie.

La théorie des intervalles a débutée dans les années 1950, motivée par la procédure du calcul de $\sqrt{2}$ illustrée dans la figure Figure II.1. Au début du calcul, le fait que le nombre 1 est l’entier maximal dont le carré est inférieur à 2, implique que la solution est contenue dans l’intervalle réel $[1, 2]$. De la même manière, le plus grand réel à un chiffre après la virgule dont le carré est inférieur à 2 étant 1.4, l’intervalle de recherche est réduit à $[1, 4, 1, 5]$, puis $[1, 41, 1, 42]$, $[1, 414, 1, 415]$, etc... [Sunaga 1958; 2009] En conséquence, la notion d’intervalle est équivalente au nombre réel mais plus illustrable. Alors, pour décrire une quantité physique, la représentation d’un intervalle devient plus convenable qu’un nombre réel concret, permettant de relier les mathématiques pures à la réalité, ainsi que l’analyse pure à l’analyse appliquée.

Depuis, l’analyse par intervalle s’est fait des développements remarquables à des moments historiques [Sunaga 1958; Moore 1966; Kearfott et Kreinovich 1996; Jaulin *et al.* 2001a]. Elle sert au calcul mathématique visant des ensembles, au lieu des chiffres numériques. Elle permet de résoudre les problèmes insolubles de façon normale, tels que les problèmes définis par un ensemble d’équations non linéaires [Neumaier 1990], ou d’inégalités non linéaires [Moore 1992; Jaulin et Walter 1993; Walter et Jaulin 1994] ainsi que les problèmes d’optimisation globale de fonctions non convexes [Hansen et Walster 2003]. La solution garantie est fournie en incluant toutes les solutions possibles du problème abordé dans un intervalle. Ceci s’avère possible même dans le contexte de fortes non-linéarités ou d’ensembles discontinus de solutions, grâce aux propriétés robustes d’un intervalle. Dans cette section, nous introduisons brièvement le concept du calcul par intervalles, suite à la définition de base.

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

A.2.1 Définition des intervalles

L'ensemble de toutes les variables réelles ξ respectant la condition de $\alpha \leq \xi \leq \beta$ est appelé un intervalle et est désigné par $[\alpha, \beta]$. Alors en général, dans un espace réel \mathbb{R} , un intervalle réel $[x]$ est un sous-ensemble connexe inclus dans \mathbb{R} et défini comme suit,

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}, \quad (\text{A.1})$$

où les notions \underline{x} et \bar{x} désignent respectivement la borne inférieure $\inf [x]$ et la borne supérieure $\sup [x]$ de l'intervalle $[x]$, pouvant être finies ou infinies. L'ensemble vide \emptyset est également considéré comme un intervalle réel si \emptyset représente l'absence de solution. L'ensemble de tous les intervalles réels est noté par \mathbb{IR} . L'intervalle réel peut être caractérisé par son centre $c([x])$ et sa largeur $w([x])$, définis comme suit,

$$\begin{aligned} c([x]) &= \frac{\underline{x} + \bar{x}}{2}, \\ w([x]) &= \bar{x} - \underline{x}. \end{aligned} \quad (\text{A.2})$$

$w([x])$ possède également l'attribut de la taille de l'intervalle $[x]$, déterminée par l'incertitude présente sur la quantité x . Dans le cas où $w([x])$ est nulle, l'intervalle est dégénéré à un nombre réel x . En utilisant les notions $c([x])$ et $w([x])$, un intervalle $[x]$ peut être reformulé comme suit,

$$[x] = c([x]) + \left[-\frac{1}{2}w([x]), \frac{1}{2}w([x]) \right] = c([x]) + \frac{1}{2}w([x])[-1, 1]. \quad (\text{A.3})$$

Cette formulation est très pratique pour les opérations de l'analyse par intervalles, comme démontré dans la section A.3. La définition de l'intervalle peut être étendue au cas multidimensionnel. Soit $[\mathbf{x}]$ un intervalle vectoriel de \mathbb{R}^n . Il est défini par le produit cartésien de n intervalles réels comme suit,

$$[\mathbf{x}] = [x_1] \times [x_2] \times \dots \times [x_n], \quad i = 1, \dots, n. \quad (\text{A.4})$$

$[\mathbf{x}]$ est appelé aussi pavé ou boîte, dont l'élément $[x_i] = [x_i, \bar{x}_i]$ désigne la $i^{\text{ème}}$ composante de l'intervalle obtenue par projection de $[\mathbf{x}]$ sur le $i^{\text{ème}}$ axe. L'ensemble vide de \mathbb{R}^n est noté comme $\emptyset_1 \times \emptyset_2 \times \dots \times \emptyset_n$ si et seulement si toutes les composantes sont vides. L'ensemble des pavés dans l'espace \mathbb{R}^n est noté \mathbb{IR}^n . Les bornes inférieures et supérieures sont composées des bornes de toutes les composantes comme suit,

$$\begin{aligned} \underline{\mathbf{x}} &= (\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n)^T, \\ \bar{\mathbf{x}} &= (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)^T. \end{aligned} \quad (\text{A.5})$$

La taille de $[\mathbf{x}]$ est définie comme la taille maximale des différents éléments,

$$w([\mathbf{x}]) = \max_{1 \leq i \leq n} w([x_i]), \quad (\text{A.6})$$

et le centre de $[\mathbf{x}]$ est donné par

$$c([\mathbf{x}]) = (c([x_1]), c([x_2]), \dots, c([x_n]))^T. \quad (\text{A.7})$$

A.2.2 Analyse par intervalles

Une fois le concept de l'intervalle est défini, il serait intéressant d'introduire les règles de calcul au cadre des intervalles, la théorie de l'analyse par intervalles étant fondée sur la théorie des ensembles et son calcul étant en fait un cas particulier du calcul sur des ensembles. Sur cette base-là, des opérations utiles sont élaborés et définies par conséquence, telles que les opérations ensemblistes et arithmétiques et les fonctions d'inclusion. Dans cette section, nous introduisons uniquement les opérations et fonctions de base. Plus de détails sont fournis dans le Chapitre II.

A.2.2.1 Opérations de base

Un intervalle réel possède une double nature, à savoir celle d'un nombre réel et celle d'un ensemble, ce qui permet de lui appliquer des opérations du nombre et de l'ensemble [Moore et Moore 1979; Kearfott et Kreinovich 1996; Jaulin *et al.* 2001a]. Nous présentons ici les opérations de base, notamment les opérateurs ensemblistes et arithmétiques.

◇ *Opérations ensemblistes scalaires :*

Similairement avec la théorie des ensembles, les opérateurs de base sont également adaptés aux intervalles. Soient deux intervalles réels $[x]$ et $[y]$ de \mathbb{IR} à la condition que $\underline{x} \geq \underline{y}$ et $\bar{x} \leq \bar{y}$, ils ont naturellement la relation d'inclusion, décrite comme suit,

$$[x] \subseteq [y], \text{ si } \forall x \in [x], x \in [y]. \quad (\text{A.8})$$

De la même manière, l'intersection des intervalles $[x]$ et $[y]$, qui est également un intervalle, est décrite comme suit,

$$[x] \cap [y] = \begin{cases} \emptyset, & \text{si } \bar{x} < \underline{y} \text{ ou } \bar{y} < \underline{x}, \\ [\max\{\underline{x}, \underline{y}\}, \min\{\bar{x}, \bar{y}\}], & \text{sinon.} \end{cases} \quad (\text{A.9})$$

Les intervalles n'héritent pas exactement de l'union ensembliste qui aurait dû être définie par $[x] \cup [y] = \{z \in [x] \text{ ou } z \in [y]\}$. Dans le cas de deux intervalles disjoints $[x]$ et $[y]$, une nouvelle opération de réunion des intervalles est définie par,

$$[x] \sqcup [y] = [\min\{\underline{x}, \underline{y}\}, \max\{\bar{x}, \bar{y}\}]. \quad (\text{A.10})$$

Un intervalle $[x]$ privé de l'intervalle $[y]$ est définie comme $[x] \setminus [y] = [\{x \in [x] \mid x \notin [y]\}]$. Alternativement, il est possible de l'exprimer comme suit,

$$[x] \setminus [y] = \begin{cases} \emptyset, & \text{si } [x] \subseteq [y], \\ [\underline{x}, \underline{y}], & \text{si } \underline{x} \leq \underline{y} \leq \bar{x} \leq \bar{y}, \\ [\bar{y}, \bar{x}], & \text{si } \underline{y} \leq \underline{x} \leq \bar{y} < \bar{x}, \\ [x], & \text{sinon.} \end{cases} \quad (\text{A.11})$$

◇ *Opérations ensemblistes en cas de pavés :*

Un pavé $[\mathbf{x}] \in \mathbb{IR}^n$ est inclus dans un autre pavé $[\mathbf{y}] \in \mathbb{IR}^n$ si toutes les composantes réelles de $[\mathbf{x}]$ sont incluses dans celles du $[\mathbf{y}]$,

$$[\mathbf{x}] \subseteq [\mathbf{y}], \text{ si } \forall i, [x_i] \subseteq [y_i], 1 \leq i \leq n. \quad (\text{A.12})$$

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

L'intersection de deux pavés $[\mathbf{x}] \in \mathbb{I}\mathbb{R}^n$ et $[\mathbf{y}] \in \mathbb{I}\mathbb{R}^n$, qui est également un vecteur de $\mathbb{I}\mathbb{R}^n$, est définie par,

$$[\mathbf{x}] \cap [\mathbf{y}] = \begin{cases} \emptyset^n, & \text{si } \exists i, 1 \leq i \leq n, [x_i] \cap [y_i] = \emptyset, \\ ([x_1] \cap [y_1]) \times ([x_2] \cap [y_2]) \times \dots \times ([x_n] \cap [y_n]), & \text{sinon,} \end{cases} \quad (\text{A.13})$$

où \emptyset^n est le produit cartésien de n \emptyset . La notion \emptyset^n sera représentée par \emptyset tout simplement.

Similairement, l'union de deux pavés $[\mathbf{x}] \in \mathbb{I}\mathbb{R}^n$ et $[\mathbf{y}] \in \mathbb{I}\mathbb{R}^n$ est le produit cartésien des unions de toutes les composantes des intervalles $[\mathbf{x}]$ et $[\mathbf{y}]$,

$$[\mathbf{x}] \sqcup [\mathbf{y}] = ([x_1] \sqcup [y_1]) \times ([x_2] \sqcup [y_2]) \times \dots \times ([x_n] \sqcup [y_n]). \quad (\text{A.14})$$

Un pavé $[\mathbf{x}]$ privé d'un pavé $[\mathbf{y}]$ produisant un pavé désigné par $[\mathbf{z}] = [\mathbf{x}] \setminus [\mathbf{y}]$, dont la composante réelle $[z_i]$ est illustrée de plus comme suit,

$$[z_i] = \begin{cases} [x_i] \setminus [y_i], & \text{si } \forall j \neq i, 1 \leq j \leq n, [x_j] \subseteq [y_j], \\ [x_i], & \text{sinon.} \end{cases} \quad (\text{A.15})$$

◇ *Opérations arithmétiques :*

Les intervalles réels sont composés de nombres réels. Les quatre opérateurs de base de l'arithmétique des réels, notamment l'addition (+), la soustraction (−), la multiplication (*) et la division (/), sont adaptés également à l'arithmétique des intervalles profitant de la double nature. Soit le symbole ◇ un opérateur de la liste (+, −, *, /). Alors, l'expression d'opération arithmétique des intervalles $[x]$ et $[y]$ est donnée comme suit,

$$[x] \diamond [y] = [\{ x \diamond y \in \mathbb{R} \mid x \in [x], y \in [y] \}], \text{ avec } \diamond \in \{+, -, *, /\}. \quad (\text{A.16})$$

L'opération concernée est effectuée en utilisant les bornes inférieures et supérieures des intervalles,

$$[x] \diamond [y] = [\min\{\underline{x} \diamond \underline{y}, \underline{x} \diamond \bar{y}, \bar{x} \diamond \underline{y}, \bar{x} \diamond \bar{y}\}, \max\{\underline{x} \diamond \underline{y}, \underline{x} \diamond \bar{y}, \bar{x} \diamond \underline{y}, \bar{x} \diamond \bar{y}\}]. \quad (\text{A.17})$$

Notez que, cette définition pour la division n'est vraie que dans le cas où $[y]$ ne comprend pas le 0. L'équation A.17 peut être simplifiée en outre pour les opérations d'addition et de soustraction comme suit,

$$\begin{aligned} [x] + [y] &= [\underline{x} + \underline{y}, \bar{x} + \bar{y}], \\ [x] - [y] &= [\underline{x} - \bar{y}, \bar{x} - \underline{y}]. \end{aligned} \quad (\text{A.18})$$

En plus, les opérations telles que la négation et la puissance carrée sont aussi applicables,

$$\begin{aligned} -[x] &= [\{-x \mid x \in [x]\}] = [-\bar{x}, -\underline{x}], \\ [x]^k &= \{x^k \mid x \in [x]\}, \quad k \in \{1, 2, \dots\}. \end{aligned} \quad (\text{A.19})$$

À condition que les bornes des arguments sont strictement positives, les intervalles réels engendrés par la multiplication, la division et la puissance carrée sont donnés par la suite,

$$\begin{aligned} [x] * [y] &= [\underline{x} * \underline{y}, \bar{x} * \bar{y}], \\ [x]/[y] &= [\underline{x}/\bar{y}, \bar{x}/\underline{y}], \\ [x]^2 &= [\underline{x}^2, \bar{x}^2]. \end{aligned} \tag{A.20}$$

Notez que dans l'équation A.19, $[x]^2 \neq [x] * [x]$ ce qu'il faut distinguer lors du calcul des intervalles.

◇ *Opérations arithmétiques en cas de pavés :*

Les opérations arithmétiques définies pour les intervalles peuvent être aussi étendues aux pavés, où le symbole \diamond est en qualité d'opérateur binaire (+, −, *, /, ^) comme suit,

$$[\mathbf{x}] \diamond [\mathbf{y}] = ([x_1] \diamond [y_1]) \times ([x_1] \diamond [y_1]) \times \dots \times ([x_n] \diamond [y_n]). \tag{A.21}$$

Les propriétés algébriques classiques au sujet de l'arithmétique, telles que la commutativité et l'associativité appliquées sur l'addition et la multiplication sont également correctes pour les opérations des intervalles. Nous citons comme suit,

$$\begin{aligned} [\mathbf{x}] + [\mathbf{y}] &= [\mathbf{y}] + [\mathbf{x}], \\ [\mathbf{x}] * [\mathbf{y}] &= [\mathbf{y}] * [\mathbf{x}], \\ \alpha \cdot [\mathbf{x}] &= (\alpha \cdot [x_1]) \times (\alpha \cdot [x_2]) \times \dots \times (\alpha \cdot [x_n]), \\ [\mathbf{x}]^T * [\mathbf{y}] &= [x_1] * [y_1] + [x_2] * [y_2] + \dots + [x_n] * [y_n], \end{aligned} \tag{A.22}$$

où $\alpha \in \mathbb{R}$, T désigne l'opération de transposition. Notez que, la loi distributive de l'arithmétique ordinaire ne parvient pas à s'y appliquer. Toutefois, on tombe dans une loi sous-distributive telle que

$$[x] * ([y] + [z]) \subseteq [x] * [y] + [x] * [z]. \tag{A.23}$$

En outre, la propriété d'inclusion est combinée avec les opérateurs arithmétiques des intervalles comme suit,

$$[x_1] \diamond [y_1] \subseteq [x_2] \diamond [y_2], \text{ si } [x_1] \subseteq [x_2] \text{ et } [y_1] \subseteq [y_2]. \tag{A.24}$$

Toutes ces propriétés nous offrent ainsi la faisabilité et le fondement théorique de l'analyse par intervalles, applicable à la résolution du problème d'approximation.

A.2.2.2 Fonctions d'inclusion

Considérons une fonction de mapping $f(\mathbf{x})$ associée à une variable $\mathbf{x} \in \mathbb{R}^n$ définie comme suit,

$$\mathbf{f} : \mathbf{x} \in \mathbb{R}^n \mapsto \mathbf{f}(\mathbf{x}) \in \mathbb{R}^m. \tag{A.25}$$

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

Si l'on suppose que l'argument \mathbf{x} de \mathbf{f} est un vecteur appartenant à un pavé $[\mathbf{x}] \in \mathbb{IR}^n$, alors la fonction d'intervalles associée à $[\mathbf{x}]$ est donnée par l'ensemble suivant,

$$\mathbf{f}([\mathbf{x}]) = \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\}. \quad (\text{A.26})$$

L'ensemble $\mathbf{f}([\mathbf{x}])$ n'est pas forcément un intervalle connexe. La fonction d'inclusion, notée $[\mathbf{f}]$, est produite pour remédier à cette question. Elle est définie de telle sorte à engendrer un pavé englobant $\mathbf{f}([\mathbf{x}])$ comme suit,

$$\begin{aligned} [\mathbf{f}]([\mathbf{x}]) &= \{\{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\}, \text{ satisfaisant} \\ \forall [\mathbf{x}] \subset \mathbb{IR}^n, \mathbf{f}([\mathbf{x}]) &\subset [\mathbf{f}]([\mathbf{x}]), \end{aligned} \quad (\text{A.27})$$

avec $[\mathbf{f}([\mathbf{x}]]) \in \mathbb{IR}^m$. Un schéma illustratif des fonctions d'inclusion avec $n = m = 2$ est présenté dans la Figure II.5 dans le chapitre II.

Il est évident qu'une fonction donnée \mathbf{f} peut avoir une infinité de fonctions d'inclusion. Néanmoins, il existe une fonction d'inclusion minimale pour \mathbf{f} qui donne le plus petit pavé englobant $\mathbf{f}([\mathbf{x}])$, notée par $[\mathbf{f}^*]([\mathbf{x}])$. Dans ce cas particulier, la fonction d'inclusion peut être définie par

$$[\mathbf{f}^*]([\mathbf{x}]) = [\min\{f(\underline{\mathbf{x}}), f(\bar{\mathbf{x}})\}, \max\{f(\underline{\mathbf{x}}), f(\bar{\mathbf{x}})\}]. \quad (\text{A.28})$$

Une fonction inclusion $[\mathbf{f}]$ est considérée comme monotone à condition qu'elle satisfasse,

$$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow [\mathbf{f}]([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{y}]). \quad (\text{A.29})$$

Considérons par exemple les fonctions exponentielle, logarithmique et sinusoïdale appliquées à l'argument $[\mathbf{x}] = [\underline{x}, \bar{x}]$. Leurs fonctions d'inclusion minimales sont données par

$$\begin{aligned} [\exp^*]([\mathbf{x}]) &= [\exp(\underline{x}), \exp(\bar{x})]. \\ [\log^*]([\mathbf{x}]) &= [\log(\underline{x}), \log(\bar{x})]. \\ [\sin^*]([\mathbf{x}]) &= [\sin(\underline{x}), \sin(\bar{x})], \text{ pour } x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]. \end{aligned} \quad (\text{A.30})$$

La fonction d'inclusion $[\mathbf{f}]$, définie de \mathbb{IR}^n à \mathbb{IR}^m , est alternativement représentée par le produit cartésien de m fonctions d'inclusion $[f_i] \in \mathbb{IR}, i \in \{1, 2, \dots, m\}$ de l'espace \mathbb{IR}^n . Par la suite, la fonction d'inclusion \mathbf{f} est donnée par

$$[\mathbf{f}]([\mathbf{x}]) = [f_1]([\mathbf{x}]) \times [f_2]([\mathbf{x}]) \times \dots \times [f_m]([\mathbf{x}]). \quad (\text{A.31})$$

D'autre part, pour vérifier si tous les éléments inclus dans un pavé satisfont une certaine propriété, les tests d'inclusion sont définis. Soit \mathbf{t} un test défini de \mathbb{R}^n dans $\mathbb{B} = \{0, 1\}$. Un test d'inclusion $[\mathbf{t}]$ est effectué de \mathbb{IR}^n à \mathbb{IB} . Pour tout $[\mathbf{x}] \in \mathbb{IR}^n$, $[\mathbf{t}]$ est déterminé par,

$$\begin{aligned} ([\mathbf{t}]([\mathbf{x}]) = 1) &\Rightarrow (\forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 1), \\ ([\mathbf{t}]([\mathbf{x}]) = 0) &\Rightarrow (\forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 0). \end{aligned} \quad (\text{A.32})$$

Ainsi, le résultat du test est vrai pour $t([\mathbf{x}]) = 1$ et faux pour $t([\mathbf{x}]) = 0$. Il reste cependant indéterminé pour $t([\mathbf{x}]) = [0, 1]$ et impossible pour un résultat d'ensemble vide.

La technique d'analyse par intervalles est une approche d'approximation numérique garantie, applicable sur les problèmes non linéaires rencontrés souvent dans le domaine de l'ingénierie. La théorie repose fondamentalement sur l'idée d'enfermer des nombres réels dans des intervalles, ou des vecteurs réels dans des pavés assurant des résultats garantis. Dans les sections suivantes, nous appliquerons la théorie des intervalles au problème de localisation.

A.3 Méthode de localisation centralisée

La capacité de localisation des capteurs est l'une des exigences fondamentales et essentielles dans un réseau de capteurs sans fil. Parmi les méthodes de localisation émergentes au cours des dernières années, celles basées sur les RSSI sont considérées avantageuses du point de vue de la simplicité, de la flexibilité ainsi que du coût de matériel. Au sujet du traitement des signaux, au lieu de convertir simplement les RSSI en distances, des méthodes alternatives font une comparaison entre la puissance du signal reçu et un seuil de puissance, conduisant à des données de connectivités et donc des contraintes sur les distances à combiner éventuellement à l'aide de l'algorithme séquentiel de Monte-Carlo [Hu et Evans 2004; Baggio et Langendoen 2006a], l'analyse par intervalles [Mourad *et al.* 2008; 2009a], le filtrage variationnel [Teng *et al.* 2010], etc... D'autres méthodes s'appuient sur la collection des informations de scénarios ou fingerprinting faisant la comparaison par la suite de la puissance à la réception du nœud aux informations déjà collectées dans le sens du pattern matching [Tapus et Siegart 2006; Robles *et al.* 2010]. En considérant également la mobilité des capteurs, des travaux ont exploité à la fois les données de connectivités et les contraintes de mobilité, fournissant leurs solutions à l'aide de Monte-Carlo [Hu et Evans 2004; Baggio et Langendoen 2006a], de l'analyse par intervalles [Mourad *et al.* 2008; 2009a] ou Hidden Markov [Fraile et Maybank 1998; Arthi et Murugan 2010], etc... D'autres méthodes basées sur des dispositifs supplémentaires existent telles que celles utilisant les magnétomètres collectant des données magnétiques et résolvant le problème par Monte Carlo [Haverinen et Kemppainen 2009]. D'autres utilisent l'identification par radio fréquence (RFID) [Ni *et al.* 2004] coopérant avec des capteurs ultrasons pour localiser les robots mobiles [Choi *et al.* 2011], ou installant un dispositif d'inertie donnant directement une estimation du déplacement du capteur mobile [Hsu et Yu 2009]. De notre côté, nous intégrons deux types de mesures, à savoir des radio-fingerprints et les accélérations, tout en tenant compte de l'incertitude des mesures, et nous résolvons le problème dans le cadre des intervalles.

A.3.1 Problématique

Nous proposons ici une méthode de localisation, basée sur les ancrs, et effectuant la collection des fingerprints dans une zone de surveillance de \mathbb{R}^D , où $D = 2$ ou $D = 3$ pour une zone de deux dimensions ou trois dimensions. Le réseau dispose de deux types de capteurs : *ancres* et *nœuds*. Les ancrs ont des positions fixes et connues notées par $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,D})$, $i \in \{1, \dots, N_A\}$, alors que les nœuds ont des positions inconnues, notées par $\mathbf{x}_j(t) = (x_{j,1}(t), \dots, x_{j,D}(t))$, $j \in \{1, \dots, N_X\}$, N_A et N_X étant respectivement les nombres des ancrs et des nœuds dans le réseau. Le problème de localisation consiste alors à trouver $\mathbf{x}_j(t)$, $j \in \{1, \dots, N_X\}$, en utilisant des mesures et des informations liées à \mathbf{a}_i , $i \in \{1, \dots, N_A\}$.

Une collection de fingerprints en RSSI est effectuée au préalable pour définir une cartographie de puissance sur le réseau. Pour ce faire, N_P *positions de référence* sont déployées uniformément sur la totalité du réseau, elles sont notées par $\mathbf{p}_k = (p_{k,1}, \dots, p_{k,D})$, $k \in \{1, \dots, N_P\}$. Un exemple de telle configuration est représenté dans Figure III.1. Les ancrs émettent dans le réseau des signaux avec une même puissance initiale. En plaçant consécutivement un capteur aux points de référence, ces signaux sont collectés à ces points et leurs puissances sont mesurées. Soit $\boldsymbol{\xi}_{P_k}$, $k \in \{1, \dots, N_P\}$ le vecteur de RSSIs de référence mesurés à la position \mathbf{p}_k , il est donné par

$$\boldsymbol{\xi}_{P_k} = (\xi_{P_k,1}, \dots, \xi_{P_k,N_A}), \quad k \in \{1, \dots, N_P\}, \quad (\text{A.33})$$

où $\xi_{P_k,i}$ est la puissance du signal émis par l'ancre i et reçu à la position de référence \mathbf{p}_k . La procédure de configuration conduit alors à une base de données de fingerprints constituée de N_P couples $(\mathbf{p}_k, \boldsymbol{\xi}_{P_k})$, $k \in \{1, \dots, N_P\}$. Cette base de données est stockée au centre de calcul pour être utilisée ultérieurement avec les mesures effectuées en temps réel par les nœuds afin de les localiser.

A.3.1.1 Localisation à base de fingerprints

Pendant la procédure de positionnement en temps réel, les nœuds se déplacent librement à travers le réseau et recueillent les puissances de signaux de toutes les ancrs de la même manière. Par la suite, le vecteur des RSSIs mesurés en temps réel est noté comme suit,

$$\boldsymbol{\xi}_{X_j}(t) = (\xi_{X_j,1}(t), \dots, \xi_{X_j,i}(t), \dots, \xi_{X_j,N_A}(t)), \quad j \in \{1, \dots, N_X\}, \quad (\text{A.34})$$

où $\xi_{X_j,i}(t)$ signifie la puissance à la réception du signal émis par l'ancre i et reçu par le nœud j à l'instant t .

Puis, afin de localiser le nœud j , on cherche un pattern matching dans la base de données des fingerprints, en s'appuyant sur l'algorithme des K plus proches voisins [Navarro *et al.* 2010; Chuenurajit *et al.* 2013]. Cet algorithme calcule ainsi pour chaque nœud j les distances euclidiennes $\|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_k}\|$ entre son vecteur $\boldsymbol{\xi}_{X_j}(t)$ obtenu à l'instant t et tous les vecteurs de RSSI de référence $\boldsymbol{\xi}_{P_k}$, $k \in \{1, \dots, N_P\}$. Ensuite, les K plus petites distances sont sélectionnées. Soit $I_j(t)$ l'ensemble des indices des positions de référence produisant ces K plus petites distances de puissance. Par conséquent, l'estimation de

la position du nœud j par fingerprinting est effectuée comme suit,

$$\hat{\mathbf{x}}_{f,j}(t) = \sum_{k \in I_j(t)} \omega_k \mathbf{p}_k, \quad (\text{A.35})$$

où ω_k est le facteur de poids associé à la position de référence \mathbf{p}_k choisie. Le poids ω_k , $k \in I_j(t)$, est choisi inversement proportionnel à la distance euclidienne des vecteurs de RSSIs comme suit,

$$\omega_k = \frac{\|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_k}\|^{-\alpha}}{\sum_{u \in I_j(t)} \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_u}\|^{-\alpha}}, \quad (\text{A.36})$$

ou sous forme exponentielle,

$$\omega_k = \frac{\exp(-\alpha \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_k}\|)}{\sum_{u \in I_j(t)} \exp(-\alpha \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P_u}\|)}, \quad (\text{A.37})$$

où α est un paramètre déterminant du facteur de poids qui pourrait être choisi de manière empirique.

A.3.1.2 Localisation à base d'accélération

Chaque nœud j est supposé être équipé d'un accéléromètre [Hsu et Yu 2009], donnant, à chaque instant t , les accélérations du nœud selon les différentes coordonnées, notées par $\boldsymbol{\gamma}_j(t) = (\gamma_{j,1}(t), \dots, \gamma_{j,D}(t))$. Basé uniquement sur ses données inertielles, le nœud j calcule la deuxième estimation de sa position selon le système d'équations suivant,

$$\begin{aligned} \hat{\mathbf{x}}_{\gamma,j}(t) &= \hat{\mathbf{x}}_{\gamma,j}(t-1) + \boldsymbol{\nu}_j(t-1)\Delta t + \frac{1}{2}\boldsymbol{\gamma}_j(t)\Delta t^2, \\ \boldsymbol{\nu}_j(t) &= \boldsymbol{\nu}_j(t-1) + \boldsymbol{\gamma}_j(t)\Delta t, \end{aligned} \quad (\text{A.38})$$

où $\boldsymbol{\nu}_j(t)$ est la vitesse estimée du nœud j à l'instant t . Notez que ce modèle d'ordre deux suppose que les accélérations sont constantes entre deux instants consécutifs. Si les courbes des accélérations varient légèrement, ce système conduit à des estimations précises. Cependant, avec des changements brusques d'accélération ou dans le cas des accélérations plus interrompues, les estimations $\hat{\mathbf{x}}_{\gamma,j}(t)$ pourraient être sensiblement déviées de la trajectoire vraie en raison des erreurs cumulatives du modèle.

Alternativement, un modèle simplifié en premier ordre est proposé pour calculer les estimations basées sur les accélérations,

$$\begin{aligned} \hat{\mathbf{x}}_{\gamma,j}(t) &= \hat{\mathbf{x}}_{\gamma,j}(t-1) + \boldsymbol{\nu}_j(t)\Delta t, \\ \boldsymbol{\nu}_j(t) &= \boldsymbol{\nu}_j(t-1) + \boldsymbol{\gamma}_j(t)\Delta t. \end{aligned} \quad (\text{A.39})$$

Dans ce modèle, les accélérations sont d'abord supposées constantes entre $t-1$ et t pour le calcul des vitesses à l'instant t . Ensuite, une approximation supplémentaire est effectuée en supposant que les vitesses sont constantes entre $t-1$ et t . De toute évidence, ce modèle est moins précis que le modèle d'ordre deux.

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

Ayant les accélérations aux instants $t-1$ et t , on pourrait proposer un troisième modèle afin de mieux prendre la variation des accélérations en considération par

$$\begin{aligned}\hat{\mathbf{x}}_{\gamma,j}(t) &= \hat{\mathbf{x}}_{\gamma,j}(t-1) + \boldsymbol{\nu}_j(t-1)\Delta t + \frac{1}{2}\boldsymbol{\gamma}_j(t-1)\Delta t^2 + \frac{1}{6}\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}\Delta t^3, \\ \boldsymbol{\nu}_j(t) &= \boldsymbol{\nu}_j(t-1) + \boldsymbol{\gamma}_j(t-1)\Delta t + \frac{1}{2}\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}\Delta t^2.\end{aligned}\tag{A.40}$$

Tel modèle est plus approprié aux trajectoires où les accélérations mesurées ont des changements brusques. Il est à noter que ce modèle suppose que les accélérations sont linéaires ente $t-1$ et t , avec une pente de $\frac{\boldsymbol{\gamma}_j(t) - \boldsymbol{\gamma}_j(t-1)}{\Delta t}$. Sous une petite valeur de Δt , ce modèle fonctionne très bien, puisque les courbes des accélérations approximées deviennent très proches de celles vraies.

Dans la suite, la combinaison des deux estimations sera effectuée à l'aide du calcul par intervalles en introduisant l'incertitude sur les mesures de fingerprints et des accélérations.

A.3.2 L'algorithme

Quand un nœud se déplace à travers le réseau, il recueille les informations de RSSIs et mesure les valeurs de ses accélérations instantanées en même temps. Afin de combiner l'ensemble de ces informations, nous appliquons l'analyse par intervalles [Jaulin *et al.* 2001a; Mourad *et al.* 2009a], en introduisant des incertitudes par les mesures, conduisant ainsi à des estimées en forme de boîtes incluant les vraies positions.

A.3.2.1 Reformulation des accélérations

Soit $\pm\delta_\gamma$ l'incertitude présente sur les accélérations mesurées. Il est possible dans ce cas de définir des boîtes d'accélérations de D -dimensions comme suit,

$$[\boldsymbol{\gamma}_j](t) = [\boldsymbol{\gamma}_{j,1}](t) \times \cdots \times [\boldsymbol{\gamma}_{j,D}](t) = [\boldsymbol{\gamma}_{j,1}(t) - \delta_\gamma, \boldsymbol{\gamma}_{j,1}(t) + \delta_\gamma] \times \cdots \times [\boldsymbol{\gamma}_{j,D}(t) - \delta_\gamma, \boldsymbol{\gamma}_{j,D}(t) + \delta_\gamma].\tag{A.41}$$

De cette manière, il est possible de réécrire les modèles de mobilité mentionnés ci-dessus en termes d'intervalles. Par exemple, celui du deuxième ordre est représenté comme suit,

$$\begin{aligned}[\hat{\mathbf{x}}_j]_\gamma(t) &= [\hat{\mathbf{x}}_j](t-1) + [\boldsymbol{\nu}_j](t-1)\Delta t + \frac{1}{2}[\boldsymbol{\gamma}_j](t)\Delta t^2, \\ [\boldsymbol{\nu}_j](t) &= [\boldsymbol{\nu}_j](t-1) + [\boldsymbol{\gamma}_j](t)\Delta t.\end{aligned}\tag{A.42}$$

La quantité δ_γ est soit considérée comme l'incertitude de mesure définie par le fabricant de l'instrument, ou déterminée par une procédure de calibrage de l'accéléromètre, comme le montre la remarque 1 dans le chapitre III. Considérez que $\varepsilon_{\gamma,j,\ell}(t)$ est un bruit gaussien de moyenne nulle avec un écart-type σ_γ . Par conséquent, l'incertitude δ_γ pourraient être prises pour $3\sigma_\gamma$ pour garantir une probabilité de 99,7% pour que l'accélération réelle soit incluse dans l'intervalle considéré. On pourrait également prendre δ_γ à la valeur de l'erreur maximale d'accélération enregistrée lors du calibrage. Une autre formulation de l'estimation par intervalle est introduite dans la remarque 3 du

chapitre III. Nous montrons également dans la remarque 2 de ce même chapitre comment l'on pourrait modifier la vitesse estimée pour réduire son incertitude en cours de localisation.

A.3.2.2 Reformulation des fingerprints

Dans le but de ré-écrire les estimations des positions par fingerprints, une incertitude $\pm\delta_\xi$ est introduite aux valeurs des RSSIs mesurées. En introduisant δ_ξ , nous obtenons $[\xi_{X_j}](t)$, le vecteur RSSI en boîte pour le nœud j à l'instant t , et $[\xi_{P_k}]$, celui associé à la position de référence \mathbf{p}_k , dont les éléments sont récrit par ce qui suit,

$$\begin{aligned} [\xi_{X_{j,i}}](t) &= [\xi_{X_{j,i}}(t) - \delta_\xi, \xi_{X_{j,i}}(t) + \delta_\xi], \\ &et, \\ [\xi_{P_{k,i}}] &= [\xi_{P_{k,i}} - \delta_\xi, \xi_{P_{k,i}} + \delta_\xi]. \end{aligned} \tag{A.43}$$

Cette quantité δ_ξ peut être déterminée comme la taille maximale de l'intervalle comprenant toutes les valeurs de RSSIs mesurées étant donné une distance d . Cette quantité est obtenue en effectuant plusieurs mesures dans la zone de surveillance a priori du processus de localisation.

L'estimation de positions à base de fingerprints est ainsi reformulée en boîte comme suit,

$$[\hat{\mathbf{x}}_j]_f(t) = \sum_{k \in I_j(t)} [\omega_k] \mathbf{p}_k, \tag{A.44}$$

où $[\omega_k]$ est le poids en termes d'intervalles associé à la position de référence \mathbf{p}_k , comme suit,

$$\begin{aligned} [\omega_k] &= \frac{\|[\xi_{X_j}](t), [\xi_{P_k}]\|^{-\alpha}}{\sum_{u \in I_j(t)} \|[\xi_{X_j}](t), [\xi_{P_u}]\|^{-\alpha}}, \\ &ou, \\ [\omega_k] &= \frac{\exp(-\alpha \|[\xi_{X_j}](t), [\xi_{P_k}]\|)}{\sum_{u \in I_j(t)} \exp(-\alpha \|[\xi_{X_j}](t), [\xi_{P_u}]\|)}. \end{aligned} \tag{A.45}$$

Une autre façon d'obtenir la boîte $[\hat{\mathbf{x}}_j]_f(t)$ consiste à introduire une autre quantité d'incertitude $\pm\delta_{x_\xi}$ à l'estimée finale $\hat{\mathbf{x}}_{f,j}(t)$. La quantité δ_{x_ξ} pourrait être déterminée par l'exécution d'une procédure de localisation préalable sur des positions de test, et en prenant ensuite la valeur maximale d'erreur d'estimation. Par conséquence, la position estimée est reformulée en termes d'intervalles comme suit,

$$[\hat{\mathbf{x}}_j]_f(t) = [\hat{x}_{f,j,1}(t) - \delta_{x_\xi}, \hat{x}_{f,j,1}(t) + \delta_{x_\xi}] \times \cdots \times [\hat{x}_{f,j,D}(t) - \delta_{x_\xi}, \hat{x}_{f,j,D}(t) + \delta_{x_\xi}]. \tag{A.46}$$

A.3.2.3 Combinaison

La combinaison des deux estimées peut être effectuée en appliquant une intersection entre $[\hat{\mathbf{x}}_j]_f(t)$ et $[\hat{\mathbf{x}}_j]_\gamma(t)$ pour le nœud j à l'instant t . En effet,

$$[\hat{\mathbf{x}}_j](t) = [\hat{\mathbf{x}}_j]_f(t) \cap [\hat{\mathbf{x}}_j]_\gamma(t). \tag{A.47}$$

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

Le centre de $[\hat{\mathbf{x}}_j](t)$ donne l'estimation finale de la position du nœud j à l'instant t . Voir Figure III.2 du chapitre III pour illustration.

A.3.3 Analyse et simulations

La performance de la méthode proposée est analysée et illustrée dans une zone de surveillance s'étendant sur un plan $100m \times 100m$ à deux dimensions, où les ancres et les positions de référence sont déployées uniformément. Nous considérons un seul nœud mobile, en raison de simplicité. La trajectoire du nœud est générée en intégrant deux fois les fonctions d'accélération de chaque axe de \mathbb{R}^2 , où les signaux d'accélération sont d'abord générés sur la 1^{ère} et la 2^{ième} coordonnées variant sur une période de 100s avec un intervalle de temps $\Delta t = 1s$ en utilisant la fonction sinus, comme c'est montré dans Figure III.3 et Figure III.4. Les vitesses initiales sont réglées à zéro, ce qui signifie que le nœud est supposé fixe, avec une position connue au début de la localisation. Veuillez consulter les figures et tableaux présentés dans le chapitre III pour toutes les simulations.

Les valeurs de RSSIs sont simulées par le modèle de perte du canal [Medeisis et Kajackas 2000a] exprimé ci-dessous,

$$\begin{aligned}\xi_{P k,i} &= \xi_0 - 10n_P \log_{10} \|\mathbf{a}_i, \mathbf{p}_k\| + \varepsilon_{\xi,k,i}, \\ \xi_{X i}(t) &= \xi_0 - 10n_P \log_{10} \|\mathbf{a}_i, \mathbf{x}(t)\| + \varepsilon_{\xi,i}(t),\end{aligned}\tag{A.48}$$

où ξ_0 est la puissance d'émission à une distance d' $1m$, prise à $100dBm$, n_P est le paramètre du modèle de valeur 4 et $\varepsilon_{\xi,k,i}$ et $\varepsilon_{\xi,i}(t)$ sont des bruits gaussiens à moyenne nulle avec un écart-type σ_ξ . De la même manière, des bruits gaussiens à moyenne nulle $\varepsilon_{\gamma,\ell}(t)$ sont ajoutés aux valeurs d'accélération $\tilde{\gamma}_\ell(t)$, avec un écart-type noté par σ_γ ,

$$\gamma_\ell(t) = \tilde{\gamma}_\ell(t) + \varepsilon_{\gamma,\ell}(t), \quad \ell \in \{1, 2\},\tag{A.49}$$

A.3.3.1 Illustration générale

En utilisant la trajectoire de la Figure III.4, nous réglons le nombre d'ancre N_A à 49, le nombre de points de référence N_P à 196, le nombre de voisins K à 3 et les écarts-types de bruits σ_γ et σ_ξ à $0.004m.s^{-2}$ et $0.1dB$ respectivement. Table III.1 montre les erreurs d'estimations en mètres en utilisant les fingerprints seuls, les accélérations seules avec les trois modèles de mobilité et la combinaison. Figure III.5 montre en rouge les boîtes estimées obtenues par notre méthode, en bleu celles obtenues par fingerprinting, en vert celles obtenues à partir des accélérations. Les résultats montrent que la méthode combinée, avec le modèle de mobilité d'ordre trois, donne les meilleures estimations. D'où l'on appliquera le modèle trois pour la suite.

A.3.3.2 Impact des paramètres

Les performances de notre méthode varient avec les différents paramètres, tels que K , N_A , N_P , ainsi que le coefficient α des poids et les écarts-types des bruits σ_ξ et σ_γ associés aux RSSIs et accélérations. Pour montrer ceci, nous effectuons les simulations en

modifiant à chaque fois l'un des paramètres et en laissant les autres fixes. Les résultats sont illustrés dans Figure III.8 - Figure III.14 pour la trajectoire de Figure III.7. Nous observons que lorsque K est supérieur à 2, les erreurs d'estimation sont légèrement variables, donc on peut fixer $K = 3$ pour les simulations restantes. Pour N_A et N_P , l'erreur d'estimation diminue généralement avec l'augmentation du nombre des ancrs ou des positions de référence. Par contre, l'erreur d'estimation augmente avec l'augmentation du bruit des RSSIs σ_ξ ou des accélérations σ_γ . De plus, nous considérons toujours que les positions de référence sont distribuées uniformément vu que la méthode fonctionne mieux qu'avec une distribution aléatoire selon nos simulations. Finalement, nous envisageons aussi les effets de l'expression du poids, ce qui nous mène à la formule d'inversion avec $\alpha = 2$ dans les simulations suivantes. En conclusion, ça mérite de noter que la performance de notre méthode combinée utilisant les intervalles est la meilleure, comparée aux méthodes utilisant seulement les fingerprints ou les accélérations dans tous les cas.

A.3.3.3 Comparaison avec une autre méthode

Notre méthode proposée est comparée par la suite avec une méthode basée sur les connectivités qui estime la position du nœud mobile en utilisant le filtre particulaire (Monte-Carlo). La solution est proposée dans l'intersection de tous les disques de rayon égal à la portée r et centrés sur les ancrs dont les RSSIs sont supérieures à un certain seuil ξ_r associé à la distance r [Baggio et Langendoen 2006a]. Pour cette raison, nous fixons $K = 3$, $N_A = 49$, $N_P = 196$, $\sigma_\gamma = 0.001m.s^{-2}$ et $\sigma_\xi = 0.1dBm$. Nous fixons également le nombre de particules à 100, et $r = 15$, menant à 3.33 ancrs détectées en moyenne et un seuil de puissance de $53dBm$ avec $\xi_0 = 100dBm$ et $n_P = 4$. Les simulations montrent que la méthode basée sur les connectivités donne une erreur d'estimation de $3.6986m$, comparée à $0.8216m$ pour notre méthode combinée, tout en consommant $1.167ms$ en temps de calcul moyen comparé à $0.252ms$ pour notre méthode.

A.4 Méthode de localisation clusterisée

La nouvelle génération de réseaux de capteurs sans fil tend à être plus flexible, robuste et de puissance, particulièrement pour des applications basées sur les réseaux de grandes échelles. Techniquement, l'architecture centralisée considère seulement un centre de fusion qui a le rôle de traiter toutes les informations dans le réseau [Tharmarasa *et al.* 2011]. Dans ce cas, le réseau consomme plus d'énergie vu que toutes les informations devraient être acheminées vers le centre de fusion et puis relayées de nouveau vers le réseau dès que l'analyse et le traitement nécessaire sont réalisés. En introduisant les agents individuels, qui pourraient être des capteurs intelligents dans les réseaux, l'information pourrait être analysée et traitée à proximité, réduisant ainsi la consommation de communication. En outre, l'architecture clusterisée est généralement plus efficace et plus robuste à l'échec que celle centralisée tenant compte de plusieurs points de collection et de calcul [Tharmarasa *et al.* 2011; Ferrari et Pagliari 2006].

Dans les réseaux clusterisés, la zone de surveillance est construite de plusieurs petits clus-

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

ters en fonction de l'emplacement des agents individuels. Alors l'algorithme clusterisé offre des estimations locales en parallélisme correspondant à chaque cluster, combinées par la suite pour l'obtention d'une estimation globale. Cette section discutera du problème de localisation clusterisée utilisant la technique des K plus proches voisins [Chen et Chang 2008; Aldhubaib et Shuley 2010] ainsi que les données inertielles [Schuler *et al.* 1967; Lee et Huang 2002], et cherchant les solutions à l'aide de la théorie des intervalles [Moore 1966; Jaulin *et al.* 2001b; Mourad *et al.* 2009b] et du filtre de Kalman [Kalman 1960; Chan *et al.* 1979; Honghui et Moore 2002].

A.4.1 Configuration du réseau

La méthode proposée consiste à diviser la zone entière en plusieurs clusters. Ceci peut être réalisé en s'appuyant sur la configuration des obstacles dans la zone de surveillance, ou plutôt en considérant des clusters de dimensions égales. Nous considérons que le réseau est composé de N_A ancras et N_X nœuds mobiles, dénotés respectivement par $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,D})$, $i = 1, \dots, N_A$, et $\mathbf{x}_j(t) = (x_{j,1}(t), \dots, x_{j,D}(t))$, $j = 1, \dots, N_X$, t étant l'itération actuelle et D la dimension de la zone ($D = 2$ ou 3). Une illustration est fournie dans Figure IV.1 avec 4 clusters égaux, chacun équipé d'une tête de cluster ou calculateur, pouvant être un ordinateur ou même un capteur ayant la capacité de réaliser des calculs.

Supposons qu'il existe N_C clusters dans le réseau, et donc N_C têtes de cluster dont les coordonnées sont dénotées par $\mathbf{h}_{(c)} = (h_{(c),1}, \dots, h_{(c),D})$, $c = 1, \dots, N_C$. La méthode étant basée sur les fingerprints, $N_{P(c)}$ positions de référence sont uniformément distribuées dans chaque cluster $c = 1, \dots, N_C$; leurs coordonnées sont dénotées par $\mathbf{p}_{(c),n} = (p_{(c),n,1}, \dots, p_{(c),n,D})$, $n = 1, \dots, N_{P(c)}$. Les ancras émettent ensuite des signaux dans le réseau avec la même puissance initiale. De ce fait, on mesure les RSSIs aux positions de référence notées par

$$\boldsymbol{\xi}_{P(c),n} = (\xi_{P(c),n,1}, \dots, \xi_{P(c),n,N_A}), \quad (\text{A.50})$$

où $\xi_{P(c),n,i}$ est la puissance émise par l'ancre i et reçue à la position de référence $\mathbf{p}_{(c),n}$ du cluster c . Telle configuration conduit alors à une base de données de fingerprints composée de N_C bases locales, ayant chacune $N_{P(c)}$ couples comprenant les positions de référence avec leurs vecteurs de RSSIs $(\mathbf{p}_{(c),n}, \boldsymbol{\xi}_{P(c),n})$, $n = 1, \dots, N_{P(c)}$ et $c = 1, \dots, N_C$. Chaque base locale de fingerprints est stockée dans la tête de cluster correspondant, pour être utilisée après avec les données en ligne afin d'effectuer la localisation.

A.4.2 L'algorithme clusterisé

Une fois que la configuration clusterisée est effectuée, l'algorithme procède à l'estimation utilisant à la fois les fingerprints et les lectures inertielles. Dans ce qui suit, le problème est résolu à l'aide de la théorie des intervalles et du filtre de Kalman.

A.4.2.1 Estimation par fingerprints

Un nœud j en mouvement collecte les signaux émis par les ancrs et mesure leurs RSSIs, obtenant un vecteur de RSSIs,

$$\boldsymbol{\xi}_{X_j}(t) = (\xi_{X_{j,1}}(t), \dots, \xi_{X_{j,N_A}}(t)), \quad j \in \{1, \dots, N_X\}. \quad (\text{A.51})$$

Il envoie par la suite son vecteur de RSSIs aux têtes de cluster dans sa portée de communication. Soit $I_{C_j}(t)$ l'ensemble des indices de toutes les têtes de cluster recevant les RSSIs du nœud j à l'itération t . Afin de localiser le nœud j , chaque tête de cluster appartenant à $I_{C_j}(t)$ applique l'algorithme des K plus proches voisins. Les K positions de référence donnant les K plus petites distances $\|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P(c),n}\|$ sont sélectionnées. Soit $I_{P(c),j}(t)$, $c \in I_{C_j}(t)$, l'ensemble des indices de ces positions de référence. En conséquence, chaque tête de cluster calcule une estimée locale de la position du nœud j comme suit,

$$\hat{\boldsymbol{x}}_{(c),j}(t) = \sum_{n \in I_{P(c),j}(t)} \omega_{(c),j,n}(t) \boldsymbol{p}_{(c),n}, \quad c \in I_{C_j}(t), \quad (\text{A.52})$$

où $\omega_{(c),j,n}(t)$ est le poids associé à la position de référence $\boldsymbol{p}_{(c),n}$ pour l'estimation de la position du nœud j à l'instant t , donné en termes de distances inversées par

$$\omega_{(c),j,n}(t) = \frac{\|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P(c),n}\|^{-\alpha}}{\sum_{u \in I_{P(c),j}(t)} \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P(c),u}\|^{-\alpha}}, \quad (\text{A.53})$$

ou en termes de fonction exponentielle comme suit,

$$\omega_{(c),j,n}(t) = \frac{\exp\left(-\alpha \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P(c),n}\|\right)}{\sum_{u \in I_{P(c),j}(t)} \exp\left(-\alpha \|\boldsymbol{\xi}_{X_j}(t), \boldsymbol{\xi}_{P(c),u}\|\right)}. \quad (\text{A.54})$$

Les estimées locales $\hat{\boldsymbol{x}}_{(c),j}(t)$ sont ensuite combinées pour obtenir l'estimée finale en considérant les poids de chaque cluster. Les poids liés aux clusters, dénotés par $\lambda_{(c),j}(t)$, sont déterminés par les puissances $\xi_{H(c),j}(t)$ à la réception des signaux émis par le nœud j et reçus aux têtes de cluster c , comme suit,

$$\lambda_{(c),j}(t) = \frac{\exp\left(\xi_{H(c),j}(t)\right)}{\sum_{u \in I_{C_j}(t)} \exp\left(\xi_{H(u),j}(t)\right)}. \quad (\text{A.55})$$

D'où, la première estimée globale pour le nœud j à l'itération t est donnée par

$$\hat{\boldsymbol{x}}_j^-(t) = \sum_{c \in I_{C_j}(t)} \lambda_{(c),j}(t) \hat{\boldsymbol{x}}_{(c),j}(t), \quad j \in \{1, \dots, N_X\}. \quad (\text{A.56})$$

La combinaison se fait à l'une des têtes de cluster voisines au nœud.

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

A.4.2.2 Estimation par mobilité

La deuxième estimée peut être obtenue similairement à la méthode centralisée. Pourtant, nous souhaitons dans cette section l'étendre au cas de rotation du nœud. En d'autres termes, en utilisant les mesures d'accélération directement, la méthode suppose que le nœud ne peut pas tourner, et donc que le système de coordonnées du nœud reste le même que le système de coordonnées global du réseau. Cependant, les nœuds ont bien la possibilité de tourner en se déplaçant. Pour surmonter ce problème, les nœuds devraient être équipés de gyroscopes, qui donnent leurs orientations par rapport au système de coordonnées global. En $3D$, un nœud possède trois angles de rotations, $\vartheta_j(t)$, $\psi_j(t)$ and $\phi_j(t)$, notamment celui d'autour du troisième, du premier et du deuxième axe du système de coordonnées global respectivement, comme illustré dans Figure IV.2.

Soit $\gamma'_{j,1}$, $\gamma'_{j,2}$ et $\gamma'_{j,3}$ les accélérations mesurées au nœud j à l'instant t dans son système de coordonnées. Alors les accélérations $\gamma_{j,1}$, $\gamma_{j,2}$ et $\gamma_{j,3}$ dans le système de coordonnées global sont obtenues par

$$\begin{pmatrix} \gamma_{j,1} \\ \gamma_{j,2} \\ \gamma_{j,3} \end{pmatrix} = \mathfrak{R}(\vartheta_j, \psi_j, \phi_j) \begin{pmatrix} \gamma'_{j,1} \\ \gamma'_{j,2} \\ \gamma'_{j,3} \end{pmatrix}, \quad (\text{A.57})$$

où la matrice de rotation $3D$ est donnée par

$$\begin{pmatrix} \cos \vartheta_j \cos \phi_j & -\sin \vartheta_j \cos \phi_j & \sin \phi_j \\ \cos \vartheta_j \sin \psi_j \sin \phi_j + \sin \vartheta_j \cos \psi_j & -\sin \vartheta_j \sin \psi_j \sin \phi_j + \cos \vartheta_j \cos \psi_j & -\sin \psi_j \cos \phi_j \\ -\cos \vartheta_j \cos \psi_j \sin \phi_j + \sin \vartheta_j \sin \psi_j & \sin \vartheta_j \cos \psi_j \sin \phi_j + \cos \vartheta_j \sin \psi_j & \cos \psi_j \cos \phi_j \end{pmatrix}. \quad (\text{A.58})$$

En $2D$, la rotation se produit uniquement dans le plan avec l'angle de rotation ϑ_j étant donné que $\phi_j = \psi_j = 0$, ainsi les accélérations dans le repère global sont obtenues par

$$\begin{pmatrix} \gamma_{j,1} \\ \gamma_{j,2} \end{pmatrix} = \begin{pmatrix} \cos \vartheta_j & -\sin \vartheta_j \\ \sin \vartheta_j & \cos \vartheta_j \end{pmatrix} \begin{pmatrix} \gamma'_{j,1} \\ \gamma'_{j,2} \end{pmatrix}. \quad (\text{A.59})$$

Ainsi à chaque instant, le nœud mesure ses accélérations et ses orientations à l'aide de son accéléromètre et son gyroscope. Ses accélérations dans le repère global sont ensuite calculées pour être utilisées par la suite dans les équations de mobilité afin d'obtenir la deuxième estimée de position $\hat{\mathbf{x}}_j^+(t)$.

Dans ce qui suit, nous illustrons la façon de combiner les deux estimées, à l'aide de la théorie des intervalles ainsi que le filtre de Kalman.

A.4.2.3 Résolution par la théorie des intervalles

La solution par intervalles tient à calculer la première estimée de position utilisant les RSSIs en termes de boîte, à corriger ensuite par la deuxième estimée aussi en termes de boîte utilisant les informations d'inertie.

Soient $[\xi_{P(c),n}]$, $c = 1, \dots, N_C$ et $n = 1, \dots, N_{P(c)}$, les boîtes de puissance de référence définies similairement à la méthode centralisée à l'aide de l'incertitude $\pm \delta_\xi$. De la même

manière, $[\xi_{X_j}](t)$, $j = 1, \dots, N_X$, dénotent les boîtes de puissance en ligne. D'après les distances en termes intervalles $\|[\xi_{X_j}](t), [\xi_{P(c),n}]\|$, les K indices de positions de référence donnant les plus petits centres des intervalles sont sélectionnés afin de définir l'ensemble $I_{P(c),j}(t)$. Les estimées locales $[\hat{\mathbf{x}}_{(c),j}](t)$ sont redonnées par,

$$[\hat{\mathbf{x}}_{(c),j}](t) = \sum_{n \in I_{P(c),j}(t)} [\omega_{(c),j,n}](t) \mathbf{p}_{(c),n}, \quad c \in I_{C_j}(t), \quad (\text{A.60})$$

employant les poids $[\omega_{(c),j,n}](t)$ en termes d'intervalles. Alors, la première estimée globale en boîte pour le nœud j est donnée en comptant les poids en intervalles $[\lambda_{(c),j}](t)$ par

$$[\hat{\mathbf{x}}_j^-](t) = \sum_{c \in I_{C_j}(t)} [\lambda_{(c),j}](t) [\hat{\mathbf{x}}_{(c),j}](t), \quad j \in \{1, \dots, N_X\}. \quad (\text{A.61})$$

La deuxième estimée en boîte peut être obtenue en introduisant une incertitude $\pm \delta_\gamma$ aux accélérations. L'estimation $[\hat{\mathbf{x}}_j^+](t)$ peut être reformulée à l'aide de δ_γ en termes des intervalles de la même façon que dans la section A.3.

Ayant les deux estimées en termes d'intervalles pour le nœud j à l'itération t , on peut calculer l'estimée finale par l'intersection des deux boîtes à chaque instant comme ci-dessous,

$$[\hat{\mathbf{x}}_j](t) = [\hat{\mathbf{x}}_j^-](t) \cap [\hat{\mathbf{x}}_j^+](t). \quad (\text{A.62})$$

L'estimation de la position exacte est aussi donnée par le centre de la boîte estimée.

A.4.2.4 Résolution par le filtre de Kalman

La combinaison peut également être effectuée en utilisant le filtre de Kalman, qui permet une estimation linéaire sur la base du critère des moindres carrés [Kalman 1960]. L'état inconnu est ici défini par la position du nœud $\mathbf{x}_j(t)$ et sa vitesse $\boldsymbol{\nu}_j(t)$, en d'autres termes l'état $\mathbf{X}_j(t)$ est donné par $(\mathbf{x}_j(t)^T \boldsymbol{\nu}_j(t)^T)^T$. L'idée de base du filtre de Kalman est de prédire l'état inconnu en utilisant les équations de mobilité, employant les accélérations, puis de le corriger en utilisant les observations, définies ici par la 1^{ère} estimation globale obtenue en utilisant les fingerprints. Le problème d'après Kalman est alors donné par

$$\begin{aligned} \mathbf{X}_j(t) &= A \mathbf{X}_j(t-1) + \mathbf{u}_j(t) + \zeta_j(t), \\ \hat{\mathbf{x}}_j^-(t) &= C \mathbf{X}_j(t) + \varepsilon_j(t), \end{aligned} \quad (\text{A.63})$$

où $\zeta_j(t) \sim \mathcal{N}(0, V)$ est le bruit du modèle d'état ayant une distribution normale à moyenne nulle, avec la matrice de covariance V de taille $(2D \times 2D)$, et $\varepsilon_j(t) \sim \mathcal{N}(0, R)$ est le bruit de l'équation d'observation ayant une distribution normale à moyenne nulle, avec la matrice de covariance R de taille $(D \times D)$.

Dans notre contexte, nous définissons la matrice de transition d'état A et le vecteur d'entrée de commande bruitée $\mathbf{u}_j(t)$ selon les modèles de mobilité. La matrice d'observation C est une matrice de taille $(D * 2D)$ donnée par la matrice identité \mathbf{I}_D complétée par des zéros. Ayant défini les paramètres des équations du filtre de Kalman, nous pouvons alors procéder à la phase de prédiction en utilisant l'estimée précédente et l'équation d'état et à la phase de correction à l'aide de l'équation d'observation.

A.4.3 Analyse et simulations

Nous présentons ici les performances de la méthode proposée. Pour cette raison, nous considérons une zone de surveillance en $2D$ de dimensions $100m \times 100m$, divisée en N_C clusters qui sont tous égaux, avec les têtes de cluster à leurs centres. N_A ancrés et N_P positions de référence sont distribués régulièrement dans le réseau, produisant $\frac{N_P}{N_C}$ positions de référence dans chaque cluster. Les vecteurs RSSI sont également données par le modèle de perte du canal [Medeisis et Kajackas 2000b] comme suit,

$$\begin{aligned}\xi_{P(c),n,i} &= \xi_0 - 10n_P \log_{10} \|\mathbf{a}_i, \mathbf{p}_{(c),n}\|, \\ \xi_{X_i}(t) &= \xi_0 - 10n_P \log_{10} \|\mathbf{a}_i, \mathbf{x}(t)\|, \\ \xi_{H(c)}(t) &= \xi_0 - 10n_P \log_{10} \|\mathbf{h}_{(c)}, \mathbf{x}(t)\|.\end{aligned}\tag{A.64}$$

Des bruits gaussiens avec l'écart-type σ_ξ sont ajoutés à tous les RSSIs présentés ci-dessus. Des bruits gaussiens avec les écarts-types σ_γ et σ_θ sont également ajoutés respectivement aux valeurs d'accélération et aux angles en cas de rotation.

A.4.3.1 Illustration générale

Figure IV.7 montre les trajectoires estimées par intervalles et Kalman ainsi que leurs boîtes estimées dans le cas où $N_C = 4$, $N_A = 16$, $N_P = 100$, $K = 3$, $\alpha = 2$, $\sigma_\gamma = 0.01m/s^2$ et $\sigma_\xi = 1dBm$. Table IV.1 montre les erreurs d'estimation en mètres et le temps de calcul moyen en millisecondes par itération. Comme on s'y attendait, la méthode combinée conduit à des résultats plus précis au prix de l'augmentation du temps d'estimation, qui reste quand même très faible. Le tableau montre également que le modèle de troisième ordre effectue des meilleurs résultats que les autres. Dans ce qui suit, nous considérons les modèles du deuxième ordre et du troisième ordre dans la méthode proposée et celle basée uniquement sur les accélérations.

A.4.3.2 Impact des paramètres

Comme à l'analyse de la méthode centralisée, nous étudions également les impacts de tous les paramètres, K , N_C , N_A , N_P , α , σ_γ et σ_ξ en comparant les erreurs et le temps de calcul à la fois, entre les deux modèles de mobilité. Les résultats sont illustrés dans Figure IV.9 - Figure IV.14. En observant les résultats, nos deux approches dépassent les autres méthodes en particulier avec le modèle de mobilité du troisième ordre. Notons également que les simulations avec $K = 3$, $N_C = 4$ et $\alpha = 2$ donnent de meilleurs résultats; nous les fixons alors à ces valeurs pour les simulations suivantes. L'erreur d'estimation par rapport à N_A et N_P diminue avec l'augmentation du nombre, au détriment du temps de calcul qui augmente légèrement.

Par ailleurs, avec l'augmentation des écarts-types des bruits, les erreurs augmentent également. Nous explorons par la suite la performance par rapport aux différentes distributions des bruits comme indiqué dans Table IV.2. En particulier, nous considérons le cas de rotation en $2D$ avec l'écart-type du bruit d'angle de rotation σ_θ , les résultats étant indiqués dans Table IV.3. D'après ces résultats, la méthode donne de bons résultats même avec différentes distributions du bruit d'accélération et aussi dans le cas

de rotation. En outre, l'approche fondée sur Kalman est légèrement plus performante que celle basée sur les intervalles, en particulier avec le modèle de mobilité de troisième ordre.

A.4.3.3 Comparaison à d'autres méthodes

Il existe une variété de méthodes de localisation proposées dans la littérature. Dans Table IV.4 et Table IV.5, nous effectuons une comparaison de notre méthode à des méthodes basées sur l'estimation de la distance inter-capteurs, telles que les algorithmes appliquant la latération, le min-max [Priygharm et Chemtanomwong 2011] ou le filtre particule [Farmani *et al.* 2012] en variant le bruit σ_ξ et le nombre d'ancres N_A respectivement. Évidemment, notre méthode dépasse la précision des autres méthodes. Nous procédons ensuite à une comparaison à une méthode basée sur les connectivités utilisant l'approche de Monte-Carlo [Baggio et Langendoen 2006b]. Pour cette raison, fixons $N_A = 49$, $N_P = 196$, $\sigma_\gamma = 0.001m/s^2$ et $\sigma_\xi = 0.1dBm$. Du côté de Monte-Carlo, nous prenons le nombre de particules $N = 100$ et la portée de détection $r = 15m$. La méthode basée sur les connectivités donne une erreur d'estimation de $3.7989m$, à comparer à $1.0554m$ et $0.9031m$ avec notre méthode proposée par intervalles et Kalman respectivement. Le temps de calcul passe de $2.167ms$, à $0.1609ms$ et $0.1547ms$ avec notre méthode. Figure IV.15 montre les erreurs d'estimation relatives, ainsi que le temps de calcul, de notre méthode à base de Kalman sur celle basée sur les connectivités par rapport au nombre d'ancres dans le réseau avec $N_P = 100$ (le seul paramètre qui change).

Enfin, la méthode proposée est comparée à des méthodes utilisant les fingerprints, tels que les algorithmes du plus proche voisin (NN), des K plus proches voisins (KNN) [Navarro *et al.* 2010; Chuenurajit *et al.* 2013], Fuzzy K voisins les plus proches (FKNN) [Rozyyev *et al.* 2012], Fuzzy Clusters (FC) [Liu *et al.* 2012], Fuzzy C-Means clustering (FCM) [Suroso *et al.* 2011], ainsi que les méthodes basées sur les réseaux de neurones (NNet) [Gogolak *et al.* 2011], les Support Vector Machines (SVM) [Farjow *et al.* 2011] ou la Ridge Regression (RR) [Mahfouz *et al.* 2013]. Table IV.6, Table IV.7 et Table IV.8 montrent les erreurs d'estimation pour ces diverses méthodes en variant l'écart-type du bruit RSSI σ_ξ , le nombre d'ancres N_A et le nombre de points de référence N_P respectivement. Parmi ces nombreuses méthodes, nos méthodes proposées sont considérées généralement supérieures que les autres en terme de précision d'estimation de localisation.

A.5 Extension au zonage

Dans les applications de scénarios indoors, on cherche parfois à déterminer la pièce ou la zone où un objet réside, au lieu de connaître sa position précise. Ce type de localisation s'appelle zonage, dit aussi localisation au niveau de la pièce. Le zonage sert aux tâches de surveillance de l'habitat, les détecteurs & rapports des constructions intelligentes et au suivi de cibles [Stojmenovic 2005; Schiller et Voisard 2004; Krisp 2013]. Beaucoup de

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

travaux se dirigent vers la localisation par zonage, nous en citons les méthodes basées sur un trigger d'entrée [Noury *et al.* 2002; Lo *et al.* 2012], les méthodes basées sur les RSSIs et la compensation path-loss, ou le chemin restreint [D'hoë *et al.* 2008; Chiu *et al.* 2009], ainsi que les méthodes très connues basées sur les probabilités et la classification par SVM [Castro *et al.* 2001; Hotta *et al.* 2012].

Dans cette section, nous proposons une approche originale effectuant du zonage, basée sur la technique de la reconnaissance de zone basée sur les fingerprints, ainsi que l'information de mobilité. Le problème d'estimation est résolu à l'aide des intervalles et de la théorie des fonctions de croyance, qui nous permet de désigner des facteurs de confiance à toutes les solutions possibles.

A.5.1 Théorie des fonctions de croyance

Les facteurs de confiance sont introduits par la *théorie des fonctions de croyance* (TFC, ou the Belief Functions Theory, BFT), dite aussi *la théorie de Dempster-Shafer* (the Dempster-Shafer theory) [Shafer *et al.* 1976; Dempster 1967]. Celle-ci est complétée par un cadre connu sous le nom du *Modèle des croyances transférables* (Transferable Belief Model, TBM) [Smets et Kennes 1994], considéré comme un outil mathématique effectuant la fusion de données issues de sources différentes telle que la fusion d'informations multi-capteurs [Denoeux 1997; Lefevre *et al.* 2002; Ristic et Smets 2005]. Dans notre application, elle nous permet de combiner toutes les possibilités en considérant les confiances associées aux zones estimées des nœuds.

A.5.1.1 Description de la théorie

Considérons un système quelconque dont on cherche à déterminer l'état x défini dans le domaine $\Omega = \{\omega_1, \dots, \omega_n\}$. Ω est connu sous le nom de cadre de discernement avec $\{\omega_i\}_{i=1}^n$ discrètes. L'ensemble des parties de Ω est noté par,

$$2^\Omega = \{\emptyset, \{\omega_1\}, \dots, \Omega\}. \quad (\text{A.65})$$

Toutes les mesures concernant l'état x sont considérées comme des sources d'informations dans la TFC. Ces sources offrant des confiances peuvent être représentées de différentes manières dans le cadre du TBM, dont la plus utilisée est la Fonction de Masse de Croyance (FM), notée généralement m_S^Ω , et donnée par

$$\begin{aligned} m_S^\Omega : 2^\Omega &\rightarrow [0, 1], \quad A \mapsto m_S^\Omega(A), \quad \text{satisfaisant,} \\ \sum_{A \in 2^\Omega} m_S^\Omega(A) &= 1, \end{aligned} \quad (\text{A.66})$$

où S représente la source d'information dont est issue m_S^Ω . La masse $m_S^\Omega(A)$, correspond à la croyance accordée par la source S à l'hypothèse selon laquelle l'état x existe dans A . Tous les sous-ensembles A de Ω satisfaisant $m_S^\Omega(A) > 0$ sont appelés ensembles focaux. Notez que, dans le cas de $\Omega = \mathbb{R}$, ou $\Omega = \mathbb{R}^n$ les éléments focaux seront en termes des intervalles, ou des pavés de n dimensions [Moore et Moore 1979; Jaulin *et al.* 2001a].

Une fois les informations issues de différentes sources sont disponibles, on peut les intégrer par des opérateurs de combinaison permettant d'obtenir une information plus fiable et concentrée sur l'état recherché [Smets 1993]. Soient m_1 and m_2 deux FM's issues de deux sources distinctes et fiables S_1 and S_2 . Une règle de combinaison conjonctive, notée CRC (Conjunctive rule of combination) est définie comme suit,

$$m_{1\ominus 2}(A) = (m_1 \ominus m_2)(A) = \sum_{i,j|B_i \cap C_j = A} m_1(B_i) m_2(C_j), \quad \forall A \subseteq \Omega. \quad (\text{A.67})$$

Notez que la règle de CRC est commutative et associative. Si les sources d'informations sont conflictuelles, ils peuvent en résulter des intersections vides. La définition de $m_{1\ominus 2}(\cdot)$ peut être suivie par une procédure de normalisation afin d'obtenir une FM dont la somme est égale à un.

En outre, la règle de combinaison disjonctive (DRC, Disjunctive rules of combination) est alternativement utilisée pour combiner deux FM's distinctes dans le cas où au moins une est fiable. La combinaison des deux FM's m_1 et m_2 selon la DRC est donnée par

$$m_{1\cup 2}(A) = (m_1 \cup m_2)(A) = \sum_{i,j|B_i \cup C_j = A} m_1(B_i) m_2(C_j), \quad \forall A \subseteq \Omega. \quad (\text{A.68})$$

Notez que la FM obtenue par CRC est plus informative que m_1 et m_2 en raison que la règle transfère les masses des éléments focaux vers des ensembles de cardinalité plus petite. Par contre, la FM obtenue de la DRC est moins informative en raison d'un effet de généralisation.

Au sujet de la prise de décision, plusieurs règles de décision sont utilisées, telles que le maximum de masse de croyance, le maximum de plausibilité, le maximum de probabilité pignistique, etc [Denoëux 1997]. Citons un exemple de la probabilité pignistique notée par $BetP$, qui est une distribution de probabilité calculée à partir de m^Ω en utilisant la transformation pignistique [Smets et Kennes 1994]. Soit m^Ω une FM associée à Ω et A_1, \dots, A_r les ensembles focaux de m^Ω . La transformation pignistique consiste à répartir uniformément la masse des ensembles $A_i, i \in 1, \dots, r$ comme suit,

$$BetP(\omega_k) = \sum_{i|A_i \subseteq \Omega, \omega_k \in A_i} \frac{m^\Omega(A_i)}{|A_i|}, \quad (\text{A.69})$$

où $|A_i|$ dénote le cardinal de A_i . Dans le cas où $\Omega = \mathbb{R}$, les éléments focaux sont en termes des intervalles $[x]_1, \dots, [x]_r$, alors la probabilité pignistique est redonnée par la fonction de densité comme suit,

$$BetP(x) = \sum_{i=1}^r \mathbb{I}_{[x]_i}(x) \frac{m^\Omega([x]_i)}{|[x]_i|}, \quad (\text{A.70})$$

où $\mathbb{I}_{[x]_i}(x)$, la fonction indicatrice, explicite l'affiliation de x à $[x]_i$ avec $\mathbb{I}_{[x]_i}(x) = 1$ ou non avec $\mathbb{I}_{[x]_i}(x) = 0$. $|[x]_i|$ est la largeur de l'intervalle $[x]_i$.

A.5.2 Algorithme de zonage

Dans le cadre du zonage, nous collectons les informations de fingerprints ainsi que les informations d'inertie dans le réseau composé de zones indépendantes, telles que des bureaux, des salles de classes, des ateliers ou des couloirs, l'objet à localiser se déplaçant librement à travers toutes les zones. Les informations sont utilisées pour adresser le problème du zonage à l'aide de la théorie des intervalles [Jaulin *et al.* 2001a] et la théorie des fonctions de croyance [Smets et Kennes 1994].

A.5.2.1 Configuration du réseau

La méthode proposée est une technique centralisée basée sur les mesures de RSSIs des signaux émis par les ancres. Considérons une région de surveillance de \mathbb{R}^D composée de N_Z zones indépendantes ayant toute les mêmes dimensions. Les zones sont dénotées par un attribut de D éléments (z_1, \dots, z_D) , où $z_d \in \{1, \dots, n_{Z_d}\}$ désigne le numéro de la zone projetée sur la direction d , $d \in \{1, \dots, D\}$, avec n_{Z_d} la valeur maximale de z_d et $N_Z = \prod_{d=1}^D n_{Z_d}$. Comme la méthode centralisée de localisation, nous considérons dans le réseau les ancres $\mathbf{a}_i = (a_{i,1}, \dots, a_{i,D})$, $i \in \{1, \dots, N_A\}$, avec les nœuds mobiles $\mathbf{x}_j(t) = (\mathbf{x}_{j,1}(t), \dots, \mathbf{x}_{j,D}(t))$, $j \in \{1, \dots, N_X\}$. Pour déterminer l'attribut $\mathbf{z}_{\mathbf{x}_j}(t) = (z_{x_{j,1}}(t), \dots, z_{x_{j,D}}(t))$ de la zone où le nœud j se trouve à l'instant t , nous effectuons la collection des données de fingerprints par avance en introduisant les N_P positions de référence $\mathbf{p}_k = (p_{k,1}, \dots, p_{k,D})$, $k \in \{1, \dots, N_P\}$, avec leurs attributs de zones $\mathbf{z}_{\mathbf{p}_k} = (z_{p_{k,1}}, \dots, z_{p_{k,D}})$, $k \in 1, \dots, N_P$. Un exemple du réseau de 2 dimensions ayant neuf zones est illustré dans Figure V.2, où $n_{Z_1} = n_{Z_2} = 3$.

Les ancres émettent par la suite des signaux radios dans le réseau avec la même puissance initiale. La base de données des RSSIs est ainsi construite en mesurant les RSSIs $\xi_{P_k,i}$ à la réception aux positions \mathbf{p}_k , $k \in \{1, \dots, N_P\}$, des signaux émis par les ancres \mathbf{a}_i , $i \in \{1, \dots, N_A\}$. Notons le vecteur de RSSIs ξ_{P_k} , de taille N_A , collectées à la position \mathbf{p}_k , alors

$$\xi_{P_k} = (\xi_{P_k,1}, \dots, \xi_{P_k,N_A}), \quad k \in \{1, \dots, N_P\}. \quad (\text{A.71})$$

En conséquence, la base de données de fingerprints est composée de N_P couples $(\xi_{P_k}, \mathbf{z}_{\mathbf{p}_k})$, où les attributs de zones des points de référence sont considérés et non pas leurs positions exactes. Ceci facilite la construction de la base de données, à comparer avec les méthodes précédentes.

A.5.2.2 Zonage par fingerprints

Comme la méthode centralisée, lorsque nœud j se déplace à travers le réseau, il mesure les RSSIs des signaux reçus des ancres, désignés comme suit,

$$\xi_{X_j}(t) = (\xi_{X_j,1}(t), \dots, \xi_{X_j,N_A}(t)), \quad j \in \{1, \dots, N_X\}. \quad (\text{A.72})$$

Les K points de référence fournissant les K plus petites distances de puissance $\|\xi_{X_j}(t), \xi_{P_k}\|$, $k \in 1, \dots, N_P$, sont sélectionnées. Soit $I_j(t)$ l'ensemble de leurs indices. L'estimation de

zone est par la suite calculée en utilisant les attributs de zones des positions sélectionnées comme suit,

$$\hat{z}_{x_j,d}(t) = \sum_{k \in I_j(t)} \omega_{j,k}(t) z_{p_{k,d}}, \quad d \in \{1, \dots, D\}, \quad (\text{A.73})$$

où $\omega_{j,k}(t)$ est le poids calculé par la version à l'inverse des distances euclidienne donnée dans le chapitre III, en prenant $\alpha = 2$.

Les quantités $\hat{z}_{x_j,d}(t)$, $d \in \{1, \dots, D\}$, sont les estimées des coordonnées de zones selon les directions d de la région de surveillance de D dimensions. Cependant, $\hat{z}_{x_j,d}(t)$ obtenue n'est pas forcément un nombre entier. Pour cette raison, nous introduisons $\underline{z}_{x_j,d}(t)$ et $\bar{z}_{x_j,d}(t)$ qui représentent les entiers respectivement précédant et suivant $\hat{z}_{x_j,d}(t)$. Alors $\tilde{z}_{x_j,d}(t)$, l'estimée entière de la vraie coordonnée de zone $z_{x_j,d}(t)$ du nœud j à l'instant t est redonnée par deux valeurs $\tilde{z}_{x_j,d}(t) \in \{\underline{z}_{x_j,d}(t), \bar{z}_{x_j,d}(t)\}$. La combinaison des différentes valeurs de $\tilde{z}_{x_j,d}(t)$ fournit 2^D attributs estimés pour la vraie zone du nœud. Basée sur la TFC, l'approche proposée consiste alors à supposer le calcul par fingerprinting comme une source d'information, donnant une fonction de masse $m_{f_j,t}(\cdot)$ ayant 2^D attributs ou éléments focaux. Soit $\tilde{\mathbf{x}}_j(t) = (\tilde{z}_{x_{j,1}}(t), \dots, \tilde{z}_{x_{j,D}}(t))$ l'un des estimés, alors $m_{f_j,t}(\cdot)$ est définie par

$$m_{f_j,t}(\tilde{\mathbf{x}}_j(t)) = \prod_{d=1}^{d=D} (1 - |\tilde{z}_{x_{j,d}}(t) - \hat{z}_{x_{j,d}}(t)|). \quad (\text{A.74})$$

Notamment, la quantité $m_{f_j,t}(\tilde{\mathbf{x}}_j(t))$ représente la croyance allouée à l'estimée de l'attribut de zone $\tilde{\mathbf{x}}_j(t)$. Plus l'élément $\hat{z}_{x_{j,d}}(t)$ est proche de $\tilde{z}_{x_{j,d}}(t)$, plus $|\tilde{z}_{x_{j,d}}(t) - \hat{z}_{x_{j,d}}(t)|$ est petite, et donc plus la croyance accordée à $\tilde{z}_{x_{j,d}}(t)$ devrait être grande. L'estimation en utilisant les fingerprints est à combiner avec celle issue de l'accéléromètre afin d'affiner les choix.

A.5.2.3 Zonage par mobilité

Les boîtes de positions $[\hat{\mathbf{x}}_j](t)$ sont générées au fur et à mesure de la même façon que dans la section A.3.2.1. Les estimées de la zone du nœud j à l'instant t sont alors obtenues en intersectant $[\hat{\mathbf{x}}_j](t)$ avec toutes les zones de la région de surveillance. Nous définissons ici la deuxième fonction de masse $m_{\gamma_j,t}(\cdot)$, dont les éléments focaux sont les attributs des zones ayant une intersection avec $[\hat{\mathbf{x}}_j](t)$, avec leurs masses associées définies en fonction de la surface de l'intersection entre $[\hat{\mathbf{x}}_j](t)$ et chaque zone, comme illustré en détail dans la Figure V.3.

A.5.2.4 Combinaison

Une fois les deux fonctions de masses obtenues, il nous reste la procédure de combinaison en utilisant la règle conjonctive. Premièrement, les attributs de zones estimés par fingerprints sont pris en intersection avec ceux de l'accéléromètre. Ensuite, les masses liés

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

aux attributs résultants sont obtenues par la multiplication des masses correspondantes comme suit,

$$m_{\cap j,t}(\tilde{\mathbf{z}}_{\mathbf{x}_j}(t)) = m_{f j,t}(\tilde{\mathbf{z}}_{\mathbf{x}_j}(t)) * m_{\gamma j,t}(\tilde{\mathbf{z}}_{\mathbf{x}_j}(t)), \quad (\text{A.75})$$

où $\tilde{\mathbf{z}}_{\mathbf{x}_j}(t) = (\tilde{z}_{x_{j,1}}(t), \dots, \tilde{z}_{x_{j,D}}(t))$ est l'attribut de zone compris parmi les zones estimées par fingerprints et accéléromètre. Notez que dans le cas de conflit entre $m_{f j,t}(\cdot)$ et $m_{\gamma j,t}(\cdot)$, le calcul de $m_{\cap j,t}(\cdot)$ doit être suivi par une phase de normalisation pour que $\Sigma m_{\cap j,t}(\cdot) = 1$. La fonction de masse globale possède alors une ou plusieurs zones avec des confiances leur correspondant. Plus la masse d'une zone est grande, plus la possibilité d'avoir le nœud j dedans est grande. Une liste de zones possibles est alors fournie, avec un ordre de priorité qui est le sens descendant de leurs masses.

A.5.3 Analyse et simulations

Les performances de notre méthode de zonage sont illustrées dans cette partie. Pour cette raison, nous considérons une région sous surveillance de $100m \times 100m$ composée de N_Z zones où les ancres et les positions de référence sont déployées uniformément. Les données de deux signaux d'accélération et la trajectoire sont simulées de la même manière que précédemment; elles sont montrées respectivement dans Figure V.4 et Figure V.5 avec 16 ancres et 81 positions de références. Les valeurs de RSSIs, $\xi_{P k,i}$ et $\xi_{X i}(t)$ sont également générées en utilisant les distances $\|\mathbf{a}_i, \mathbf{p}_k\|$, ou $\|\mathbf{a}_i, \mathbf{x}(t)\|$ selon le modèle de perte du canal, avec la puissance de transmission $\xi_0 = 100dBm$ mesurée à la distance $d_0 = 0.1m$. On ajoute aux RSSIs des bruits gaussiens $\varepsilon_{\xi,k,i}$ et $\varepsilon_{\xi_i}(t)$ de moyenne nulle et d'écart-type σ_ξ . On ajoute également aux accélérations un bruit gaussien $\varepsilon_{\gamma_d}(t)$ de moyenne nulle et d'écart-type σ_γ comme suit,

$$\gamma_d(t) = \tilde{\gamma}_d(t) + \varepsilon_{\gamma_d}(t), \quad d = 1, 2. \quad (\text{A.76})$$

A.5.3.1 Illustration générale

Pour démontrer la performance de notre méthode, nous prenons $N_A = 16$, $N_P = 81$, $N_Z = 9$ et $K = 3$. Avec $N_Z = 9$, $n_{Z_1} = n_{Z_2} = 3$. σ_ξ est réglé de 1% à 10% de l'écart-type de la base de donnée des RSSIs ξ_P et ξ_X , et σ_γ est également réglé de 1% à 10% de l'écart-type de l'accélération γ . Les résultats du zonage sont enregistrés dans Table V.1 en termes du taux de bonnes réponses ou *hitting rate* sur les 100 positions du nœud avec des valeurs différentes de σ_ξ et σ_γ . Il est clair que la méthode proposée donne de bons résultats même avec des bruits grandissants.

A.5.3.2 Impact des paramètres

Nous explorons ensuite les impacts des paramètres, tels que le nombre des zones N_Z , le nombre de positions de référence N_P , le nombre des ancres N_A et les écarts-types des bruits σ_γ et σ_ξ . En utilisant la trajectoire de Figure V.5, les résultats obtenus sont montrés dans Table V.2 ~ Table V.6. À partir des résultats, l'erreur d'estimation par rapport à N_A et N_P diminue avec l'augmentation du nombre, au détriment du temps de

calcul qui augmente légèrement. Par contre, l'erreur d'estimation par rapport à N_Z , σ_γ et σ_ξ augmente avec l'augmentation de leurs valeurs. Toutefois, notre méthode améliore la précision de zonage par rapport à l'utilisation d'un seul type de mesures, pour tous les deux cas avec 1 zone proposée et 2 zones.

A.5.3.3 Comparaison aux méthodes de classification par SVM

Le problème du zonage peut être vu comme un problème de classification, où l'on cherche à déterminer la classe ou zone d'appartenance du nœud mobile. Une technique de classification universelle est connue comme le support vector machine (SVM) basée sur les noyaux [Sangnier *et al.* 2013; Tsochantaridis *et al.* 2004]. De ce point de vue, nous comparons notre méthode proposée et celle basée sur les SVM à noyaux, ainsi que celle des SVMs structurés [Alain Rakotomamonjy 2008; Thorsten Joachims 2008]. Table V.8 montre les résultats de simulations en termes de taux de bonnes réponses sur les 100 positions du nœud pour 50 tentatives. D'après ces résultats, la méthode de fingerprinting et la méthode combinée dépassent en performance les méthodes de SVM. En outre, le fait de fournir un second choix de zones réalise un taux de quasi-100% dans le cas où le premier choix est erroné.

A.6 Conclusion et perspectives

La recherche sur les techniques de localisation dans les réseaux de capteurs sans fil a été développée depuis une vingtaine d'années, fournissant toutes sortes d'idées, d'essais, même d'implémentation au cours de ces années. Ce manuscrit a présenté plusieurs techniques d'estimation d'état pour les réseaux de capteurs sans fil, en utilisant les informations de fingerprints et d'accélération. Il a abordé les problèmes de la localisation indoor et du zonage dans le cadre de l'analyse par intervalles. Dans la suite, les méthodes de localisation proposées sont brièvement révisées, suivies par les perspectives de nos travaux.

A.6.1 Conclusion

Les méthodes proposées se situent autour du concept de l'incertitude sur les positions estimées. Elles visent à rassembler toutes les estimations de positions dans une boîte incluant sûrement la vraie position du capteur. Elles sont résumées comme suit,

- Ce manuscrit commence avec une configuration de réseau relativement simple où une localisation centralisée est autorisée à être mise en œuvre. Dans ce réseau, il existe deux types de capteurs, des ancres avec des positions connues et des nœuds mobiles avec des positions inconnues. En utilisant les positions de référence, la méthode de fingerprintings localise alors le nœud mobile en utilisant la puissance du signal reçu et la reconnaissance de formes de position. En même temps, les nœuds mobiles équipés d'accéléromètres fournissent également des informations de mobilité, telles que les accélérations. Selon les modèles de mobilité proposées,

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

les positions de nœuds sont facilement calculées. En considérant les bruits inévitables existants à l'intérieur ou à l'extérieur, les estimations exactes sont indéterminées. La méthode proposée introduit alors l'incertitude sur les mesures, les puissances des signaux et les accélérations, conduisant à des estimations en boîtes dans laquelle les positions réelles sont supposées incluses. Ayant deux boîtes de positions estimées, chacune déduite d'un type de mesures, la méthode proposée en prend l'intersection réduisant par conséquent la gamme de possibilités. La boîte mise à jour sert alors à la localisation dans la prochaine estimation.

- Un système de localisation clusterisée est également développé. Il se distingue par la configuration du réseau par rapport à la méthode centralisée. En effet, il procède à la collection de fingerprints dans un cadre clustérisé, où la zone de surveillance est divisée en plusieurs blocs individuels avec des agents individuels ou centres de calcul. Les estimations exécutées aux agents individuels conduisent à plusieurs estimations locales des emplacements des nœuds. Leur combinaison utilisant les poids des blocs conduit à une estimation globale. Ce régime considère en outre les informations inertielles des nœuds, fournies par les accéléromètres et les gyroscopes, en considérant la rotation du nœud, ce qui se produit normalement lorsque les nœuds se déplacent ou même restent statiques. À part la solution par l'analyse par intervalles, nous proposons une méthode basée sur le filtre de Kalman utilisant respectivement les informations de RSSIs et d'accélérations pour l'observation et la prédiction.
- Le zonage, ou localisation par pièce, est considéré dans la dernière partie. L'idée de localisation proposée est étendue au cadre du zonage, où la surface de surveillance entière est divisée en plusieurs zones, ou en unités. L'objectif est de retrouver la zone à laquelle appartient l'objet mobile. De même que pour la localisation centralisée, le procédé de zonage effectue la collecte de fingerprints sur toutes les zones, et puis applique l'algorithme des K plus proches voisins en utilisant les RSSIs du nœud mobile. Au lieu d'utiliser les positions de voisins, l'algorithme proposé utilise leurs attributs de zones pour faire estimer les zones concernées. En même temps, les nœuds déduisent leur adresses de zones en utilisant les informations inertielles et la technique d'analyse par intervalles. En ajoutant une incertitude sur les valeurs d'accélérations mesurées, la position calculée s'avère être aussi en termes d'intervalles, à savoir boîtes avec certaines surfaces sur des zones spécifiques. En utilisant la théorie des fonctions de croyance, les mesures conduisent à des ensembles de zones possibles, avec des facteurs de confiance associés. La combinaison selon la règle conjonctive rétrécit les possibilités de zones où les nœuds résident. En général, la méthode de zonage proposée offre deux choix de zones estimées dans un ordre spécifique, assurant un taux de bonnes réponses de plus de 90% pour le premier choix, et jusqu'à 100% pour le deuxième choix quand le premier est erroné.

Les performances des méthodes proposées sont analysées avec différents réglages de paramètres. Elles sont également comparées à d'autres méthodes avec diverses con-

figurations de réseaux. D'une manière générale, les méthodes proposées réalisent une meilleure précision de localisation et un coût de calcul plus faible dans la plupart des cas. Toutefois, les désavantages existent, et restent à être discutés dans la suite.

A.6.2 Perspectives

Les méthodes de localisation proposées visent à localiser un nœud mobile aussi précisément que possible mais avec un temps et une consommation d'énergie aussi faibles que possible. Le concept de combinaison de différents types d'informations et l'analyse par intervalles fournissent une solution possible pour réaliser cet objectif. Cependant, les méthodes proposées ont encore quelques aspects à améliorer et travailler à l'avenir.

- Sélection du KNN : Il existe parfois des cas où les K plus proches voisins sélectionnés ne soient pas les bons. En d'autres termes, certaines des positions les plus proches géographiquement sont exclues de l'ensemble des voisins quand ils sont trop près d'une ancre par rapport aux autres comme indiqué dans la Figure V.6. Ce problème entraîne un léger biais dans l'estimation de la position finale pour le problème de localisation. Il pourrait toutefois donner une adresse fausse au niveau du zonage. En fait, ceci est issu de la nature de la relation logarithmique existante entre les RSSIs et les distances séparant deux capteurs. Pour remédier à ce problème, on pourrait travailler sur l'optimisation des positions de référence par rapport aux ancres à l'avance.
- Calculs sur les intervalle : Comme discuté dans la section A.3 lors du calcul des facteurs de poids, il existe certains problèmes sur l'opération de division en termes d'analyse par intervalles. D'une part, il n'est pas possible de diviser par un intervalle comprenant le zéro, sinon ceci conduit à une valeur de poids infinie. Une question similaire relève aussi des incertitudes δ_γ et δ_ξ . En fait, il existe un compromis entre l'incertitude et la précision des estimations finales. Se elles sont beaucoup plus petites que les valeurs normales, les estimations en boîte n'incluent pas la position réelle pour certains points. Si, cependant, elles sont réglées pour être beaucoup plus grandes, les boîtes sont sans signification puisque elles seront trop larges. Ces problèmes restent donc à être traités dans l'avenir.
- Choix entre les deux types d'estimées : Les algorithmes proposés considèrent l'intersection des estimées par fingerprinting et accélérations. Toutefois, si ces données sont conflictuelles, l'intersection s'avère vide. Dans ce cas, les algorithmes adoptent une stratégie de pick-up choisissant l'une ou l'autres des estimations individuelles. En pratique, on suppose que la méthode de fingerprintings est plus fiable que la méthode d'accéléromètre puisque un brusque changement au cours des mesures des accélérations peut-être conduit à un résultat inattendu. Par conséquent, lorsque les boîtes estimées ne se croisent pas, les résultats de fingerprintings seront adoptés. Toutefois, cette mesure fait parfois des erreurs plus importantes. Il arrive aussi, même quand l'intersection existe, d'obtenir

A. APPENDIX: RÉSUMÉ EN FRANÇAIS

une erreur finale plus grande que l'erreur d'une des deux estimations. Ainsi, la stratégie d'accepter ou de refuser le résultat reste à être discutée.

- Problème de zonage : L'algorithme proposé suppose que les zones sont égales, ce qui permet d'adresser le problème en utilisant l'algorithme de KNN sur les zones de référence. Cependant, dans les vrais scénarios, les zones sont les pièces, les bureaux, les couloirs ou autres, ayant des architectures et des dimensions différentes. Une solution possible consiste à diviser l'espace entier en zones égales, indépendamment de l'architecture globale, ou choisir les dimensions spécifiques pour les zones, ce qui permet une subdivision de toutes les pièces en zones égales. Les travaux futurs dans le cadre du zonage porteront sur le problème de la classification où la structure quelconque de l'espace pourrait être manipulée.

References

- [Akyildiz *et al.*, 2002] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam et E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38 (4): 393–422, 2002. 150, 151
- [Alain Rakotomamonjy, 2008] Alain Rakotomamonjy. SVM and Kernel Methods Matlab code and related softwares. 2008. <http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html>. 177
- [Aldhubaib et Shuley, 2010] F. Aldhubaib et N. Shuley. Radar target recognition based on modified characteristic polarization states. *IEEE Transactions on Aerospace and Electronic Systems*, 46 (4): 1921–1933, 2010. 166
- [Arthi et Murugan, 2010] R. Arthi et K. Murugan. Localization in wireless sensor networks by hidden markov model. In *Advanced Computing (ICoAC), 2010 Second International Conference on*, pages 14–18. IEEE, 2010. 159
- [Baggio et Langendoen, 2006a] A. Baggio et K. Langendoen. Monte-carlo localization for mobile wireless sensor networks. In *Mobile Ad-hoc and Sensor Networks*, pages 317–328. Springer, 2006. 159, 165
- [Baggio et Langendoen, 2006b] A. Baggio et K. Langendoen. Monte-carlo localization for mobile wireless sensor networks. *Second international conference on mobile ad hoc and sensor networks (MSN)*, 2006. 171
- [Bahl et Padmanabhan, 2000] P. Bahl et V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. IEEE, 2000. 153
- [Bandyopadhyay et Coyle, 2003] S. Bandyopadhyay et E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1713–1723. IEEE, 2003. 151
- [Bharathidasan et Ponduru, 2002] A. Bharathidasan et V. A. S. Ponduru. Sensor networks: An overview. *Department of Computer Science. University of California*, 2002. 150
- [Castro *et al.*, 2001] P. Castro, P. Chiu, T. Kremenek et R. Muntz. A probabilistic room location service for wireless networked environments. In *UbiComp 2001: Ubiquitous Computing*, pages 18–34. Springer, 2001. 172
- [Chan *et al.*, 1979] Y. Chan, A. G. Hu et J. Plant. A kalman filter based tracking scheme with input estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15 (2): 237–244, 1979. 166
- [Chehri *et al.*, 2009] A. Chehri, P. Fortier et P. M. Tardif. Uwb-based sensor networks for localization in mining environments. *Ad Hoc Networks*, 7 (5): 987–1000, 2009. 151
- [Chen et Chang, 2008] H. Chen et K. Chang. K-nearest neighbor particle filters for dynamic hybrid bayesian networks. *IEEE Transactions on Aerospace and Electronic Systems*, 44 (3): 1091–1101, 2008. 166

REFERENCES

- [Cheong *et al.*, 2011] P. Cheong, K.-F. Chang, Y.-H. Lai, S.-K. Ho, I.-K. Sou et K.-W. Tam. A zigbee-based wireless sensor network node for ultraviolet detection of flame. *Industrial Electronics, IEEE Transactions on*, 58 (11): 5271–5277, 2011. [151](#)
- [Chiu *et al.*, 2009] Y.-M. Chiu, K. Wang, R.-H. Jan, Y.-J. Hu et T.-H. Ku. An efficient room-based indoor localization scheme for wireless sensor networks. 2009. [172](#)
- [Choi *et al.*, 2011] B.-S. Choi, J.-W. Lee, J.-J. Lee et K.-T. Park. A hierarchical algorithm for indoor mobile robot localization using rfid sensor fusion. *Industrial Electronics, IEEE Transactions on*, 58 (6): 2226–2235, 2011. [151](#), [159](#)
- [Chuenurajit *et al.*, 2013] T. Chuenurajit, S. Phimmasean et P. Cherntanomwong. Robustness of 3d indoor localization based on fingerprint technique in wireless sensor networks. In *10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6, May 2013. [160](#), [171](#)
- [Dagher *et al.*, 2014] R. Dagher, N. Mitton, I. Amadou *et al.* Towards wsn-aided navigation for vehicles in smart cities: An application case study. In *1st International IEEE Percom Workshop on Pervasive Systems for Smart Cities (PerCity 2014)*, 2014. [150](#)
- [Dempster, 1967] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, pages 325–339, 1967. [172](#)
- [Denoeux, 1997] T. Denoeux. Analysis of evidence-theoretic decision rules for pattern classification. *Pattern recognition*, 30 (7): 1095–1107, 1997. [172](#), [173](#)
- [D’hoë *et al.*, 2008] K. D’hoë, G. Ottoy, J. P. Goemaere et L. De Strycker. Indoor room location estimation. *Advances in electrical and computer engineering*, 8 (2): 78–81, 2008. [172](#)
- [Erasala et Yen, 2002] N. Erasala et D. C. Yen. Bluetooth technology: a strategic analysis of its role in global 3g wireless communication era. *Computer Standards & Interfaces*, 24 (3): 193–206, 2002. [151](#)
- [Farjow *et al.*, 2011] W. Farjow, A. Chehri, M. Hussein et X. Fernando. Support vector machines for indoor sensor localization. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 779–783, March 2011. [171](#)
- [Farmani *et al.*, 2012] M. Farmani, H. Moradi et M. Asadpour. A hybrid localization approach in wireless sensor networks using a mobile beacon and inter-node communication. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 269–274, May 2012. [171](#)
- [Ferrari et Pagliari, 2006] G. Ferrari et R. Pagliari. Decentralized binary detection with noisy communication links. *IEEE Transactions on Aerospace and Electronic Systems*, 42 (4): 1554–1563, 2006. [165](#)
- [Fraile et Maybank, 1998] R. Fraile et S. J. Maybank. Vehicle trajectory approximation and classification. In *BMVC*, volume 98, pages 832–840, 1998. [159](#)
- [Gao et Guo, 2010] B. Gao et S. Guo. Development of an infrared sensor-based wireless intelligent fish-like underwater microrobot. In *Information and Automation (ICIA), 2010 IEEE International Conference on*, pages 1314–1318. IEEE, 2010. [151](#)

-
- [Gogolak *et al.*, 2011] L. Gogolak, S. Pletl et D. Kukolj. Indoor fingerprint localization in wsn environment based on neural network. In *Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on*, pages 293–296. IEEE, 2011. 171
- [Hande *et al.*, 2006] A. Hande, T. Polk, W. Walker et D. Bhatia. Self-powered wireless sensor networks for remote patient monitoring in hospitals. *Sensors*, 6 (9): 1102–1117, 2006. 150
- [Hansen et Walster, 2003] E. Hansen et G. W. Walster. *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press, 2003. 153
- [Haverinen et Kemppainen, 2009] J. Haverinen et A. Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57 (10): 1028–1035, 2009. 159
- [Hill, 2003] J. L. Hill. *System architecture for wireless sensor networks*. Thèse de doctorat, University of California, Berkeley, 2003. 150
- [Hofmann-Wellenhof *et al.*, 2004] B. Hofmann-Wellenhof, H. Lichtenegger et J. Collins. *Global Positioning System: Theory and Practice*. Springer, September 2004. 152
- [Honghui et Moore, 2002] Q. Honghui et J. B. Moore. Direct kalman filtering approach for gps/ins integration. *IEEE Transactions on Aerospace and Electronic Systems*, 38 (2): 687–693, 2002. 166
- [Hotta *et al.*, 2012] S. Hotta, Y. Hada et Y. Yaginuma. A robust room-level localization method based on transition probability for indoor environments. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–8, Nov 2012. 172
- [Hsu et Yu, 2009] C.-H. Hsu et C.-H. Yu. An accelerometer based approach for indoor localization. In *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on*, pages 223–227. IEEE, 2009. 159, 161
- [Hu et Evans, 2004] L. Hu et D. Evans. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 45–57. ACM, 2004. 159
- [Ilyas et Mahgoub, 2004] M. Ilyas et I. Mahgoub. *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press, 2004. 150
- [Jaulin *et al.*, 2001a] L. Jaulin, M. Kieffer, O. Didrit et E. Walter. *Applied interval analysis*. Springer, 2001. 153, 155, 162, 172, 174
- [Jaulin *et al.*, 2001b] L. Jaulin, M. Kieffer, O. Didrit et E. Walter. *Applied interval analysis*. Springer, 2001. 166
- [Jaulin et Walter, 1993] L. Jaulin et E. Walter. Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Mathematics and Computers in Simulation*, 35 (2): 123–137, 1993. 153
- [Kaemarungsi et Krishnamurthy, 2004] K. Kaemarungsi et P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1012–1022. IEEE, 2004. 153

REFERENCES

- [Kalman, 1960] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82 (1): 35–45, 1960. [166](#), [169](#)
- [Kearfott et Kreinovich, 1996] R. B. Kearfott et V. Kreinovich. *Applications of interval computations*, volume 3. Kluwer Academic Dordrecht, 1996. [153](#), [155](#)
- [Krisp, 2013] J. M. Krisp. *Progress in location-based services*. Springer, 2013. [171](#)
- [Lee et Huang, 2002] S.-C. Lee et Y.-C. Huang. Innovative estimation method with measurement likelihood for all-accelerometer type inertial navigation system. *IEEE Transactions on Aerospace and Electronic Systems*, 38 (1): 339–346, 2002. [166](#)
- [Lefevre et al., 2002] E. Lefevre, O. Colot et P. Vannoorenberghe. Belief function combination and conflict management. *Information fusion*, 3 (2): 149–162, 2002. [172](#)
- [Li et al., 2011] L. Li, H. Xiaoguang, C. Ke et H. Ketai. The applications of wifi-based wireless sensor network in internet of things and smart grid. In *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, pages 789–793. IEEE, 2011. [151](#)
- [Liu et al., 2012] D. Liu, D. Tang, Z. Xu, Y. Hu et C. Huang. The research of the fuzzy cluster algorithm for indoor location based on rssi. In *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pages 5–7. IEEE, 2012. [171](#)
- [Lo et al., 2012] C.-C. Lo, C.-C. Chen, Y.-C. Tseng, J.-C. Chiang, K.-C. Feng, L.-C. Kuo et Y.-C. Wang. A room-based localization system using wireless triggers and pattern matching techniques. In *IEEE VTS Asia Pacific Wireless Communications Symposium B. Kyoto: Susumu Yoshida*, volume 4, pages 11–13, 2012. [172](#)
- [Mahfouz et al., 2013] S. Mahfouz, F. Mourad-Chehade, P. Honeine, H. Snoussi et J. Farah. Kernel-based localization using fingerprinting in wireless sensor networks. In *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 744–748, June 2013. [171](#)
- [Mainwaring et al., 2002] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk et J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002. [150](#)
- [Mao et al., 2007] G. Mao, B. Fidan et B. Anderson. Wireless sensor network localization techniques. *Computer networks*, 51 (10): 2529–2553, 2007. [152](#)
- [Medeisis et Kajackas, 2000a] A. Medeisis et A. Kajackas. On the use of the universal okumura-hata propagation prediction model in rural areas. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 3, pages 1815–1818. IEEE, 2000. [164](#)
- [Medeisis et Kajackas, 2000b] A. Medeisis et A. Kajackas. On the use of the universal okumura-hata propagation prediction model in rural areas. In *Vehicular Technology Conference Proceedings*, volume 3, pages 1815–1818, VTC Tokyo, May 2000. [170](#)
- [Moore, 1992] R. Moore. Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, 34 (2): 113–119, 1992. [153](#)

- [Moore, 1966] R. E. Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966. [153](#), [166](#)
- [Moore et Moore, 1979] R. E. Moore et R. Moore. *Methods and applications of interval analysis*, volume 2. SIAM, 1979. [155](#), [172](#)
- [Mourad, 2010] F. Mourad. *Auto-localisation et suivi de cibles dans les réseaux de capteurs mobiles*. Thèse de doctorat, Université de Technologie de Troyes, Décembre 2010. [151](#)
- [Mourad et al., 2008] F. Mourad, H. Snoussi, F. Abdallah et C. Richard. Guaranteed boxed localization in manets by interval analysis and constraints propagation techniques. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5. IEEE, 2008. [159](#)
- [Mourad et al., 2009a] F. Mourad, H. Snoussi, F. Abdallah et C. Richard. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *Signal Processing, IEEE Transactions on*, 57 (8): 3226–3239, 2009. [159](#), [162](#)
- [Mourad et al., 2009b] F. Mourad, H. Snoussi, F. Abdallah et C. Richard. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on Signal Processing*, 57 (8): 3226–3239, 2009. [166](#)
- [Navarro et al., 2010] E. Navarro, B. Peucker et M. Quan. Wi-fi localization using rssi fingerprinting. In *California Polytechnic State University, USA*, pages 1–6, 2010. [160](#), [171](#)
- [Neumaier, 1990] A. Neumaier. *Interval methods for systems of equations*, volume 37. Cambridge university press, 1990. [153](#)
- [Ni et al., 2004] L. M. Ni, Y. Liu, Y. C. Lau et A. P. Patil. Landmarc: indoor location sensing using active rfid. *Wireless networks*, 10 (6): 701–710, 2004. [159](#)
- [Noury et al., 2002] N. Noury, G. Virone et T. Creuzet. The health integrated smart home information system (his 2): rules based system for the localization of a human. In *Microtechnologies in Medicine & Biology 2nd Annual International IEEE-EMB Special Topic Conference on*, pages 318–321. IEEE, 2002. [172](#)
- [Pal, 2010] A. Pal. Localization algorithms in wireless sensor networks: Current approaches and future challenges. *Network Protocols and Algorithms*, 2 (1): 45–73, 2010. [152](#)
- [Priwgharm et Chemtanomwong, 2011] R. Priwgharm et P. Chemtanomwong. A comparative study on indoor localization based on rssi measurement in wireless sensor network. In *Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6, May 2011. [171](#)
- [Ristic et Smets, 2005] B. Ristic et P. Smets. Target identification using belief functions and implication rules. *Aerospace and Electronic Systems, IEEE Transactions on*, 41 (3): 1097–1103, 2005. [172](#)
- [Robles et al., 2010] J. J. Robles, M. Deicke et R. Lehnert. 3d fingerprint-based localization for wireless sensor networks. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, pages 77–85. IEEE, 2010. [159](#)

REFERENCES

- [Rozyyev *et al.*, 2012] A. Rozyyev, H. Hasbullah et F. Subhan. Combined k-nearest neighbors and fuzzy logic indoor localization technique for wireless sensor network. *Research Journal of Information Technology*, 4 (4): 155–165, 2012. [171](#)
- [Runge *et al.*, 2011] A. Runge, M. Baunach et R. Kolla. Precise self-calibration of ultrasound based indoor localization systems. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–8. IEEE, 2011. [151](#)
- [Sangnier *et al.*, 2013] M. Sangnier, J. Gauthier et A. Rakotomamonjy. Filter bank kernel learning for nonstationary signal classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3183–3187. IEEE, 2013. [177](#)
- [Schiller et Voisard, 2004] J. Schiller et A. Voisard. *Location-based services*. Elsevier, 2004. [171](#)
- [Schuler *et al.*, 1967] A. R. Schuler, A. Grammatikos et K. A. FEGLEY. Measuring rotational motion with linear accelerometers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-3 (3): 465–472, 1967. [166](#)
- [Schweinzer et Syafrudin, 2010] H. Schweinzer et M. Syafrudin. Losnus: An ultrasonic system enabling high accuracy and secure tdoa locating of numerous devices. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–8. IEEE, 2010. [153](#)
- [Shafer *et al.*, 1976] G. Shafer *et al.* *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976. [172](#)
- [Shahid *et al.*, 2010] B. Shahid, A. A. Kannan, N. H. Lovell et S. Redmond. Ultrasound user-identification for wireless sensor networks. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 5756–5759. IEEE, 2010. [151](#)
- [Sifuentes *et al.*, 2011] E. Sifuentes, O. Casas et R. Pallas-Areny. Wireless magnetic sensor node for vehicle detection with optical wake-up. *Sensors Journal, IEEE*, 11 (8): 1669–1676, 2011. [153](#)
- [Simon *et al.*, 2004] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai et K. Frampton. Sensor network-based countersniper system. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 1–12. ACM, 2004. [150](#)
- [Smets, 1993] P. Smets. Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of approximate reasoning*, 9 (1): 1–35, 1993. [173](#)
- [Smets et Kennes, 1994] P. Smets et R. Kennes. The transferable belief model. *Artificial intelligence*, 66 (2): 191–234, 1994. [172](#), [173](#), [174](#)
- [Stojmenovic, 2005] I. Stojmenovic. *Handbook of sensor networks: algorithms and architectures*, volume 49. John Wiley & Sons, 2005. [152](#), [171](#)
- [Sunaga, 1958] T. Sunaga. Theory of an interval algebra and its application to numerical analysis. *Research Association of Applied Geometry (RAAG) Memoirs*, 2: 29–46, 1958. [153](#)

- [Sunaga, 2009] T. Sunaga. Theory of an interval algebra and its application to numerical analysis. *Japan Journal of Industrial and Applied Mathematics*, 26 (2): 125–143, 2009. 153
- [Suroso *et al.*, 2011] D. J. Suroso, P. Cherntanomwong, P. Sooraksa et J.-i. Takada. Fingerprint-based technique for indoor localization in wireless sensor networks using fuzzy c-means clustering algorithm. In *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*, pages 1–5. IEEE, 2011. 171
- [Tapus et Siegwart, 2006] A. Tapus et R. Siegwart. A cognitive modeling of space using fingerprints of places for mobile robot navigation. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1188–1193. IEEE, 2006. 159
- [Teng *et al.*, 2010] J. Teng, H. Snoussi et C. Richard. Decentralized variational filtering for target tracking in binary sensor networks. *Mobile Computing, IEEE Transactions on*, 9(10): 1465–1477, 2010. 159
- [Tharmarasa *et al.*, 2011] R. Tharmarasa, T. Kirubarajan, A. Sinha et T. Lang. Decentralized sensor selection for large-scale multisensor-multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 47 (2): 1307–1324, 2011. 165
- [Thorsten Joachims, 2008] Thorsten Joachims. An implementation of Support Vector Machines (SVMs). 2008. <http://svmlight.joachims.org/>. 177
- [Tsochantaridis *et al.*, 2004] I. Tsochantaridis, T. Hofmann, T. Joachims et Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004. 177
- [Walter et Jaulin, 1994] E. Walter et L. Jaulin. Guaranteed characterization of stability domains via set inversion. *Automatic Control, IEEE Transactions on*, 39 (4): 886–889, 1994. 153
- [Yick *et al.*, 2008] J. Yick, B. Mukherjee et D. Ghosal. Wireless sensor network survey. *Computer networks*, 52 (12): 2292–2330, 2008. 152
- [Zhao et Guibas, 2004] F. Zhao et L. J. Guibas. *Wireless sensor networks: an information processing approach*. Morgan Kaufmann, 2004. 150

REFERENCES

References

- I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.
- Alain Rakotomamonjy. SVM and Kernel Methods Matlab code and related softwares. 2008. <http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html>.
- F. Aldhubaib and N. Shuley. Radar target recognition based on modified characteristic polarization states. *IEEE Transactions on Aerospace and Electronic Systems*, 46(4):1921–1933, 2010.
- R. Arthi and K. Murugan. Localization in wireless sensor networks by hidden markov model. In *Advanced Computing (ICoAC), 2010 Second International Conference on*, pages 14–18. IEEE, 2010.
- J. Bachrach, R. Nagpal, M. Salib, and H. Shrobe. Experimental results for and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks. *Telecommunication Systems*, 26(2-4):213–233, 2004.
- A. Baggio and K. Langendoen. Monte-carlo localization for mobile wireless sensor networks. In *Mobile Ad-hoc and Sensor Networks*, pages 317–328. Springer, 2006a.
- A. Baggio and K. Langendoen. Monte-carlo localization for mobile wireless sensor networks. *Second international conference on mobile ad hoc and sensor networks (MSN)*, 2006b.
- A. Baggio and K. Langendoen. Monte carlo localization for mobile wireless sensor networks. *Ad Hoc Networks*, 6(5):718–733, 2008.
- P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. IEEE, 2000.
- S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1713–1723. IEEE, 2003.
- F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget. Revising hull and box consistency. In *Int. Conf. on Logic Programming*. Citeseer, 1999.

-
- A. Bharathidasan and V. A. S. Ponduru. Sensor networks: An overview. *Department of Computer Science. University of California*, 2002.
- E. Biagioni. Pods: Interpreting spatial and temporal environmental information. In *In Usability Evaluation and Interface Design: Cognitive Engineering, Intelligent Agents, and Virtual Reality, Volume I of the Proceedings of HCI International 2001, the 9th International Conference on Human-Computer Interaction*. Citeseer, 2001.
- E. S. Biagioni and K. Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16(3):315–324, 2002.
- P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004.
- S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994.
- N. Bulusu, V. Bychkovskiy, D. Estrin, and J. Heidemann. Scalable, ad hoc deployable, rf-based localization. In *Proceedings of the Grace Hopper Conference on Celebration of Women in Computing*, volume 31, 2002.
- N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, 7(5):28–34, 2000.
- N. Bulusu, J. Heidemann, and D. Estrin. Adaptive beacon placement. In *Distributed Computing Systems, 2001. 21st International Conference on.*, pages 489–498. IEEE, 2001.
- T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.
- S. Čapkun, M. Hamdi, and J.-P. Hubaux. Gps-free positioning in mobile ad hoc networks. *Cluster Computing*, 5(2):157–167, 2002.
- P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A probabilistic room location service for wireless networked environments. In *Ubicomp 2001: Ubiquitous Computing*, pages 18–34. Springer, 2001.
- Y. Chan, A. G. Hu, and J. Plant. A kalman filter based tracking scheme with input estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(2):237–244, 1979.
- A.-K. Chandra-Sekaran, P. Dheenathayalan, P. Weisser, C. Kunze, and W. Stork. Empirical analysis and ranging using environment and mobility adaptive rssi filter for patient localization during disaster management. In *Networking and Services, 2009. ICNS'09. Fifth International Conference on*, pages 276–281. IEEE, 2009.
- A. Chehri, P. Fortier, and P. M. Tardif. Uwb-based sensor networks for localization in mining environments. *Ad Hoc Networks*, 7(5):987–1000, 2009.
- H. Chen and K. Chang. K-nearest neighbor particle filters for dynamic hybrid bayesian networks. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1091–1101, 2008.

-
- P. Cheong, K.-F. Chang, Y.-H. Lai, S.-K. Ho, I.-K. Sou, and K.-W. Tam. A zigbee-based wireless sensor network node for ultraviolet detection of flame. *Industrial Electronics, IEEE Transactions on*, 58(11):5271–5277, 2011.
- Y.-M. Chiu, K. Wang, R.-H. Jan, Y.-J. Hu, and T.-H. Ku. An efficient room-based indoor localization scheme for wireless sensor networks. 2009.
- B.-S. Choi, J.-W. Lee, J.-J. Lee, and K.-T. Park. A hierarchical algorithm for indoor mobile robot localization using rfid sensor fusion. *Industrial Electronics, IEEE Transactions on*, 58(6):2226–2235, 2011.
- T. Chuenurajit, S. Phimmasean, and P. Cherntanomwong. Robustness of 3d indoor localization based on fingerprint technique in wireless sensor networks. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, pages 1–6. IEEE, 2013a.
- T. Chuenurajit, S. Phimmasean, and P. Cherntanomwong. Robustness of 3d indoor localization based on fingerprint technique in wireless sensor networks. In *10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6, May 2013b.
- J. G. Cleary. Logical arithmetic. 1986.
- M. B. Clowes. On seeing things. *Artificial intelligence*, 2(1):79–116, 1971.
- W. S. Conner, J. Heidemann, L. Krishnamurthy, X. Wang, and M. Yarvis. Workplace applications of sensor networks. *USC/ISI Technical Report ISI-TR-2004-591*, 2004.
- A. Czubak and J. Wojtanowski. On applications of wireless sensor networks. In *Internet-Technical Development and Applications*, pages 91–99. Springer, 2009.
- R. Dagher, N. Mitton, I. Amadou, et al. Towards wsn-aided navigation for vehicles in smart cities: An application case study. In *1st International IEEE Percom Workshop on Pervasive Systems for Smart Cities (PerCity 2014)*, 2014.
- E. Davis. Constraint propagation with interval labels. *Artificial intelligence*, 32(3):281–331, 1987.
- A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, pages 325–339, 1967.
- B. Denis, J.-B. Pierrot, and C. Abou-Rjeily. Joint distributed synchronization and positioning in uwb ad hoc networks using toa. *Microwave Theory and Techniques, IEEE Transactions on*, 54(4):1896–1911, 2006.
- T. Denoeux. Analysis of evidence-theoretic decision rules for pattern classification. *Pattern recognition*, 30(7):1095–1107, 1997.
- K. D’hoë, G. Ottoy, J. P. Goemaere, and L. De Strycker. Indoor room location estimation. *Advances in electrical and computer engineering*, 8(2):78–81, 2008.
- L. Doherty, L. El Ghaoui, et al. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663. IEEE, 2001.

-
- N. Erasala and D. C. Yen. Bluetooth technology: a strategic analysis of its role in global 3g wireless communication era. *Computer Standards & Interfaces*, 24(3):193–206, 2002.
- W. Farjow, A. Chehri, M. Hussein, and X. Fernando. Support vector machines for indoor sensor localization. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 779–783. IEEE, 2011a.
- W. Farjow, A. Chehri, M. Hussein, and X. Fernando. Support vector machines for indoor sensor localization. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 779–783, March 2011b.
- M. Farmani, H. Moradi, and M. Asadpour. A hybrid localization approach in wireless sensor networks using a mobile beacon and inter-node communication. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 269–274, May 2012.
- G. Ferrari and R. Pagliari. Decentralized binary detection with noisy communication links. *IEEE Transactions on Aerospace and Electronic Systems*, 42(4):1554–1563, 2006.
- R. Fraile and S. J. Maybank. Vehicle trajectory approximation and classification. In *BMVC*, volume 98, pages 832–840, 1998.
- B. Gao and S. Guo. Development of an infrared sensor-based wireless intelligent fish-like underwater microrobot. In *Information and Automation (ICIA), 2010 IEEE International Conference on*, pages 1314–1318. IEEE, 2010.
- L. Gogolak, S. Pletl, and D. Kukulj. Indoor fingerprint localization in wsn environment based on neural network. In *Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on*, pages 293–296. IEEE, 2011.
- I. Guvenc and C.-C. Chong. A survey on toa based wireless localization and nlos mitigation techniques. *Communications Surveys & Tutorials, IEEE*, 11(3):107–124, 2009.
- A. Hande, T. Polk, W. Walker, and D. Bhatia. Self-powered wireless sensor networks for remote patient monitoring in hospitals. *Sensors*, 6(9):1102–1117, 2006.
- E. Hansen and G. W. Walster. *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press, 2003.
- J. Haverinen and A. Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10):1028–1035, 2009.
- T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95. ACM, 2003.
- J. L. Hill. *System architecture for wireless sensor networks*. PhD thesis, University of California, Berkeley, 2003.
- B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System. Theory and Practice*. Springer, 1993.
- B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer, September 2004. ISBN 3211835342.

-
- Q. Honghui and J. B. Moore. Direct kalman filtering approach for gps/ins integration. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):687–693, 2002.
- S. Hotta, Y. Hada, and Y. Yaginuma. A robust room-level localization method based on transition probability for indoor environments. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–8, Nov 2012.
- C.-H. Hsu and C.-H. Yu. An accelerometer based approach for indoor localization. In *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on*, pages 223–227. IEEE, 2009a.
- C.-H. Hsu and C.-H. Yu. An accelerometer based approach for indoor localization. In *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, UIC-ATC'09*, pages 223–227. IEEE, 2009b.
- L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 45–57. ACM, 2004.
- M. Ilyas and I. Mahgoub. *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press, 2004.
- L. Jaulin. Solution globale et garantie de problèmes ensemblistes; application à l'estimation non linéaire et à la commande robuste. orsay, france, 1994.
- L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied interval analysis*. Springer, 2001a.
- L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied interval analysis*. Springer, 2001b.
- L. Jaulin and E. Walter. Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Mathematics and Computers in Simulation*, 35(2):123–137, 1993.
- X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2652–2661. IEEE, 2004.
- J.-A. Jiang, X.-Y. Zheng, Y.-F. Chen, C.-H. Wang, P.-T. Chen, C.-L. Chuang, and C.-P. Chen. A distributed rss-based localization using a dynamic circle expanding mechanism. *Sensors Journal, IEEE*, 13(10):3754–3766, 2013.
- P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebrantet. In *ACM Sigplan Notices*, volume 37, pages 96–107. ACM, 2002.
- K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1012–1022. IEEE, 2004.
- R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- A. A. Kannan, G. Mao, and B. Vucetic. Simulated annealing based localization in wireless sensor network. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 2–pp. IEEE, 2005.

-
- A. A. Kannan, G. Mao, and B. Vucetic. Simulated annealing based wireless sensor network localization with flip ambiguity mitigation. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 2, pages 1022–1026. IEEE, 2006.
- R. B. Kearfott and V. Kreinovich. *Applications of interval computations*, volume 3. Kluwer Academic Dordrecht, 1996.
- M. Kieffer. *Estimation ensembliste par analyse par intervalles Application à la localisation d'un véhicule*. PhD thesis, Université Paris Sud-Paris XI, 1999.
- D. Koks. Numerical calculations for passive geolocation scenarios. Technical report, DTIC Document, 2007.
- H. Koyuncu and S. H. Yang. Indoor positioning with virtual fingerprint mapping by using linear and exponential taper functions. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 1052–1057. IEEE, 2013.
- L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 64–75. ACM, 2005.
- J. M. Krisp. *Progress in location-based services*. Springer, 2013.
- A. Kulaib, R. Shubair, M. Al-Qutayri, and J. W. Ng. An overview of localization techniques for wireless sensor networks. In *Innovations in Information Technology (IIT), 2011 International Conference on*, pages 167–172. IEEE, 2011.
- S.-C. Lee and Y.-C. Huang. Innovative estimation method with measurement likelihood for all-accelerometer type inertial navigation system. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):339–346, 2002.
- E. Lefevre, O. Colot, and P. Vannooenberghe. Belief function combination and conflict management. *Information fusion*, 3(2):149–162, 2002.
- L. Li, H. Xiaoguang, C. Ke, and H. Ketai. The applications of wifi-based wireless sensor network in internet of things and smart grid. In *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, pages 789–793. IEEE, 2011.
- T.-C. Liang, T.-C. Wang, and Y. Ye. A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization. *Sanford University, formal report*, 5, 2004.
- D. Liu, D. Tang, Z. Xu, Y. Hu, and C. Huang. The research of the fuzzy cluster algorithm for indoor location based on rssi. In *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pages 5–7. IEEE, 2012.
- C.-C. Lo, C.-C. Chen, Y.-C. Tseng, J.-C. Chiang, K.-C. Feng, L.-C. Kuo, and Y.-C. Wang. A room-based localization system using wireless triggers and pattern matching techniques. In *IEEE VTS Asia Pacific Wireless Communications Symposium B. Kyoto: Susumu Yoshida*, volume 4, pages 11–13, 2012.

-
- K. Lorincz and M. Welsh. Motetrack: A robust, decentralized approach to rf-based location tracking. In *Location-and Context-Awareness*, pages 63–82. Springer, 2005.
- S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi. Decentralized localization using fingerprinting and kernel methods in wireless sensor networks. In *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*, pages 1–5, Sept 2013a.
- S. Mahfouz, F. Mourad-Chehade, P. Honeine, H. Snoussi, and J. Farah. Kernel-based localization using fingerprinting in wireless sensor networks. In *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on*, pages 744–748. IEEE, 2013b.
- S. Mahfouz, F. Mourad-Chehade, P. Honeine, H. Snoussi, and J. Farah. Kernel-based localization using fingerprinting in wireless sensor networks. In *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 744–748, June 2013c.
- A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002.
- G. Mao, B. Fidan, and B. Anderson. Wireless sensor network localization techniques. *Computer networks*, 51(10):2529–2553, 2007.
- A. Medeisis and A. Kajackas. On the use of the universal okumura-hata propagation prediction model in rural areas. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 3, pages 1815–1818. IEEE, 2000a.
- A. Medeisis and A. Kajackas. On the use of the universal okumura-hata propagation prediction model in rural areas. In *Vehicular Technology Conference Proceedings*, volume 3, pages 1815–1818, VTC Tokyo, May 2000b.
- L. Meertens and S. Fitzpatrick. The distributed construction of a global coordinate system in a network of static computational nodes from inter-node distances. *Kestrel Institute TR KES. U*, 4, 2004.
- D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61. ACM, 2004.
- R. Moore. Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, 34(2):113–119, 1992.
- R. E. Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966.
- R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to interval analysis*. Siam, 2009.
- R. E. Moore and R. Moore. *Methods and applications of interval analysis*, volume 2. SIAM, 1979.
- F. Mourad. *Auto-localisation et suivi de cibles dans les réseaux de capteurs mobiles*. PhD thesis, Université de Technologie de Troyes, Décembre 2010.

-
- F. Mourad, H. Snoussi, F. Abdallah, and C. Richard. Guaranteed boxed localization in manets by interval analysis and constraints propagation techniques. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5. IEEE, 2008.
- F. Mourad, H. Snoussi, F. Abdallah, and C. Richard. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *Signal Processing, IEEE Transactions on*, 57(8): 3226–3239, 2009a.
- F. Mourad, H. Snoussi, F. Abdallah, and C. Richard. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on Signal Processing*, 57(8): 3226–3239, 2009b.
- R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Information Processing in Sensor Networks*, pages 333–348. Springer, 2003.
- E. Navarro, B. Peuker, and M. Quan. Wi-fi localization using rssi fingerprinting. In *California Polytechnic State University, USA*, pages 1–6, 2010. URL <http://digitalcommons.calpoly.edu/cpesp/17/>.
- A. Neumaier. *Interval methods for systems of equations*, volume 37. Cambridge university press, 1990.
- L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. Landmarc: indoor location sensing using active rfid. *Wireless networks*, 10(6):701–710, 2004.
- D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2926–2931. IEEE, 2001.
- D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1734–1743. IEEE, 2003.
- N. Noury, G. Virone, and T. Creuzet. The health integrated smart home information system (his 2): rules based system for the localization of a human. In *Microtechnologies in Medicine & Biology 2nd Annual International IEEE-EMB Special Topic Conference on*, pages 318–321. IEEE, 2002.
- A. Pal. Localization algorithms in wireless sensor networks: Current approaches and future challenges. *Network Protocols and Algorithms*, 2(1):45–73, 2010.
- R. Priwgharm and P. Chemtanomwong. A comparative study on indoor localization based on rssi measurement in wireless sensor network. In *Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6, May 2011.
- N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000.
- N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 1–14. ACM, 2001.

-
- M. Quan, E. Navarro, and B. Peuker. Wi-fi localization using rssi fingerprinting. 2010.
- C. S. J. Rabaey and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX technical annual conference*, 2002.
- J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr, D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 33(7):42–48, 2000.
- T. S. Rappaport, J. Reed, and B. D. Woerner. Position location using wireless communications on highways of the future. *Communications Magazine, IEEE*, 34(10):33–41, 1996.
- B. Ristic and P. Smets. Target identification using belief functions and implication rules. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(3):1097–1103, 2005.
- J. J. Robles, M. Deicke, and R. Lehnert. 3d fingerprint-based localization for wireless sensor networks. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, pages 77–85. IEEE, 2010a.
- J. J. Robles, M. Deicke, and R. Lehnert. 3d fingerprint-based localization for wireless sensor networks. In *International Workshop on Positioning, Navigation and Communication WPNC, Dresden, Germany (March 2010)*, pages 77– 85, 2010b.
- N. Roseveare and B. Natarajan. Distributed tracking with energy management in wireless sensor networks. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3494–3511, 2012.
- A. Rozyyev, H. Hasbullah, and F. Subhan. Combined k-nearest neighbors and fuzzy logic indoor localization technique for wireless sensor network. *Research Journal of Information Technology*, 4(4):155–165, 2012.
- A. Runge, M. Baunach, and R. Kolla. Precise self-calibration of ultrasound based indoor localization systems. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–8. IEEE, 2011.
- D. Sam-Haroud and B. Faltings. Consistency techniques for continuous constraints. *Constraints*, 1(1-2):85–118, 1996.
- M. Sangnier, J. Gauthier, and A. Rakotomamonjy. Filter bank kernel learning for nonstationary signal classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3183–3187. IEEE, 2013.
- A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001.
- A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 112–121. ACM, 2002.
- J. Schiller and A. Voisard. *Location-based services*. Elsevier, 2004.
- A. R. Schuler, A. Grammatikos, and K. A. FEGLEY. Measuring rotational motion with linear accelerometers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-3(3):465–472, 1967.

-
- H. Schweinzer and M. Syafrudin. Losnus: An ultrasonic system enabling high accuracy and secure tdoa locating of numerous devices. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–8. IEEE, 2010.
- L. Schwiebert, S. K. Gupta, and J. Weinmann. Research challenges in wireless networks of biomedical sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 151–165. ACM, 2001.
- G. Shafer et al. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.
- B. Shahid, A. A. Kannan, N. H. Lovell, and S. Redmond. Ultrasound user-identification for wireless sensor networks. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 5756–5759. IEEE, 2010.
- Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz. Localization from connectivity in sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 15(11):961–974, 2004.
- Y. Shang and W. Ruml. Improved mds-based localization. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2640–2651. IEEE, 2004.
- Y. Shang, W. Ruml, Y. Zhang, and M. P. Fromherz. Localization from mere connectivity. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 201–212. ACM, 2003.
- R. Shorey, A. Ananda, M. C. Chan, and W. T. Ooi. *Mobile, wireless, and sensor networks: technology, applications, and future directions*. John Wiley & Sons, 2006.
- E. Sifuentes, O. Casas, and R. Pallas-Areny. Wireless magnetic sensor node for vehicle detection with optical wake-up. *Sensors Journal, IEEE*, 11(8):1669–1676, 2011.
- S. N. Simic and S. Sastry. Distributed localization in wireless ad hoc networks. Technical report, Technical Report UCB/ERL, 2002.
- G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 1–12. ACM, 2004.
- P. Smets. Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of approximate reasoning*, 9(1):1–35, 1993.
- P. Smets and R. Kennes. The transferable belief model. *Artificial intelligence*, 66(2):191–234, 1994.
- I. Stojmenovic. *Handbook of sensor networks: algorithms and architectures*, volume 49. John Wiley & Sons, 2005.
- T. Sunaga. Theory of an interval algebra and its application to numerical analysis. *Research Association of Applied Geometry (RAAG) Memoirs*, 2:29–46, 1958.
- T. Sunaga. Theory of an interval algebra and its application to numerical analysis. *Japan Journal of Industrial and Applied Mathematics*, 26(2):125–143, 2009.

-
- D. J. Suroso, P. Cherntanomwong, P. Sooraksa, and J.-i. Takada. Fingerprint-based technique for indoor localization in wireless sensor networks using fuzzy c-means clustering algorithm. In *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*, pages 1–5. IEEE, 2011.
- A. Tapus and R. Siegwart. A cognitive modeling of space using fingerprints of places for mobile robot navigation. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1188–1193. IEEE, 2006.
- J. Teng, H. Snoussi, and C. Richard. Decentralized variational filtering for target tracking in binary sensor networks. *Mobile Computing, IEEE Transactions on*, 9(10):1465–1477, 2010.
- R. Tharmarasa, T. Kirubarajan, A. Sinha, and T. Lang. Decentralized sensor selection for large-scale multisensor-multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):1307–1324, 2011.
- Thorsten Joachims. An implementation of Support Vector Machines (SVMs). 2008. <http://svmlight.joachims.org/>.
- G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, et al. A macroscope in the redwoods. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 51–63. ACM, 2005.
- I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004.
- I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 154–165. ACM, 2005.
- F. Viani, P. Rocca, L. Lizzi, M. Rocca, G. Benedetti, and A. Massa. Wsn-based early alert system for preventing wildlife-vehicle collisions in alps regions. In *Antennas and Propagation in Wireless Communications (APWC), 2011 IEEE-APS Topical Conference on*, pages 106–109. IEEE, 2011.
- E. Walter and L. Jaulin. Guaranteed characterization of stability domains via set inversion. *Automatic Control, IEEE Transactions on*, 39(4):886–889, 1994.
- D. L. Waltz. Generating semantic descriptions from drawings of scenes with shadows. Technical report, DTIC Document, 1972.
- J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.
- H. Zhang and P. N. Pathirana. Uplink power control via adaptive hidden-markov-model-based pathloss estimation. *Mobile Computing, IEEE Transactions on*, 12(4):657–665, 2013.
- P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 227–238. ACM, 2004.
- F. Zhao and L. J. Guibas. *Wireless sensor networks: an information processing approach*. Morgan Kaufmann, 2004.

Xiaowei LV

Doctorat : Optimisation et Sûreté des Systèmes

Année 2015

Localisation indoor dans les réseaux de capteurs sans fil

Ce manuscrit est dédié à la résolution du problème de localisation dans les réseaux de capteurs sans fil mobiles. Les méthodes développées se basent principalement sur des caractéristiques de fingerprints ainsi que sur des informations de mobilité. Les premières s'attaquent aux valeurs de RSSI entre capteurs tandis que les deuxièmes prennent en considération la mobilité des capteurs mesurée à l'aide d'accéléromètres et de gyroscopes. La combinaison des données collectées est effectuée dans le cadre de l'analyse par intervalles, ou bien du filtrage de Kalman. Les travaux proposés introduisent des modèles de mobilité d'ordres un, deux ou trois, permettant d'approximer au mieux les trajectoires des capteurs à l'aide des accélérations mesurées. Ceux-ci sont couplés à l'algorithme des K plus proches voisins, d'abord dans un système centralisé. Ensuite, les modèles de mobilités sont améliorés pour prendre en compte les rotations des nœuds. Une méthode de localisation décentralisée est également proposée dans ce qui suit, s'adaptant au mécanisme fonctionnel des réseaux de capteurs de grande échelle. Enfin, ce manuscrit propose une méthode de zonage visant à déterminer les zones dans lesquelles les capteurs résident. La méthode proposée aborde le problème de zonage en utilisant à la fois la théorie des fonctions de croyance et l'analyse par intervalles.

Mots clés : réseaux de capteurs (technologie) - calcul sur des intervalles - accéléromètres - estimation, théorie de l' - Kalman, filtrage de - gyroscopes.

Indoor Localization in Wireless Sensor Networks

This thesis is dedicated to solve the localization problem in mobile wireless sensor networks. It works mainly with fingerprints features and inertial movements information. The former tackles the RSSIs values between sensors while the latter deals with the objects movement attitude by using accelerometer and gyroscope. The combination of both information is performed in terms of interval analysis, or Kalman filtering. The proposed work introduces three orders mobility models to approximate nodes trajectories using accelerations, combined then to the weighted K nearest neighbors algorithm in a centralized scheme. Then the mobility models are extended up to the inertial information taking into consideration the rotations of the nodes. A decentralized localization method is also proposed in the following in view of the working mechanism of large scale sensor networks. Finally, this thesis proposes a zoning localization method aiming at determining the zones in which the nodes reside. The proposed method addresses the zoning problem by using both the belief functions theory and the interval analysis.

Keywords: sensor networks - interval analysis (mathematics) - accelerometers - estimation theory - Kalman filtering - gyroscopes.

Thèse réalisée en partenariat entre :

